Ranked Time Series Matching by Interleaving Similarity Distances

Conference Paper · December 2017

DOI: 10.1109/BigData.2017.8258343

3 authors, including:

Cuong Nguyen
Worcester Polytechnic Institute
2 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Rodica Neamtu
Worcester Polytechnic Institute
12 PUBLICATIONS **8** CITATIONS
SEE PROFILE

Ranked Time Series Matching by Interleaving Similarity Distances

Cuong Nguyen
Worcester Polytechnic Institute
ctnguyendinh@wpi.edu

Charles Lovering
Worcester Polytechnic Institute
cjlovering@wpi.edu

Rodica Neamtu
Worcester Polytechnic Institute
rneamtu@cs.wpi.edu

***Abstract*—Similarity analytics of time series data are critical for a wide range of applications ranging from medical to financial, and from weather forecasting to image processing. Yet these analytics tasks are known to be prohibitively expensive for large data sets, especially when accounting for varying temporal alignments and lengths. Our proposed framework tackles this challenge by adopting a preprocess-once and query-many-times paradigm. We extend a previous formal model interleaving the inexpensive Euclidean distance with the robust Dynamic Time Warping (DTW) to retrieve the k most similar matches to a given sample sequence. Our extended ONline EXploration of top k time series similarity system (K-ONEX) first encodes similarity relationships by compressing the raw time series into Euclidean-based groups; these groups are further explored using the elastic DTW to find similar sequences of any length and temporal alignment with response times that are almost as fast as retrieving only one best match. Our empirical results illustrate that K-ONEX provides response times that are 2-3 orders of magnitude faster than the benchmark and state-of-the-art methods while achieving 100% accuracy by exploring less than 0.5% of the sequences in each dataset.**

***Index Terms*—subsequence matching, data mining, time series analytics, k-similarity search, dynamic time warping, visualization of time series similarity.**

I. Introduction

The monitoring of stock trends, weather, and medical history through various sensors has led to a dramatic increase in the availability of large-scale collections of time series. Mining this staggering amount of data is a daunting task, especially when trying to reveal insights beyond the traditional retrieval of a single best match to a given sample. For example, a financial analyst might be interested in finding stocks with similar historical price trends to avoid constructing portfolios that are not diverse; or a neurologist might determine abnormal brain activity by comparing brain signals of a patient to similar time series in a massive electroencephalogram database [1]. Thus, many techniques have been proposed to find the best match or groups of sequences similar to a given sample [2], [3], [4], [5]. Complex data mining tasks require the use of elastic alignment tools such as dynamic time warping (DTW) [6], whose robustness enables the comparison of sequences with different lengths and those that are not aligned in time. This "elasticity" comes with a quadratic complexity [6] which is compounded by the fact that DTW is not a metric, and as such there is no proven triangle inequality to help mitigate the cost of using it over large datasets. Many research efforts have been dedicated to improving the scalability of DTW; however, most of them focused on finding a single best match to a given sample. Potential extensions to finding more than one match are difficult to achieve without forgoing much of the efficiency advantage of the proposed techniques [5], [7], [8].

Motivating examples

We show in the following examples that finding multiple similar matches for a given sample sequence is beneficial and necessary for analysts to better understand their datasets.

1) The best match is not always the most reasonable answer. It has been shown that DTW can produce pathological results through paths that yield minimum distance but do not have much practical meaning [9]. Although constraints such as the Sakoe-Chiba band [10] can help, they require an iterative process of adjusting parameters and rerunning the entire search. In many cases, a better match might be found by just retrieving more similar sequences. As illustrated in Fig. 1, on the left side, the sequence represented by the solid red line (denoted as "1st best match") is retrieved as the best match of the sample query (blue dashed line) due to a pathological warping path (marked by the arrow). However, we show on the right side of the same figure that the second match (solid red line, denoted as "2nd best match") is a "more reasonable" match, because it better captures the shape similarity, despite having a higher distance to the sample than the first match.

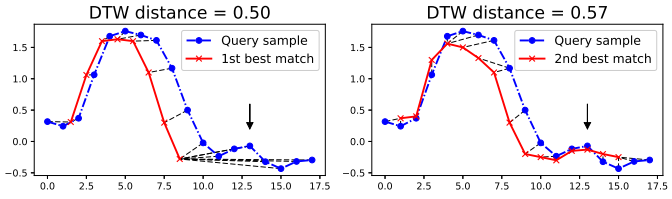

Fig. 1. The first best match has a smaller DTW due to over-warping at the right end point of the path, pointed by the arrow. The second best match is a more reasonable answer capturing the shape similarity despite the higher DTW between this match and the sample.

2) The best match is not always the most interesting answer. Sometimes the best match can be too trivial to be useful. For example, the stock price of a subsidiary company is likely to follow the trend of the stock price of its holding

company. Thus, finding only the best match in this case would not provide meaningful additional information. Instead, retrieving several similar sequences would enable a domain expert to gain better insights by incorporating their domain knowledge to ignore trivial results and analyze the more interesting ones. As shown in Fig. 2, the sample (blue line) is very similar in a trivial manner to the sequences (dashed red lines) in the top two and bottom left sub-figures and having the three lowest distances (DTW); the fourth match illustrated in the bottom right sub-figure provides better insights, despite having a higher similarity distance to the sample.

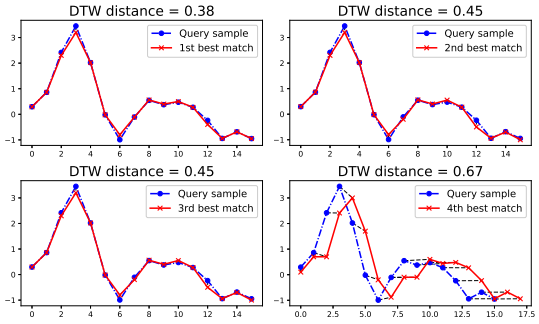

Fig. 2. The first three best matches are trivial and overlap the query almost completely; furthermore, the values in the third best match happen to be identical to the values of the second best match, although they are different sequences altogether. The fourth best match potentially provides more information by capturing the time misalignment and shape similarity.

3) Taking advantage of the existing intermediate computation to retrieve many similar sequences for almost no extra computational expense. Searching for similar time series involves exploring the dataset by performing all pairwise similarity comparisons between sequences, due to the non-metric nature of DTW. The quadratic complexity of DTW leads to very high computational expenses, which are only used to retrieve one single match [4], [5]. Reusing the existing computation to retrieve many similar matches to a given sequences would be beneficial by giving analysts a better understanding of their data at much-reduced cost.

Research Challenges and State-of-the-Art Limitations.

