Effective Sub-Sequence-Based Dynamic Time Warping

Mohammed Alshehri[1,2], Frans Coenen[1], and Keith Dures[1]

[1] Department of Computer Science, University of Liverpool, Liverpool, UK
[2] Department of Computer Science, King Khalid University, Abha, Saudi Arabia
{M.A.Alshehri, Coenen, K.Dures}@liverpool.ac.uk

Abstract. k Nearest Neighbour classification techniques, where $k = 1$, coupled with Dynamic Time Warping (DTW) are the most effective and most frequently used approaches for time series classification. However, because of the quadratic complexity of DTW, research efforts have been directed at methods and techniques to make the DTW process more efficient. This paper presents a new approach to efficient DTW, the Sub-Sequence-Based DTW approach. Two variations are considered, fixed length sub-sequence segmentation and fixed number sub-sequence segmentation. The reported experiments indicate that the technique improvs efficiency, compared to standard DTW, without adversely affecting effectiveness.

Keywords: Time Series Analysis · Dynamic Time Warping · k-Nearest Neighbor Classification · Splitting Method.

1 Introduction

Over recent years there has been a significant increase in the amount of data that commercial enterprises and institutions collect. This has largely been as a consequence of technical advances. The data collected takes many forms; one such form is temporal data, specifically time series data [10]. In the field of data mining, much research, development and empirical experimentation have been conducted in the usage of time series data [15]. Time series data typically comprises a collection of values recorded chronologically, such as electrocardiogram (ECG) data [8], daily stocks prices [5] or daily temperatures [4]. However, the points do not have to be chronologically ordered; they can be ordered in some other way, for example, the outline of an object in an image [16]. In time series analysis, determination of the similarity between time series is a challenge [15]. One of the most frequently used similarity measurement techniques is Dynamic Time Warping (DTW).

DTW is founded on the idea of identifying an optimal alignment between two time series which may be of different lengths [12, 14]. Unlike Euclidean distance similarity measurement, DTW matches time series sequences by "warping" them in a nonlinear fashion (hence the name). DTW was first proposed in the context of speech recognition for the comparison of speech patterns [11]. Subsequently, it has been used for many other applications, for example, music analysis [6].

One of the main disadvantages of DTW is its quadratic complexity. DTW operates using a "distance matrix" measuring x^2, where x is the length of the two time series being compared (assuming they are of the same length). The time complexity of the DTW algorithm is therefore $O(x^2)$. This quadratic complexity therefore renders DTW to be impractical with respect to many application domains. The idea presented in this paper is to segment the time series into s sub-sequences of roughly equal size. The time complexity of the DTW then reduces to $O(\frac{x^2}{s})$; still quadratic but substantially less than $O(x^2)$. The first question to be answered is then how to define s; either as a fixed parameter or in terms of a predefined sub-sequence length. The second question is how to define s without any loss of functionality (consequent classification accuracy).

The rest of this paper is organised as follows. Section 2 presents a brief description of DTW. Section 3 considers some relevant previous work in the domain of DTW; the following Section 4, presents the proposed mechanism, Sub-Sequence-Based DTW. The time complexity is discussed in Section 5. Section 6 presents the evaluation strategy with an overview of the evaluation Datasets. Finally, Section 7 presents the main findings of the paper. A symbol table is given in Table 1 lists the symbols used throughout the paper.

Table 1: Symbol Table

Symbol	**Description**
p or q	A point in a time series described by a single value.
S	A time series such that $S = [p_1, p_2, \dots]$ or $S = [q_1, q_2, \dots]$.
x or y	Length of a given time series.
M	A distance matrix measuring $x \times y$.
$m_{i,j}$	The distance value at location i, j in M.
WP	A warping path $[w_1, w_2, \dots]$ where $w_i \in M$.
wd	A warping distance derived from WP.
U	A time series sub-sequence after segmentation. $U \subset S$.
ℓ	Length of time series sub-sequence after segmentation, $\ell < x$ and $\ell < y$.
$\mathcal{P}$	percentage (%) of the actual length of individual time series.
s	A number of sub-sequences into which a given time series is split.
C	A set of class labels $C = \{c_1, c_2, \dots\}$.
D	A collection of time series $\{S_1, S_2, \dots, S_r\}$
r	The number of time series in D.
z	The runtime (secs.) to process a single point p in the context of DTW.