Finding the k-most-similar sequences to a given sample in large time series datasets is a daunting task, especially when accounting for sequences with various temporal alignments and lengths. The mathematical complexity of exploring a dataset containing N time series, each of length l, is $\mathcal{O}(Nl^2)$. Performing similarity comparisons using robust alignment tools such as DTW whose complexity is quadratic for all these subsequences in real-world datasets[1] is not practical as the cardinality of data increases.

The state-of-the-art has been trying to address the compromise between the choice of similarity distance and time responsiveness. Many systems speed up their response time by relying on inexpensive-to-compute distances such as Euclidean [8], [11], but they cannot handle sequences of different lengths and alignments. The alternative, the use of robust alignment tools such as DTW, is generally overshadowed by the fact that its computational complexity [6] leads to decreased responsiveness and poor scaling. Thus, researchers have to make a difficult choice between the ability to perform meaningful comparisons and the increased response times.

Retrieving k-most-similar sequences. As shown in our previous examples, getting insights into datasets often involves not just finding one best match for a sequence, but also discovering other similar sequences. Most state-of-the-art systems focus on finding the best match [5], [8] and cannot be easily extended to find many sequences without dramatically increasing the computational expenses and response times. Analysts would benefit from the ability to retrieve as many similar sequences as needed to better understand their data.

Our Approach.

The cornerstone of our work is the exploration of a compacted dataset by using the robust DTW (Sec. II-A) *only when absolutely needed*. The quality of our results of exploring "similarity groups" constructed with the inexpensive Euclidean Distance (ED) is guaranteed by our theoretical framework and further illustrated by our experimental results. Our mining strategy based on "extending" similarity properties to many groups allows us to retrieve any desired number of similar time series within almost the same time as retrieving only one single best match to a given sample query.

Contributions.

1) We formally prove a modified triangle inequality between ED and DTW that extends the DTW-based exploration from one to many ED-based groups to find the desired number of similar sequences. (Sec. III)
2) We devise an efficient strategy for fast k-similarity search, powered by indexing the group representatives and further optimized by early pruning and efficient traversal strategies. (Sec. IV)
3) We conduct experiments on eight benchmark datasets from the UCR archive [12] to show that our K-ONEX is 100% accurate and 2-3 orders of magnitude faster than the benchmark and state of the art methods. (Sec. V-B)
4) Our precomputed similarity relationships between subsequences preserved in our "similarity groups" give the analysts a "similarity-centric panorama" of the dataset. Being able to visualize this panorama by showing the distribution of similar sequences enables analysts to better understand their data. (Sec. V-B)
5) Our case study using a medical dataset shows how our method could help medical staff identify heart conditions. (Sec. V-C)

II. General Similarity Concepts.

A. Dynamic Time Warping Overview

Dynamic Time Warping (DTW) between two time series X and Y aligns these sequences according to their shapes

[1]https://followthedata.wordpress.com/2014/06/24/data-size-estimates/

rather than the ordering of their values. Suppose we have two time series $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_m)$. To align them 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, a set of elements that forms a path in the matrix from $(1, 1)$ to (n, m), is found. The t^{th} element of P denoted by $p_t = (i_t, j_t)$ 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:

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

The **DTW distance** is the weight of the path with minimum weight: $DTW(X, Y) = \min_P(w(P))$.

More details about the properties and dynamic programming strategy for computing this path can be found in [6], [13].

B. Online Similarity Exploration.

We now define some general concepts used in our theoretical framework for exploring time series similarity.

A **time series** X of length n denoted by $X = (x_1, x_2, ..., x_n)$ is an ordered set of n real values.

A **dataset** $D = \{X_1, X_2, ..., X_N\}$ is an ordered collection of N such time series.

Definition 2: The **subsequence of a time series** which is denoted by $(X_i)_j^l$ is a time series X_i of length l starting at position j where $1 \leq l \leq n$ and $1 \leq j \leq n - l + 1$ and i is the identifier of the time series.

Similar to [13], we define normalized distances in order to establish our theoretical foundation of time-warped retrieval.

Definition 3: Given two sequences $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_n)$ we define their **normalized Euclidean Distance** as

$$\overline{ED} = \frac{ED(X, Y)}{\sqrt{n}}. \quad (2)$$

Definition 4: Given two sequences $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_n)$ we define their **normalized DTW** as

$$\overline{DTW} = \frac{DTW(X, Y)}{2n}. \quad (3)$$

We consider two sequences to be similar if their pairwise distance is within a given *similarity threshold*, denoted by ST.

Definition 5: Two time series X and Y are **similar** if the chosen distance such as $\overline{ED}$ or $\overline{DTW}$ between them is within a given similarity threshold ST, or $Dist(X, Y) \leq ST$ where $Dist \in \{\overline{ED}, \overline{DTW}\}$ and ST can be any value between 0 and 1.

Similar to [13], we group subsequences of the same length that are similar according to Def. 5 using the inexpensive ED distance into "similarity groups" and summarize each of these groups by their representative.

Definition 6: Given the set T of all possible subsequences $(X_i)_j^l$ of the N time series of dataset D, assume these subsequences $(X_i)_j^l \in$ T are grouped into groups with their respective representatives R_k^l, such that all subsequences $(X_i)_j^l \in$ T are in one and only one group G_k^l. These groups are defined to be **similarity groups**, denoted by G_k^l, if the following three properties hold:

1) All subsequences $(X_i)_j^l$ in a group G_k^l must have the same length l.
2) $\overline{ED}$ between any $(X_i)_j^l$ in G_k^l and the representative R_k^l of this group G_k^l is smaller than half of the similarity threshold ST used by the system, that is $\overline{ED}((X_i)_j^l, R_k^l)) \leq ST/2, \forall i \in [1, N], \forall l \in [1, n], \forall j \in [1, n - l + 1]$.
3) $\overline{ED}$ between the subsequence $(X_i)_j^l$ and the representative R_k^l of the group G_k^l is the smallest compared to $\overline{ED}$ of $(X_i)_j^l$ and all other representatives R_p^l of the same length l defined over D, or $\overline{ED}((X_i)_j^l, R_k^l)) \leq \overline{ED}((X_i)_j^l, R_p^l)), \forall i \in [1, N], \forall l \in [1, n], \forall j \in [1, n - l + 1], \forall p \in [1, g]$, where g denotes the number of representatives of length l.

III. Theoretical Foundation for k-similarity Search interleaving ED and DTW.

We describe here the theoretical foundation for our strategy to retrieve k similar sequences by interleaving ED and DTW. The ONEX theoretical foundation [13] established a triangle inequality between $\overline{ED}$ and $\overline{DTW}$ applied to the sequences in a specific similarity group, its representative and any given sample query. This inequality guarantees that the $\overline{DTW}$ between a sample query Q and any sequence X in a group G_k^l as in Def. (6) is smaller than ST if $\overline{DTW}(R, Q) \leq \frac{ST}{2}$, where R is the representative of that group.

We now extend this property to all the similarity groups whose representatives have pairwise $\overline{DTW}$ with Q close to the similarity threshold ST. This allows us to guarantee the results of exploring many qualifying groups together to retrieve any desired number of sequences similar to a given query sample, instead of exploring the entire dataset.

Lemma 1: **Extended Triangle Inequality Based on Interleaving ED and DTW.** Given $X = (x_1, x_2, \ldots, x_n)$ an arbitrary sequence of length n in any group as per Def. (6), with the representative of a group $R = (r_1, r_2, \ldots, r_n)$ and a sample sequence $Q = (q_1, q_2, \ldots, q_m)$, then the following is true: if $\overline{ED}(X, R) \leq ST/2$, $\overline{DTW}(R, Q) \leq d$, then $\overline{DTW}(X, Q) \leq d + ST/2$, where d is the value of the distance between R and Q.

This Lemma allows us, based on the value of d, to guarantee the results of exploring more than one of our similarity group using DTW. According to Def. 6, the **similarity groups** satisfy the first condition of the Lemma. Thus, we now want to establish the values of d for which the similarity between X and a sample query Q is guaranteed, or the values for which $\overline{DTW}(R, Q)$ is close to ST.

Proof for sequences of the same length.

Given the assumptions of the Lemma when $m = n$, we have

$$\overline{ED}(X,R) \leq \frac{ST}{2} \quad (4) \qquad \overline{DTW}(R,Q) \leq d. \quad (5)$$

We want to find the values of d for which the following is true:

$$\overline{DTW}(X,Q) \leq d + \frac{ST}{2}. \quad (6)$$

Expanding Eq. (4) as per the definition of ED and $\overline{ED}$ we get

$$ED(X,R) = \sqrt{\sum_{i=1}^{n}(r_i - x_i)^2} \leq \sqrt{n}\frac{ST}{2}.$$

Squaring this we get:

$$ED^2(X,R) = \sum_{i=1}^{n}(r_i - x_i)^2 \leq n\frac{ST^2}{4}. \quad (7)$$

We define matrices $M(Q,R)$ and $M(Q,X)$ as in Sec. II. Given the assumptions related to this case we know that there is a warping path P in $M(Q,R)$ from $(1,1)$ to (n,n) with the DTW weight at most d. We now have to show that there is a warping path from $(1,1)$ to (n,n) in $M(Q,X)$ with weight at most $d + \frac{ST}{2}$. In fact we show that the same warping path P from $M(Q,R)$ is satisfactory. Let P be $P = (p_1, p_2, \ldots, p_t, \ldots, p_T)$, where $n \leq T \leq 2n-1$, $p_1 = (1,1)$, $p_T = (n,n)$, $p_t = (i_t, j_t)$.

From Def. (4) and the assumptions of the Lemma we know:

$$\sqrt{\sum_{t=1}^{T}(q_{i_t} - r_{j_t})^2} \leq 2nd. \quad (8)$$

Squaring this, we get:

$$\sum_{t=1}^{T}(q_{i_t} - r_{j_t})^2 \leq 4n^2d^2. \quad (9)$$

Based on Def. (4), we can also re-write Eq. (6) as

$$\overline{DTW}(X,Q) = \frac{\sqrt{\sum_{i=1}^{T}(q_{i_t} - x_{j_t})^2}}{2n} \leq (\frac{ST}{2} + d). \quad (10)$$

By distributing the denominator and squaring Eq. (10), we now want to prove the following inequality:

$$\sum_{t=1}^{T}(q_{i_t} - x_{j_t})^2 \leq 4n^2(\frac{ST}{2} + d)^2. \quad (11)$$

We derive a direct expansion of a single term within the summation from Eq. (11), where the step of path *t* is arbitrary (and thus elided).

$$q_i - x_j = q_i - r_j + r_j - x_j = (q_i - r_j) + (r_j - x_j). \quad (12)$$

Using the Cauchy-Schwarz inequality [14], we get:

$$(q_i - x_j)^2 \leq 2(q_i - r_j)^2 + 2(r_j - x_j)^2.$$

Then using this and (9), we get:

$$\sum_{t=1}^{T}(q_{i_t} - x_{j_t})^2 \leq 2\sum_{t=1}^{T}(q_{i_t} - r_{j_t})^2 + 2\sum_{t=1}^{T}(r_{j_t} - x_{j_t})^2 \leq 2 \times 4n^2d^2 + 2\sum_{t=1}^{T}(r_{j_t} - x_{j_t})^2. \quad (13)$$

We estimate the second term as $\sum_{i=1}^{n}(r_i - x_i)^2$ with some terms repeated. The total number of repetitions is at most n, since the length of the warping path is at most 2n. Each fixed term is repeated at most $n-1$ times. Thus from Equations (7) and (13), we have:

$$\sum_{t=1}^{T}(q_{i_t} - x_{j_t})^2 \leq 8n^2d^2 + 2n\sum_{i=1}^{n}(r_i - x_i)^2 \quad (14)$$

$$\leq 8n^2d^2 + \frac{2n^2ST^2}{4} \quad (15)$$

$$\leq 8n^2d^2 + 2n^2(\frac{ST}{2})^2. \quad (16)$$

We now show that the right hand side of Eq. (16) is conditionally less than the right hand side of Eq. (11) for specific values of d.

$$8n^2d^2 + 2n^2(\frac{ST}{2})^2 \leq 4n^2(\frac{ST}{2} + d)^2$$

$$4d^2 + (\frac{ST}{2})^2 \leq 2(ST^2 + 2d(\frac{ST}{2}) + d^2)$$

$$0 \leq (\frac{ST}{2})^2 + 4d(\frac{ST}{2}) - 2d^2$$

$$0 \leq (\frac{ST}{2})^2 + 4d(\frac{ST}{2}) + 4d^2 - 6d^2$$

$$6d^2 \leq (\frac{ST}{2} + 2d)^2 \quad \Rightarrow \quad d \leq (1.11)ST.$$

Discussion.

The important result of this Lemma shown in the last equation is that for d smaller than $1.11ST$, the similarity between a sample sequence and the representative of a group can be extended to all sequences in that specific group, as well as any other "qualifying groups", i.e. for which the above condition holds. The practical implication of this result is that we can safely explore not only the similarity groups as defined in [13], but also the ones that are slightly above the similarity threshold and still guarantee 100% accurate results. This will play a key role in retrieving any desired number of similar sequences within almost the same time as retrieving one single best match.