2 Background

The DTW process can best be described by considering two time series $S_1 = [p_1, p_2, \dots, p_x]$ and $S_2 = [q_1, q_2, \dots, q_y]$, where x and y are the lengths of the two series respectively and $x, y \in \mathbb{N}$. The first step is to construct a "distance matrix"

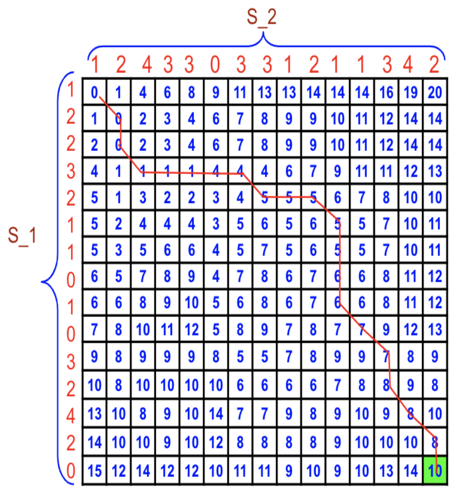

Fig. 1: Distance Matrix and Warping Path (line passes through cells) for the example time series S_1 and S_2 generated using standard DTW.

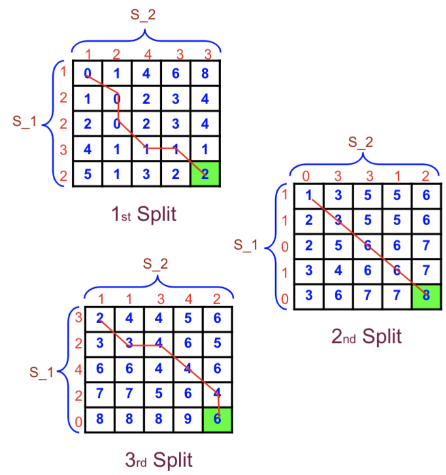

Fig. 2: Distance Matrices and Warping Paths (lines passes through cells) for the example time series S_1 and S_2 generated using sub-sequence splitting method.

M of size $x \times y$ where the value held at each cell $m_{i,j} \in M$ is the distance from point $p_i \in S_1$ to point $q_j \in S_2$. This distance is normally calculated in terms of Euclidean distance:

$$m_{i,j} = \sqrt{(p_i - q_j)^2} \tag{1}$$

An alternative might be absolute value distance calculation.

The distance matrix M is used to determine a minimum warping distance wd, which is then used as a similarity measure. A wd is a function of the minimum warping path, WP, from cell $m_{0,0}$ to cell $m_{x,y}$. A minimum warping path is thus a sequence of cell locations, $WP = [w_1, w_2, \ldots]$ in the matrix M, that minimises the warping distance. Given two time series $s_1 = [p_1, p_2, \ldots, p_x]$ of length $x \in \mathbb{N}$ and $s_2 = [q_1, q_2, \ldots, q_y]$ of length $y \in \mathbb{N}$, and using "Big O" notation, the complexity of DTW can be expressed as: $O(x \times y)$, or if $x = y$ $O(x^2)$. Thus DTW becomes computationally expensive when x and/or y are large [14].

From the foregoing, it can be seen that the operation of DTW is such that it meets the following conditions [13]:

1. **Monotonic condition**: The path will stay the same or increase. Both i and j indexes never decrease.
2. **Continuity condition**: The path continues one step at a time. Both i and j can only increase by 1 on each step along the path.
3. **Boundary condition**: The path starts at the bottom left $m_{(0,0)}$ and ends at the top right $m_{(x,y)}$.

The basic DTW process is illustrated in Figure 1. The figure shows the distance matrix M given two time series assuming two time series, $S_1 = [1, 2, 2, 3, 2, 1, 1, 0, 1, 0, 3, 2, 4, 2, 0]$ and $S_2 = [1, 2, 4, 3, 3, 0, 3, 3, 1, 2, 1, 1, 3, 4, 2]$. The minimum warping path is shown by the line passes through cells. The final warping distance arrived at is highlighted using a dark box in the corner.

3 Previous Work

This section details some related work that has been conducted to speed up DTW. These techniques are directed at limiting the number of distance matrix values to be calculated in the matrix M or at minimising the number of comparisons need to be considered. In other words by placing constraints on the matrix area to be considered when calculating a minimum warping distance. This is a different approach to that considered in this paper. To the best knowledge of the authors, the idea of sub-sequence splitting presented in this paper has not been previously reported in the literature.