Note on the Proof of sequences with the different lengths.

The $\overline{DTW}$ defined in Def. 4 also applies for to the scenario when sequences X and Y have different lengths, respectively *m* and *n*. Without loss of generality we consider here the case of $m \leq n$ but the proof is very similar for $n \leq m$. The division by $2n$ is indeed due to the warping path having length up to $m + n \leq 2n$. Then the matrix $M(X,Y)$ is an $m \times n$ matrix and the warping path connects $(1,1)$ to (m,n). Other than this, the proof for sequences of different lengths

and the proof for sequences of the same length are the same, and we arrive at the same inequality.

IV. Efficient Strategy for Retrieving K-similar Sequences

*We aim to retrieve any desired number k of **similar** time series in a dataset with a response time that is not much higher than just retrieving one single best match.*

Definition 7: We denote $k_e \geq k$ as the minimum number of sequences that we need to explore in the dataset to guarantee our results.

The top k similar sequences retrieved are a subset of these k_e sequences.

A. Strategy for retrieving K-most-similar sequences.

The K-ONEX strategy for retrieving k-most-similar sequences to a given sample Q involves a two-step process:
(1) we first find the groups whose representatives have the smallest $\overline{DTW}$ to Q and have at least k_e members combined;
(2) then we compute the pairwise $\overline{DTW}$ distances of at least k_e sequences to the sample Q and return the top k sequences with the smallest $\overline{DTW}$ distances to Q.

We first create a length-based index L where L_l is a list of representatives of length l, namely R_i^l, and we use it to retrieve the groups of each specific length. The construction methodology for the length-based groups is the same as in [13]. It has been shown [15] that in general the best match to a given sample is highly likely to be of the same or close length to that of the sample. Thus, we start our exploration with the representatives having the same length as the sample Q and continue by alternatively exploring those of next smaller and larger lengths. As we compute $\overline{DTW}$ between Q and each of the representatives, we place the corresponding groups in a max-heap H such that the group with the representative furthest away from Q is always at the root of H. The total number of sequences contained in the existing selected groups in H exceed k_e, or $\sum_i |H_i|$ exceeds k_e where $|H_i|$ is number of members in group R_i. Naturally, this condition is not satisfied while the heap is being initially filled. We only insert a new group if the distance from its representative to Q is smaller than that of the representative of the group at the root, denoted as d^*. We use the LB_{Keogh} lower bound and early-abandoning technique [5] to quickly prune unqualified representatives. After each insertion of a new representative, we maintain a minimally sufficient number of groups in H by continuously popping the root of H until the next pop action results in $\sum_i |H_i|$ less than k_e. At the end of this process, H holds the groups whose representatives are closest to Q and collectively have a total number of sequences $\sum_i |H_i| \geq k_e$. These sequences are the candidates from which we select the k most similar sequences to the sample query.

We then select k sequences from the collective $\sum_i |H_i|$ sequences retrieved during the initial search for the representatives closest to Q, as shown in Fig. 3. Note that the number of sequences we search is slightly greater than k_e as the final group in H might have a greater than the minimum number of sequences necessary to reach k_e. To retrieve these sequences we maintain a similar max-heap of sequences H_s where the sequence with the largest $\overline{DTW}$ to Q is always at the root. We insert new sequences into this heap using a similar optimized technique to the one described above for representatives. The only difference between this new heap H_s and the previous H is that here we keep at most k sequences in H_s by popping the root every time its size exceeds this limit.

Algorithm 1 Finding k similar time series to query q.

Precondition: k and k_e are integers, Q is the sample sequence, L is the length index of the representatives. MAX-HEAP creates a max-heap. PEEK fetches the distance value d^* at the root of the heap. PUSH adds a tuple to the heap, maintaining heap properties. POP pops and fetches the root group of the heap. TRY-POP returns a new heap with the root being popped without modifying the original heap. GROUP-OF fetches all sequences in a group that corresponds to a given representative.

function KONEX(k, k_e, Q, L)
  $H \leftarrow$ SEARCHREPRESENTATIVES(k, k_e, Q, L)
  $X_k \leftarrow$ SEARCHEXTENDEDK(k, Q, H)
  return X_k
function SEARCHREPRESENTATIVES(k, k_e, Q, L)
  $H \leftarrow$ MAX-HEAP()
  for $l \leftarrow 1$ to $|L|$ **do**
    for $j \leftarrow 1$ to $|L_l|$ **do**
      $d \leftarrow \overline{DTW}(Q, R_j^l)$
      $d^* \leftarrow$ PEEK(H)
      if $d \leq d^*$ **then**
        PUSH($H, (d, R_j^l)$)
        POP(H) UNTIL $\sum_i |\text{TRY-POP}(H)_i| < k_e$
  return H
function SEARCHEXTENDEDK(k, Q, H)
  $H_s \leftarrow$ MAX-HEAP()
  for $i \leftarrow 1$ to $|H|$ **do**
    $R \leftarrow H_i$
    $G \leftarrow$ GROUP-OF(R)
    for $g \leftarrow 1$ to $|G|$ **do**
      $S \leftarrow G_g$
      $d \leftarrow \overline{DTW}(Q, S)$
      $d^* \leftarrow$ PEEK(H_s)
      if $d \leq d^*$ **then**
        PUSH($H_s, (d, S)$)
        POP(H_s) UNTIL $|H_s| \leq k$
  return H_s ▷ Return best k sequences in a vector.

B. Complexity of retrieving k similar sequences.

The complexity of finding the k most similar sequences to a given sample can be broken down in three parts:
(1) the complexity of selecting the representatives of the groups that contain the k_e sequences.

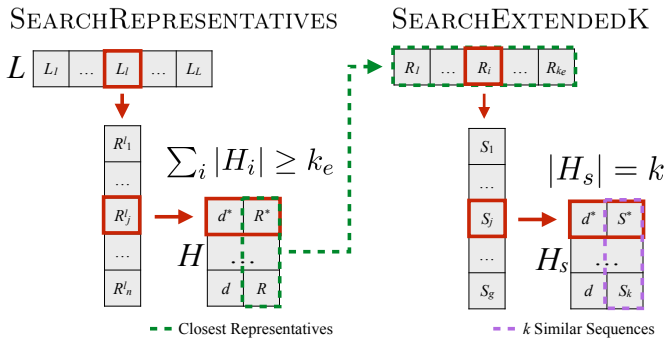

Fig. 3. Primary structures used by Algo. (1). Left, the list of representatives L for each length l is searched. For each representative R_j^l, if $\overline{DTW}(Q, R_j) \leq d^*$, (where d^* is the max-value in H), then R_j is added to H. After retrieving the top k_e sequences, they are explored to find the best k as shown on the right side of the figure.The heap H_s maintains the best k sequences to find the most similar k sequences.

(2) the complexity of selecting k sequences from the collection of sequences found in step 1.

(1) The complexity of retrieving the groups that contain the k_e most similar sequences is $O(|\boldsymbol{G}|log(|\boldsymbol{G'}|)n^2)$, where $\boldsymbol{G}$ is the set of examined groups, and $\boldsymbol{G'}$ is the set of groups selected as being the most similar to the given sample and also having a collective number of sequences greater than or equal to k_e. Here n^2 is the complexity of performing DTW pairwise comparisons of sequences of length n. Practically $|\boldsymbol{G'}|$ is much smaller than $|\boldsymbol{G}|$, so the complexity can be approximated as $O(|\boldsymbol{G}|n^2)$.

(2) We define k' as $\sum_i |\boldsymbol{G_i'}|$. We show later in our experimental evaluation that in general, k' is on the same order as k_e. The complexity of retrieving k sequences in G' is then $O(k'log(k)n^2)$. Again, because k is small, we approximate this to be $O(k'n^2)$.

Based on these components, the overall complexity of retrieving k similar sequences is $O((|\boldsymbol{G}|+k')n^2)$, with $k' \approx k_e$. We show later in our experimental evaluation that this k_e amounts to a very small percentage of the total number of sequences in the dataset to achieve 100% accuracy.

V. EXPERIMENTAL EVALUATION

A. Datasets and Experimental Methodology

We run experiments on eight datasets of varying sizes from small to large from the well-known UCR Time Series Classification Archive [12]. Each dataset has two disjoint sets: a Test set and a Training set. We use the Training sets to select our queries. We use the Test sets, which are larger, to run our experiments on retrieving k-most-similar time series. To avoid any confusion, we designate in this section the corresponding Test sets as DATA and the Training sets as QUERY of each dataset. For all datasets we first normalize all sequences X to the range $[0, 1]$ by using the formula $\frac{X_i - min(X_i)}{max(X_i) - min(X_i)}$.

Alternative methods. We compare our system **K-ONEX** to the benchmark brute force method **BF** that computes all pairwise comparisons using DTW, and to the popular data reduction method **PAA** (Piecewise Aggregate Approximation) [4] which finds an approximate solution by using an average approximation. Similar to our system, we use a max heap to maintain the top k results for both benchmark methods. All three methods are implemented in C++11, and run on a system with a 2.5 GHz Intel Core i7 CPU and 16 GB of memory. We perform three classes of experiments:

1) **k-Similarity Search.** We measure the online response time and accuracy for retrieving the top k-similar sequences to a given sample query averaged over multiple runs for different datasets varying in size from small to large. We perform "queries in the dataset" experiments by choosing our sample queries from our newly designated DATA set. We perform "queries not in the dataset" experiments by choosing the sample queries from our so-called QUERY set. We vary the number of query sequences randomly chosen from each dataset. We retrieve the top k-similar sequences in each DATA set corresponding to specific datasets, preprocessed with $ST = 0.1$. A summary of the datasets and number of sample queries is shown in Table I.
2) **Offline Computational Costs.** We estimate the cost of the offline construction of our similarity groups by measuring the **size** and the **preprocessing time** of our pregenerated information for varying similarity thresholds across the selected datasets.
3) **Visualization of compacted datasets** allows analysts to get a "similarity-centric panorama" of the distribution of similar sequences in each dataset for varying similarity thresholds.

TABLE I
DATASETS AND SAMPLE QUERIES STATISTICS.

Name	# of sequences in DATA	# of samples from DATA	# of samples from QUERY
ItalyPowerDemand	30,084	40	40
ECG200	456,000	40	40
Synthetic Control	531,000	40	40
Gun-Point	1,676,250	25	25
MedicalImages	3,686,760	25	25
Face (All)	14,390,350	10	10
50Words	16,523,325	10	10
Wafer	70,852,824	10	10

B. Experimental Results

1) k-Similarity Search: **Evaluating accuracy.** The benchmark Brute Force is an exact solution, thus we measure the **accuracy** of K-ONEX and PAA methods by comparing their solutions to the ones returned by the Brute Force method using the L_1 distance [16], defined as

$$\|\boldsymbol{X} - \boldsymbol{Y}\|_1 = \sum_i |X_i - Y_i|$$

Here $\boldsymbol{X}$ is an ordered list of distance values between a sample query and the top k similar sequences returned by **K-ONEX** or **PAA**, while $\boldsymbol{Y}$ is the ordered list of distance values between the same sample query and the top k similar

sequences returned by **BF**. The value of L_1 is 0 when there is a perfect match between the results provided by **BF** and the alternative comparison method, meaning 100% accuracy. The higher the value of L_1, the lower is the accuracy of the alternative method by comparison to **BF**.

We perform this experiment for varying numbers of desired similar sequences, namely $k = 1$, $k = 9$, $k = 15$. We vary k_e, defined in Def. 7, incrementally from the desired number k to at most 5% of the total size of DATA, or until we obtain 100% accuracy for a sample query, whichever comes first.

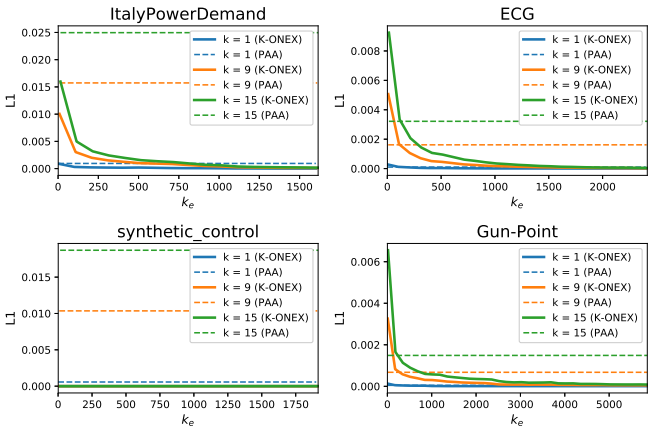

Fig. 4. Average accuracy across first four datasets. The results are averaged over all sample queries for each dataset.