An example of the approach where constraints have been placed on the matrix calculation found be found in Silva et al. [12]. Silva et al. proposed a method to speed up DTW known as PrunedDTW. The fundamental idea was to place upper bounds on the calculation process. The distances along the prime diagonal, from $m_{0,0}$ to $m_{x,y}$, are first calculated using the squared Euclidean distance. These are considered to be "upper bounds". Then for each point in the diagonal the distances along each row are calculated moving away from the diagonal, in row and column order, until a distance greater than the current upper bound is reached, further cells are "pruned" from the distance matrix. The result will be a pruned DTW which holds the minimum warping distance.

In the context of limiting the number of DTW comparisons with respect to time series classification where a new time series to be classified is compared to a bank of time series, Rakthanmanon et al. [9] reported on four different techniques for achieving this. The first promulgated the idea of early abandonment, stopping the warping path calculation if the wd value so far is equal to or larger than the best so far; otherwise, the new value is the best so far. The second considered the idea of reordering the time series in the "bank" so that the time series that are likely to be the most similar to the new time series are tested first so that the early abandonment process will result in less calculation than if the time series were not ordered in this way. One way of ordering time series is according to Euclidean distance similarity (much cheaper than DTW calculation). The third considered pruning time series that were unlikely to be a close match. One way of doing this is by using the lower bounding technique proposed in [7], the so called the LB_{Keogh} technique. This operates by superimposing a band, defined by a predefined offset value referred to as the lower bound, over each time series in the bank and calculating the complement of the overlap with the new time series. Where the calculated value exceeds a given threshold the associated time series is discounted. The fourth idea was directed at using a "cascading lower bound" where different lower bounds are considered to identify the bound most suitable for the dataset in question. Further work on lower bounding can be found in [3, 17].

4 Sub-Sequence-Based DTW

In this section, the proposed Sub-Sequence-Based DTW mechanism is presented. The fundamental idea of the proposed process is to divide (segment) the input time series (sequences) into sub-sequences. Then apply DTW to correlated sub-sequence pairs. Thus, given two time series S_1 and S_2, these would be divided into s sub-sequences so that we have $S_1 = [U_{1_1}, U_{1_2}, \ldots U_{1_s}]$ and $S_1 = [U_{2_1}, U_{2_2}, \ldots U_{2_s}]$. DTW is then applied to each sub-sequence paring U_{1_i}, U_{2_j} where $i = j$. The final minimum warping distance arrived at will then be the accumulated warping distance for each sub-sequence of s applications of DTW. Thus, returning to the example given in Figure 1, and assuming $s = 3$, there will be three subsequences in each time series of length $\ell = 5$, $S_1 = [U_{1_1}, U_{1_2}, U_{1_3}] = [[1, 2, 2, 3, 2], [1, 1, 0, 1, 0], [3, 2, 4, \ 2, 0]]$ and $S_1 = [U_{2_1}, U_{2_2}, U_{2_3}] = [[1, 2, 4, 3, 3], [0, 3, 3, 1, 2], [1, 1, 3, 4, 2]]$. Three distance matrices will result as shown in Figure2.

There are two mechanisms whereby s can be defined:

1. **Fixed Number**: The simplest is to specify s as a predefined parameter in which case the length of the individual time series sub-sequences, ℓ, will vary according to the input data; $\ell = \frac{x}{s}$. This may not be desirable.
2. **Fixed Length**: The alternative is to pre-specify the length of the desired time series sub-sequences, ℓ, in which case s will vary according to the input data, $s = \frac{x}{\ell}$. This may also not be desirable.

However, rigid implementation of s might not result in the best segmentation. Good points at which to cut the time series is where they meet, or at least at points where the distance between corresponding pairs of points is at a minimum. Thus a degree of *fuzziness* should be included to derive the best segmentation. This is defined by specifying a *tail*, t, measured backwards from ℓ, within which the cut should be applied; thus the cut will fall between $\ell - t$ and ℓ measured from the start of the time series on the first iteration and from the end of the previous segment on further iterations. The split point will be selected according to the minimum distance associated with the points from $\ell - t$ to ℓ. This is illustrated in Figure 3 which shows a "distance list" generated with respect to the two example time series given in Figure 1, and assumes $\ell = 10$ and $t = 3$; hence the selected split point is at index 9 (assuming the start of the sequence is index 1)

Having selected the split point there are three options as also illustrated in Figure 3: (i) include the split point value as the last value in the previous segment (Option A), (ii) include the split point value at the start of the following segment (Option B) or (iii) include the split point value in both the previous and following segments (Option C).

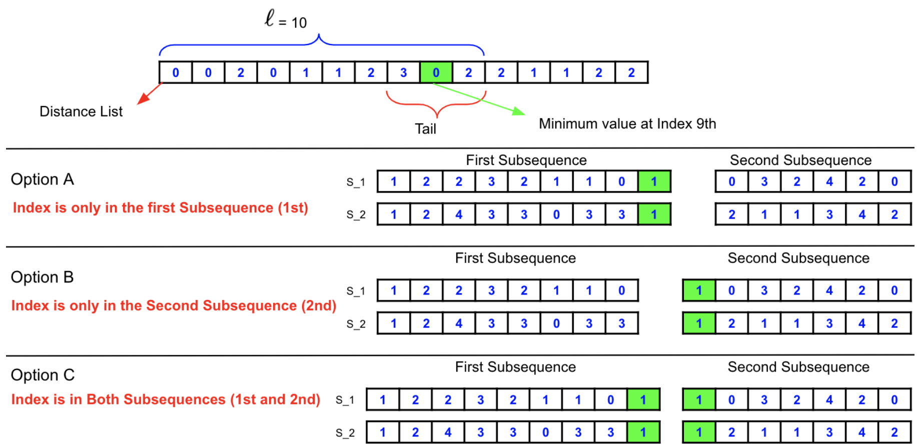

Fig. 3: Segmentation example given two time series S_1 and S_2, and Options A, B or C.

5 Time Complexity

The time complexity of the proposed mechanism is considered in this section. In time series classification, the complexity of the comparison between two time series, S_1 and S_2, when using Standard DTW, is given by $O(x \times y)$ where x and y are the lengths of S_1 and S_2 respectively. For the classification application under consideration, all the time series are of the same length, as in the case of the evaluation presented in the following section, this simplifies to:

$$DTW_{complexityStand} = O\left(x^2 \times z\right) \tag{2}$$

where z is a constant describing the time complexity associated with a single cell $m_{i,j}$ in the distance matrix M. The time complexity when using the proposed mechanism, Sub-Sequence-Based DTW, with fixed length segments ($DTW_{complexitySplit_{len}}$) and fixed number segments ($DTW_{complexitySplit_{num}}$) is then:

$$DTW_{complexitySplit_{len}} = O\left(\frac{x^2}{\frac{x}{\ell}} \times z\right) \tag{3}$$

$$DTW_{complexitySplit_{num}} = O\left(\frac{x^2}{s} \times z\right) \tag{4}$$

The most commonly used time series classification mechanism is k Nearest Neighbour (kNN) classification [1, 14] where a previously unseen time series is compared with a "data bank" of time series whose class label is known and the label for the new time series derived from the k most similar time series in the bank. The most commonly used value for k in time series analysis, using DTW, is $k = 1$, this we have 1NN. If we have a data repository with r examples the time complexity to classify a single record using 1NN is given by:

$$O\left(r \times DTW_{complexity}\right) \tag{5}$$

If there are t new time series to be classified ($t > 1$) the complexity is given by:

$$O\left(r \times DTW_{complexity} \times t\right) \tag{6}$$

In the case of cross-validation, as presented in the following section, the complexity becomes:

$$O\left(r \times DTW_{complexity} \times t \times numFolds\right) \tag{7}$$

When using ten cross validation the data set D is split into tenths, in which case $r = \frac{9 \times |D|}{10}$, $t = \frac{|D|}{10}$ and the number of fold will equal 10:

$$O\left(\frac{9 \times |D|}{10} \times DTW_{complexity} \times \frac{|D|}{10} \times 10\right) \tag{8}$$

Which simplifies to:

$$O\left(\frac{9 \times |D|^2}{100} \times DTW_{complexity}\right) \tag{9}$$

6 Evaluation

In this section, the evaluation of the proposed Sub-Sequence-Based DTW mechanism is presented. This mechanism was used in connection with the 1NN classification and the ten selected datasets from the UEA-UCR Time Series Classification repository [2]. Further detail concerning the selected data sets is given in Sub-section 6.1. The objectives of the evaluation were:

1. To compare the operation of fixed length and fixed number Sub-Sequence-Based DTW.
2. To determine the most suitable value for t, the sub-sequence tail.
3. To analyse the runtime of the proposed Sub-Sequence-Based DTW mechanism.
4. To evaluate the classification effectiveness of the proposed approach in comparison with Standard DTW (using accuracy and F1-score as the evaluation metrics).