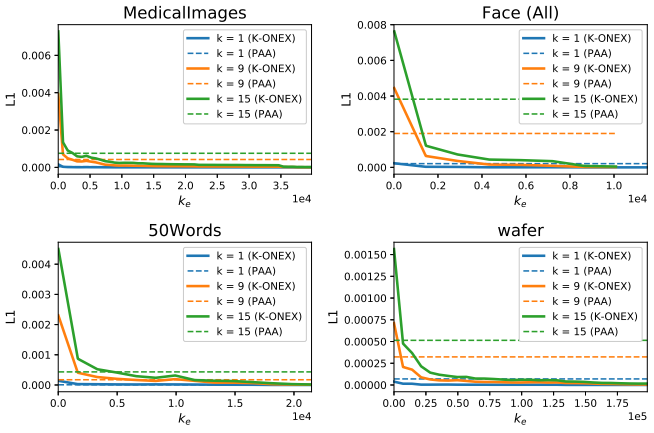

Fig. 5. Average accuracy across last four datasets. The results are averaged over all sample queries for each dataset.

We plot in Fig. 4 and Fig. 5 the accuracy for **K-ONEX** and **PAA** based on L_1 to compare the k-most-similar sequences respectively returned by the two methods. Since **BF** is an exact solution, it is not included in the plot. Each figure depicts the values of L_1 for **K-ONEX** and **PAA** based on the number k_e. We note that the value of L_1 for our system becomes quickly smaller than that of **PAA** by setting k_e to less than 0.5% of the sequences in the dataset leading to 100% accuracy.

In Fig. 6, we show the average minimum percentage of the total number of sequences that need to be explored (denoted by k' in IV-B) to find the top 15 similar sequences with 100% accuracy. We use a log scale version in the inset plot to magnify the fact that this percentage is too small to be noticed on a linear scale. We show here that on average the number of explored sequences amounts for up to 0.5% of the total number of sequences to achieve the highest accuracy.

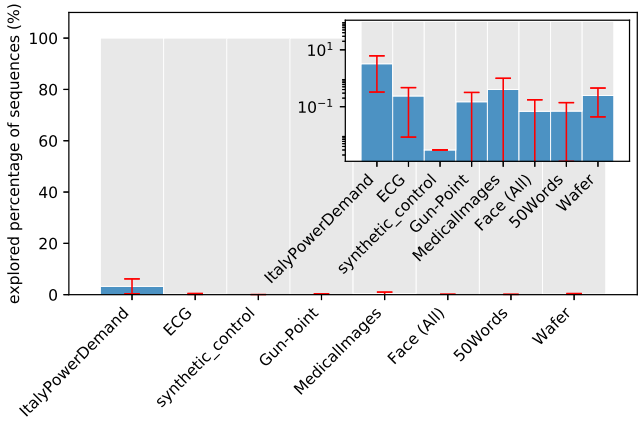

Fig. 6. Percentage of explored sequences for $k = 15$ across datasets.

Table II indicates the percentage of sequences that need to be explored in order to reach 100% accuracy for varying k values desired by analysts. This percentage is very low, as K-ONEX reaches the maximum accuracy by exploring on average as low as 0.221% of the sequences in each dataset for $k = 1$ and respectively as high as 0.552% of the sequences in each dataset for $k = 15$.

TABLE II
PERCENTAGE OF EXPLORED SEQUENCES VARYING K ACROSS DATASETS.

K	1	9	15
Min	0.0004%	0.002%	0.003%
Max	1.150%	2.583%	3.238%
Avg	0.221%	0.452%	0.552%

Evaluating response time. Fig. 7 depicts the average response times for all three methods across the eight datasets. We note that **K-ONEX** is by far faster than both benchmark methods, with average response times 1874 times faster than **BF** and 690 times faster than **PAA**.

In summary, these empirical results illustrate that our system provides response times that are 2-3 orders of magnitude faster than the benchmark methods while achieving perfect accuracy by exploring on average less than 0.5% of the sequences in each dataset.

2) Offline Computational Costs: Our goal is to find high-quality compact representations of our datasets through similarity groups that yield perfectly accurate results with very low response time. For this, we evaluate the size of our pregenerated data through the average size, compression rate and construction time for our similarity groups. Since the other two methods don't involve a preprocessing phase, we display

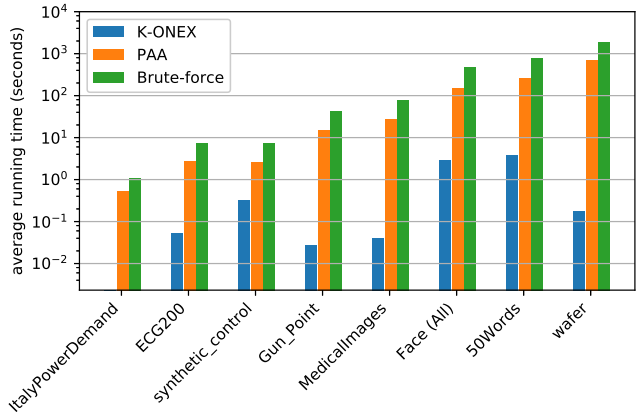

Fig. 7. Average response times across datasets.

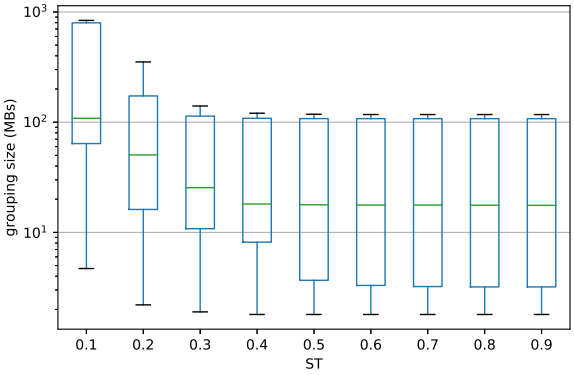

Fig. 9. Average group sizes across datasets varying ST.

only the preprocessing times and the size of the K-ONEX generated groups for varying similarity thresholds.

Fig. 8 depicts the average preprocessing time (in the box plot) and the average compression rate (in the line plot) for the eight selected datasets. We define the compression rate as

$$100\% - \frac{\#\text{ of group} + \text{avg. group size}}{\text{total \# of sequences}}\%$$

The preprocessing time decreases as ST increases. This is because for larger thresholds, more sequences are placed into the same similarity group, hence there will be a reduced number of groups. On the other hand, the compression rate increases for larger STs, it peaks at 0.4 and slowly decreases after that. The reason for this trend is that for larger STs, there are fewer groups generated, but the average number of sequences in each group increases.