Each is considered with respect to the results obtained in Sub-section 6.2 below. For each set of experiments Ten Cross Validation (TCV) was adopted. A desktop computer with a 3.5 GHz Intel Core i5 processor and 16 GB, 2400 MHz, DDR4 of primary memory was used throughout.

6.1 Data sets

In this subsection, a brief overview of the evaluation datasets is presented. Some statistics concerning the ten datasets are given in Table 2. Column five represents the overall size of each dataset calculated using $x \times r$, where x is the length (number of points) of an individual time series and r is the number of time series (records) in each dataset D; the significance is that this is a good measure of the overall size of a time series data set. The nature of the data (time series) collected for each dataset is represented by its type (column seven). Motion indicates that the time series represents body movements, Spectro means that the time series comprises spectrograph data, Sensor that the time series data were collected using sensors (such as an electric power signal sensor), Simulation means that time series data was collected using some form of simulation and image means image segmentation translated into a time series form.

Table 2: Time Series Datasets Used for Evaluation Purposes.

ID No.	Dataset	Length (x)	Num. records (r)	Size x_r	Num. Classes	Type
1	GunPoint	150	200	30000	2	Motion
2	OliveOil	570	60	34200	4	Spectro
3	Trace	275	200	55000	4	Sensor
4	ToeSegment2	343	166	56938	2	Motion
5	Car	577	120	69240	4	Sensor
6	Lightning2	637	121	77077	2	Sensor
7	ShapeletSim	500	200	100000	2	Simulated
8	DiatomSizeRed	345	322	36000	4	Image
9	Adiac	176	781	137456	37	Image
10	HouseTwenty	2000	159	318000	2	Image

6.2 Evaluation Results

To compare the operation of fixed length Sub-Sequence-Based DTW with fixed number Sub-Sequence-Based DTW, two sets of experiments were conducted. The first considered the parameters ℓ and s required by the two mechanisms using Option A and $t = 0$. A range of values for ℓ was considered from 10 to 50 increasing in steps of 10, $\ell = \{10, 20, 30, 40, 50\}$. For the parameter s this was defined in terms of a percentage of the overall length of the overall input time series length from 5% to 25%, $\{5\%, 10\%, 15\%, 20\%, 25\%\}$, which translated to $s = \{20.00, 10.00, 6.67, 5.00, 4.00\}$. Note that wherever an exact segmentation of

Table 3: Average results of TCV classification accuracy (Acc) and F1 Scores (F1), and run times (sec), obtained over 10 evaluation datasets using a range of ℓ and s values, and Option A and $t = 0$, compared with standard DTW; best results with respect to the proposed Sub-sequence Based DTW mechanism highlighted in bold font.

Fixed Length ℓ	**Avg Acc**	**Avg F1**	**Avg Run-Time (sec)**
Standard	88.15	0.87	115.46
10	85.02	0.86	16.76
20	86.36	0.87	15.90
30	86.98	0.88	16.11
40	**87.78**	**0.88**	**17.15**
50	87.39	0.88	17.91

Fixed Number s	**Avg Acc**	**Avg F1**	**Avg Run-Time (sec)**
Standard	88.15	0.87	115.46
5%	87.96	0.87	18.79
10%	88.59	0.88	21.90
15%	88.25	0.88	24.94
20%	**89.24**	**0.89**	**31.16**
25%	89.16	0.89	35.04

Table 4: Fixed Length $\ell = 40$: Accuracy, F1-Score and Runtime Results (Option = A and $t = 0$).

Dataset	**Acc (SD)**	**F1 (SD)**	**Runtime (Secs)**
Gun Point	99.47 (0.01)	0.99 (0.02)	5.76
Olive Oil	90.95 (0.12)	0.09 (0.14)	1.61
Trace	97.50 (0.03)	98 (0.04)	5.98
ToeSegmentation2	89.13 (0.05)	0.89 (0.06)	6.69
Car	83.33 (0.09)	0.82 (0.10)	4.40
Lighting2	81.54 (0.09)	0.80 (0.11)	5.00
Shapelet Sim	89.97 (0.06)	0.90 (0.06)	11.48
DiatomSize Reduction	100 (0.00)	1.00 (0.00)	20.50
Adiac	64.42 (0.04)	0.61 (0.04)	94.55
House Twenty	93.71 (0.04)	0.94 (0.05)	18.86

Table 5: Fixed Number $s = 20\%$: Accuracy, F1-Score and Runtime Results (Option = A and $t = 0$).

Dataset	**Acc (SD)**	**F1 (SD)**	**Runtime (Secs)**
Gun Point	95.50 (0.04)	0.96 (0.05)	6.39
Olive Oil	89.52 (0.15)	0.89 (0.16)	2.46
Trace	97.00 (0.04)	97 (0.04)	6.96
ToeSegmentation2	89.17 (0.05)	0.88 (0.07)	8.38
Car	82.50 (0.07)	0.82 (0.8)	7.70
Lighting2	90.32 (0.08)	0.90 (0.9)	9.66
Shapelet Sim	88.89 (0.07)	0.89 (0.07)	18.46
DiatomSize Reduction	99.68 (0.01)	0.99 (0.01)	27.05
Adiac	65.45 (0.03)	0.62 (0.04)	96.29
House Twenty	94.38 (0.05)	0.95 (0.05)	128.07

Table 6: Average results of TCV classification accuracy (Acc) and F1 Scores (F1), obtained over 10 evaluation datasets using a range of values for t, all three options and the fixed length variation with $\ell = 40$; best result in bold font.

t	Option A Acc (SD)	Option A F1 (SD)	Option B Acc (SD)	Option B F1 (SD)	Option C Acc (SD)	Option C F1 (SD)
2	88.63	0.89	89.74	0.89	**90.03**	**0.90**
3	89.19	0.89	89.35	0.89	89.48	0.89
4	89.69	0.89	89.26	0.89	89.17	0.89
5	89.36	0.89	88.58	0.88	88.43	0.88
6	88.97	0.88	89.21	0.89	87.68	0.88
7	88.58	0.88	88.83	0.88	88.26	0.88
8	88.67	0.88	88.84	0.88	88.19	0.88
9	89.22	0.89	88.15	0.88	88.22	0.88
10	88.53	0.88	88.02	0.87	88.14	0.88

Table 7: Average results of TCV classification accuracy (Acc) and F1 Scores (F1), obtained over 10 evaluation datasets using a range of values for t, all three options and the fixed number variation with $s = 20\%$; best result in bold font.

t	Option A Acc (SD)	Option A F1 (SD)	Option B Acc (SD)	Option B F1 (SD)	Option C Acc (SD)	Option C F1 (SD)
2	89.18	0.89	88.88	0.88	**89.27**	**0.89**
3	89.22	0.89	87.81	0.88	88.71	0.88
4	88.59	0.88	88.64	0.88	88.18	0.88
5	89.01	0.89	88.75	0.88	88.79	0.88
6	88.77	0.88	88.69	0.88	88.83	0.88
7	89.12	0.89	88.65	0.88	88.84	0.88
8	89.19	0.89	88.75	0.88	88.86	0.88
9	88.51	0.88	89.00	0.89	89.20	0.89
10	89.01	0.88	88.64	0.88	88.84	0.88

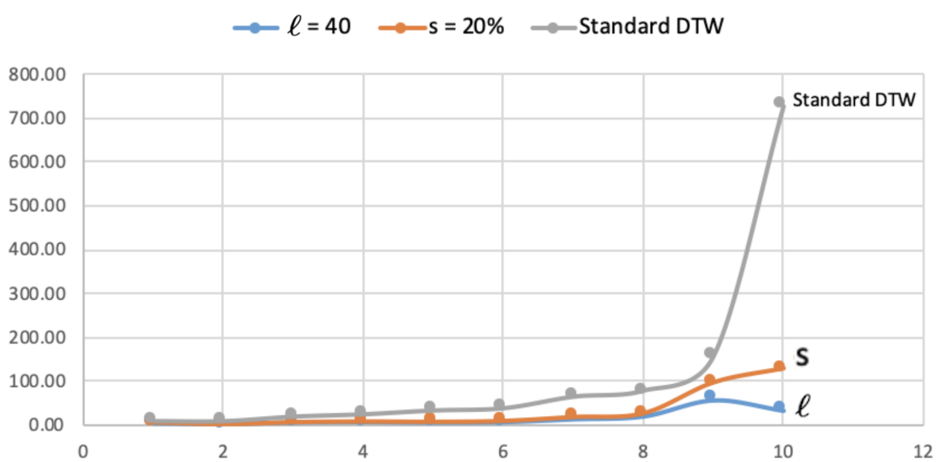

Fig. 4: Run time results (seconds) using best performing parameters for Sub-Sequence-Based DTW compared to Standard DTW.