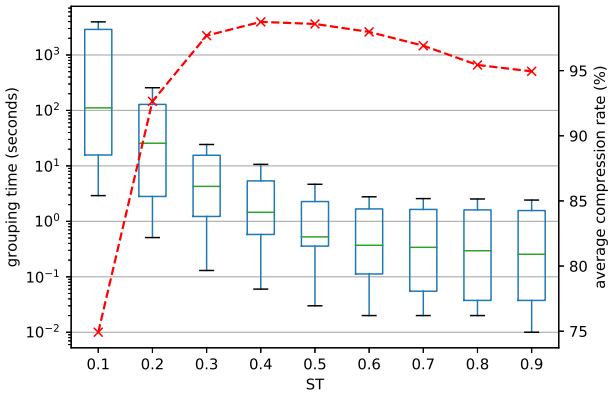

Fig. 8. Average construction times (in the box plot) and compression rates across datasets (in the line plot) varying ST.

Fig. 9 illustrates the average space needed for the grouping structures. As ST increases, the average size decreases and plateaus after $ST = 0.4$. This trend correlates with the compression rate trend described above.

In summary, our framework enables us to "compact" datasets efficiently offline and benefit from the ability to find any number of similar sequences for as many samples as desired, with 100% accuracy and lightning speed response times, comparable to the response times for retrieving one single match.

3) Visualization of compacted datasets: We depict in Fig. 10 the "changes" in the similarity panorama of the preprocessed datasets while varying ST. Each square corresponds to a specific dataset and ST pair, and contains a heat map consisting of multiple colored cells. Each cell represents a similarity group as defined in Def. 6. The area of a cell is proportional to the number of sequences in the group, while the color is correlated to the length of the sequences in the group. A cell with deeper blue represents a group of shorter-length sequences, and a cell with deeper orange is a group of longer-length sequences. For clarity of visualization, we only plot here groups in decreasing order of the number of member sequences until the total number of sequences in all plotted groups makes up 25% of the dataset. It is noticeable that the 25% of the sequences in each dataset tend to be distributed into more groups as we decrease ST and vice-versa.

Interestingly, this visualization also explains the underlying reason for which *ItalyPowerDemand* and *synthetic_control* are on the upper and lower extremes of the minimum explored percentage of sequences, as shown in Fig. 6. Specifically, the distribution of sequences in *ItalyPowerDemand* is not best for the considered range of ST, i.e. there are a few large groups, each containing a large number of all sequences in the dataset, thus "deforming" the dataset structure. This characteristic does not change significantly for varying STs. At the other end of the spectrum, *synthetic_control* appears much more dynamic in the considered ST range, i.e. the number of groups changes smoothly for different STs. For $ST = 0.1$ the groups of *synthetic_control* are compact while the dataset structure is still largely retained, therefore we achieve the highest accuracy by exploring a very small number of sequences.

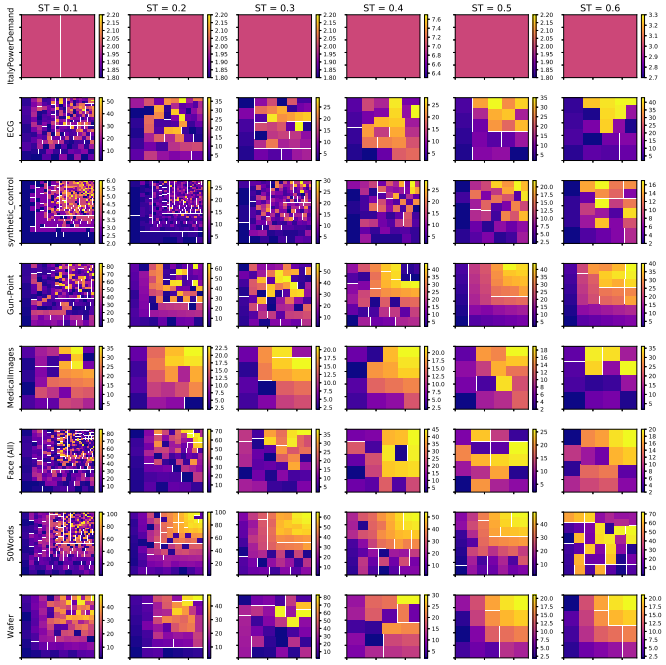

Fig. 10. "Similarity panorama" / Group heat maps for all datasets varying ST.

C. Case Study: using k similarity searches to help identify heart conditions.

We explore a real dataset containing ECG shapes and other information about patients suffering from arrhythmia [17], [18] to show how finding similar sequences can help doctors get new insights about these heart conditions. The MIT-BIH Arrhythmia Database, created by Beth Israel Deaconess Medical Center and MIT contains 48 half-hour excerpts of two-channel ambulatory ECG recordings, all of which were obtained from 47 subjects studied by the BIH Arrhythmia Laboratory between 1975 and 1979.

For our case study, we first select the first 500 points of the sample heart rate shape of the record labeled 107 in the dataset and use this sequence as our sample query. 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, displaying multiform PVCs. We explore the dataset to find the sequences that are most similar to our sample and belong to other patients. As shown in Fig. 11, the most similar sequences retrieved by K-ONEX belong to patients having records 200 and respectively 203. Both records indicate that the ECG shapes of these two male patients, 63 and respectively 43 years old, show PVCs that are multiform, including ventricular tachycardia, and ventricular trigeminy. Examining these similar matches could help medical staff better understand the similarities between the heart conditions of the three patients.

As a second example, we select a shorter sequence from the sample heart rate of the record labeled 109. This record belongs to a male patient of age 64, who has a first degree

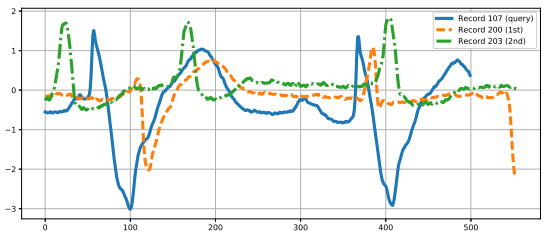

Fig. 11. First example of case study on the arrhythmia dataset.

AV block and multiform PVCs. Fig. 12 shows that the most similar sequences belong to patients with records 200, 203 and 201. These records respectively are of males, ages 64, 43, and a respectively 68, all displaying multiform PVCs, ventricular tachycardia, and ventricular trigeminy. Likewise, these similar matches portray the similarities between the records, aiding in better understanding the heart conditions of these patients.

In summary, medical staff could benefit from the ability to retrieve several similar matches to a given sequence. Such similarities could help identify patterns preceding cardiac arrests, atrial fibrillations or other conditions and contribute to better diagnosis and treatment.

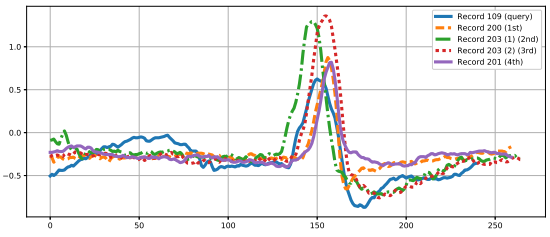

Fig. 12. Second example of case study on the arrhythmia dataset.

VI. Related Work

Euclidean Distance is one of the most frequently used distances [8], [11] due to its time and space efficiency, although it shows great limitations in comparing sequences of different lengths and with temporal misalignments [19], [4]. By contrast, DTW allows non-linear alignments between two time series to compare sequences that are similar, but locally out of phase [5]. Its popularity for mining time series similarity is only dampened by its high computational complexity. To reduce the time response of DTW, indexing techniques [8], [20], and other optimizations like early abandoning of DTW [5], cascading lower bounds to prune unpromising candidates, and reversing the query/data role by creating an envelope around the query sequence instead of the data [5] have been developed. We leverage some these techniques in our system, while also noting that simply extending their use to search for more than one best match in the aforementioned systems would not be efficient, as it would involve re-running the entire

search and leading to much increased response times.

Some systems [21] reduce the response time for the k nearest time series by only exploring whole sequences in a distributed environment. Others [11], [22] proposed efficient k-NN search algorithms, but they too focus only on whole time series matching, thus they would not scale very well to subsequence matching.

Closer to our work, [23] proposed ranked subsequence matching under time warping, finding top-k subsequences most similar to a query sequence by introducing the notion of the minimum-distance matching-window pair. Unlike us, they focus on deferred group subsequence retrieval to avoid excessive random disk I/Os and bad buffer utilization.

Range searches and nearest neighbor searches [7] are increasingly important in mining time series data, as we showed in our motivating examples. Reducing the cardinality of data by exploring meaningful sequences as representatives instead of the raw data [24] is a popular approach. Methods like the nearest centroid classifier [25] and k-means clustering replace a set of neighbors with their centroid. Conceptually similar, [26] and [27] reduce the data cardinality by grouping similar time series. K-ONEX is built on a general idea similar to [27] of finding representatives for groups of similar objects but we expand here the strategy of combining two well-known distances [13] to find the k most similar sequences.

VII. Conclusion

We introduce K-ONEX, a framework for *finding k-most-similar time series* by interleaving ED and DTW to produce guaranteed results with response times that are 2-3 orders of magnitude faster than benchmark methods. Our approach provides analysts with the ability to retrieve any desire d number k of similar sequences to a given sample with 100% accuracy within almost the same time that it would take to only retrieve one best match. K-ONEX renders more practical the retrieval of similar sequences in large time series datasets by allowing analysts to control the amount of data explored, balancing the trade-off between accuracy and latency.

Acknowledgment

We thank Prof. Elke Rundensteiner and Prof. Gabor Sarkozy from WPI for invaluable feedback. We thank the authors of the original ONEX framework, especially Ramoza Ahsan for sharing information about implementation.

References

[1] W. A. Chaovalitwongse, Y.-J. Fan, and R. C. Sachdeo, "On the time series k-nearest neighbor classification of abnormal brain activity," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 37, no. 6, pp. 1005–1016, 2007.

[2] S. Park, W. W. Chu, J. Yoon, and C. Hsu, "Efficient searches for similar subsequences of different lengths in sequence databases," in *Data Engineering, 2000. Proceedings. 16th International Conference on*. IEEE, 2000, pp. 23–32.

[3] Y. Sakurai, C. Faloutsos, and M. Yamamuro, "Stream monitoring under the time warping distance," in *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*. IEEE, 2007, pp. 1046–1055.

[4] E. J. Keogh and M. J. Pazzani, "Scaling up dynamic time warping for datamining applications," in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2000, pp. 285–289.

[5] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh, "Searching and mining trillions of time series subsequences under dynamic time warping," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 262–270.

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

[7] R. Agrawal, C. Faloutsos, and A. Swami, "Efficient similarity search in sequence databases," *Foundations of data organization and algorithms*, pp. 69–84, 1993.

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

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

[10] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE transactions on acoustics, speech, and signal processing*, vol. 26, no. 1, pp. 43–49, 1978.

[11] E. Keogh, K. Chakrabarti *et al.*, "Locally adaptive dimensionality reduction for indexing large time series databases," *ACM SIGMOD Record*, vol. 30, no. 2, pp. 151–162, 2001.

[12] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista, "The ucr time series classification archive," *URL www. cs. ucr. edu/˜ eamonn/time_series_data*, 2015.

[13] R. Neamtu, R. Ahsan, E. Rundensteiner, and G. Sarkozy, "Interactive time series exploration powered by the marriage of similarity distances," *Proceedings of the VLDB Endowment*, vol. 10, no. 3, pp. 169–180, 2016.

[14] J. M. Steele, *The Cauchy-Schwarz master class: an introduction to the art of mathematical inequalities*. Cambridge University Press, 2004.

[15] D. Yankov, E. Keogh, J. Medina, B. Chiu, and V. Zordan, "Detecting time series motifs under uniform scaling," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2007, pp. 844–853.

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

[17] 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.

[18] 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.

[19] S. Chu, E. Keogh, and et. al., "Iterative deepening dynamic time warping for time series." SIAM, 2002.

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

[21] C.-C. Hsu, P.-H. Kung, M.-Y. Yeh, S.-D. Lin, and P. B. Gibbons, "Bandwidth-efficient distributed k-nearest-neighbor search with dynamic time warping," in *Big Data (Big Data), 2015 IEEE International Conference on*. IEEE, 2015, pp. 551–560.

[22] N. Roussopoulos, S. Kelley, and F. Vincent, "Nearest neighbor queries," in *ACM sigmod record*, vol. 24, no. 2. ACM, 1995, pp. 71–79.

[23] W.-S. Han, J. Lee, Y.-S. Moon, and H. Jiang, "Ranked subsequence matching in time-series databases," in *Proceedings of the 33rd international conference on Very large data bases*. VLDB Endowment, 2007, pp. 423–434.

[24] H. Ding, Trajcevski, and et. al., "Querying and mining of time series data: experimental comparison of representations and distance measures," *Proceedings of the VLDB Endowment*, vol. 1, 2008.

[25] R. Tibshirani, T. Hastie, and et. al., "Diagnosis of multiple cancer types by shrunken centroids of gene expression," *Proceedings of the National Academy of Sciences*, vol. 99, 2002.

[26] L. Belbin, "The use of non-hierarchical allocation methods for clustering large sets of data," *Australian Computer Journal*, vol. 19, 1987.

[27] S. Hirano and et. al., "Cluster analysis of time-series medical data based on the trajectory representation and multiscale comparison techniques," in *Data Mining, 2006. ICDM'06. Sixth International Conference on*. IEEE, 2006, pp. 896–901.