a time series could not be achieved a “short” segment was included at the end of the segment collection. A summary of the results obtained is presented in Table 3, best F1 scores with respect to the proposed mechanism are highlighted in bold font. From the table it can be seen that best classification results were obtained using $\ell = 40$ and $s = 20\%$ ($s = 5$). Note also that the recorded runtimes for the proposed mechanism were significantly less than that required by “standard” DTW. More detailed results concerning $\ell = 40$ and $s = 20\%$ settings are given in Tables 4 and 5; where the numbers in parentheses are the standard deviation in each case.

Table 8: Fixed Length: Accuracy, F1-Score and Runtime Results ($\ell = 40$, Option $= C$ and $t = 2$), compared to standard DTW.

Dataset	Fixed Length			Standard DTW		
	Acc (SD)	**F1 (SD)**	**Runtime (Secs)**	**Acc (SD)**	**F1 (SD)**	**Runtime (Secs)**
GunPoint	99.00 (0.02)	0.99 (0.02)	3.11	93.97 (0.04)	0.94 (0.05)	8.11
OliveOil	90.12 (0.10)	0.89 (0.12)	1.43	89.52 (0.15)	0.88 (0.16)	8.06
Trace	96.50 (0.04)	97.00 (0.04)	4.94	99.00 (0.03)	99.00 (0.03)	18.41
ToeSegmentation2	92.26 (0.03)	0.92 (0.04)	5.85	89.07 (0.09)	0.88 (0.10)	23.81
Car	82.50 (0.10)	0.81 (0.11)	4.86	80.83 (0.07)	0.80 (0.9)	32.45
Lighting2	87.40 (0.08)	0.87 (0.9)	6.10	87.74 (0.09)	0.87 (0.8)	37.69
ShapeletSim	93.00 (0.04)	0.93 (0.04)	12.79	82.37 (0.09)	0.81 (0.011)	64.02
DiatomSizeReduction	100 (0.00)	1.00 (0.00)	19.54	99.36 (0.01)	0.99 (0.01)	77.91
Adiac	64.98 (0.03)	0.62 (0.04)	55.64	64.63 (0.03)	0.62 (0.04)	156.81
HouseTwenty	91.17 (0.07)	0.91 (0.07)	32.68	95.00 (0.03)	0.95 (0.05)	727.47
Average	**90.03**	**0.90**	**14.99**	**88.15**	**0.87**	**115.47**

The second set of experiments used $\ell = 40$ and $s = 20\%$, a range of values for t, $\{2, 3, 4, 5, 6, 7, 8, 9, 10\}$ and considered all three options A, B and C (see Sub-section 4). The results are presented in Tables 6 and 7 (standard deviations are given in parentheses). The results clearly indicate that, regardless of whether the fixed length or fixed number variation of the proposed mechanism is used, best results are obtained when $t = 2$ and Option $= C$.

Considering the recorded runtimes, from the tables it is clear that the proposed Sub-Sequence-Based DTW is faster than standard DTW without loss of effectiveness, in fact with a slight improvement in effectiveness in terms of the F1 measure. This is emphasised by the graph presented in Figure 4 which shows the recorded runtimes using the best performing parameters with respect to each variation $ell = 40$ and $s = 20\%$, and $t = 2$ and Option $= C$, and the recorded runtime using standard DTW. In the figure, the x-axis records the identification number of the relevant dataset (see Table 2) and the y-axis the runtime in seconds.

Overall there is also little to choose between the two variations of the proposed Sub-sequence Based DTW mechanism, although an argument could be made in favour of the fixed length variation. A more detailed breakdown of the results obtained using this variation with $\ell = 40$, $t = 2$ and Option $= C$ is therefore given in Table 8.

7 Conclusion

In this paper the Sub-Sequence-Based Dynamic Time Warping (DTW) mechanism has been presented, a mechanism for speeding up the DTW process without entailing approximations. The proposed mechanism has two variations for defining the number of sub-sequences (segments) into which time series should be divided, fixed length which uses a parameter ℓ and fixed number which uses a parameter s (defined in the form of a percentage of time series length). To determine the actual "split point" a third parameter t was used to define the area at the end of a potential sub-sequence where a split should take place, the idea was to choose whatever index featured the least difference in amplitude between the two time series considered. Having identified the split point there were three options as to where the split index value should be allocated: the end of the preceding sub-sequence (Option A), the start of the following sub-sequence (Option B) or both (Option C). Experiments were conducted that considered these different parameter settings, variations and options by considering a 1NN classification scenario. It was found that best results were obtained when $\ell = 40$, $s = 20\%$, $t = 2$ and using Option C. There was little to choose between the two variations, fixed length or fixed number, however, an argument could be made in favour of the fixed length variation. Whatever the case, both variations outperformed standard DTW in terms of run time by a considerable margin, with no detriment to the recorded accuracy.

References

1. Bagnall, A., Lines, J.: An experimental evaluation of nearest neighbour time series classification. arXiv preprint arXiv:1406.4757 (2014)
2. Bagnall, A., Lines, J., Bostrom, A., Large, J., Keogh, E.: The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Mining and Knowledge Discovery **31**(3), 606–660 (2017)

3. Bringmann, K., Künnemann, M.: Quadratic conditional lower bounds for string problems and dynamic time warping. In: 2015 IEEE 56th Annual Symposium on Foundations of Computer Science. pp. 79–97. IEEE (2015)
4. Byakatonda, J., Parida, B., Kenabatho, P.K., Moalafhi, D.: Analysis of rainfall and temperature time series to detect long-term climatic trends and variability over semi-arid botswana. Journal of Earth System Science **127**(2), 25 (2018)
5. Chen, M.Y., Chen, B.T.: A hybrid fuzzy time series model based on granular computing for stock price forecasting. Information Sciences **294**, 227–241 (2015)
6. Deng, J.J., Leung, C.H.: Dynamic time warping for music retrieval using time series modeling of musical emotions. IEEE Transactions on Affective Computing **6**(2), 137–151 (2015)
7. Keogh, E., Ratanamahatana, C.A.: Exact indexing of dynamic time warping. Knowledge and information systems **7**(3), 358–386 (2005)
8. Phinyomark, A., Scheme, E.: An investigation of temporally inspired time domain features for electromyographic pattern recognition. In: 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). pp. 5236–5240. IEEE (2018)
9. Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., Zakaria, J., Keogh, E.: 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. pp. 262–270. ACM (2012)
10. Roberts, D.R., Bahn, V., Ciuti, S., Boyce, M.S., Elith, J., Guillera-Arroita, G., Hauenstein, S., Lahoz-Monfort, J.J., Schröder, B., Thuiller, W., et al.: Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. Ecography **40**(8), 913–929 (2017)
11. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE transactions on acoustics, speech, and signal processing **26**(1), 43–49 (1978)
12. Silva, D.F., Batista, G.E.: Speeding up all-pairwise dynamic time warping matrix calculation. In: Proceedings of the 2016 SIAM International Conference on Data Mining. pp. 837–845. SIAM (2016)
13. Silva, D.F., Giusti, R., Keogh, E., Batista, G.E.: Speeding up similarity search under dynamic time warping by pruning unpromising alignments. Data Mining and Knowledge Discovery pp. 1–29 (2018)
14. Tan, C.W., Herrmann, M., Forestier, G., Webb, G.I., Petitjean, F.: Efficient search of the best warping window for dynamic time warping. In: Proceedings of the 2018 SIAM International Conference on Data Mining. pp. 225–233. SIAM (2018)
15. Wang, X., Mueen, A., Ding, H., Trajcevski, G., Scheuermann, P., Keogh, E.: Experimental comparison of representation methods and distance measures for time series data. Data Mining and Knowledge Discovery **26**(2), 275–309 (2013)
16. Wegner Maus, V., Câmara, G., Appel, M., Pebesma, E.: dtwsat: Time-weighted dynamic time warping for satellite image time series analysis in r. Journal of Statistical Software **88**(5), 1–31 (2019)
17. Zhou, M., Wong, M.H.: Boundary-based lower-bound functions for dynamic time warping and their indexing. Information Sciences **181**(19), 4175–4196 (2011)