Candidates Reduction and Enhanced Sub-Sequence-Based Dynamic Time Warping: A Hybrid Approach

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. Dynamic Time Warping (DTW) coupled with k Nearest Neighbour classification, where $k = 1$, is the most common classification algorithm in time series analysis. The fact that the complexity of DTW is quadratic, and therefore computationally expensive, is a disadvantage; although DTW has been shown to be more accurate than other distance measures such as Euclidean distance. This paper presents a hybrid, Euclidean and DTW time series analysis similarity metric approach to improve the performance of DTW coupled with a candidate reduction mechanism. The proposed approach results in better performance than alternative enhanced Sub-Sequence-Based DTW approaches, and the standard DTW algorithm, in terms of runtime, accuracy and F1 score.

Keywords: Time Series Analysis, Dynamic Time Warping, K-Nearest Neighbour Classification, Sub-Sequence-Based DTW, Candidate Reduction.

1 Introduction

A time series is a collection of sequentially recorded numeric values. Example application domains include stock market analysis [8] and meteorological forecasting [12]. A typical category of application is time series classification. Many techniques have been used for time series classification, examples include: Decision Trees [6], Support Vector Machines (SVM) [10] and Artificial Neural Networks [7]. However, the most frequently used classification mechanism is the k Nearest Neighbour (kNN) mechanism [11, 16, 17]. The most frequently used value for k is $k = 1$ because it has been shown to work well [14], and because it avoids the need for a conflict resolution mechanism (required when $k > 1$).

Whatever classification mechanism ends up being used, both the building of the classification model and the eventual utilisation of the model entail a significant amount of time series similarity checking. This is especially the case given long time series. Selection of an appropriate similarity checking mechanism is, therefore, an important part of the classification process [14]. Frequently sited similarity checking mechanisms include: Euclidean Distance (ED) and Dynamic Time Warping (DTW). DTW, it can be argued, is more accurate and does not require time series to be of the same length. ED, in turn, tends to be faster; this is particularly the case given very long time series. A hybrid approach, therefore, seems like a credible alternative.

In [2] the Sub-Sequence-Based DTW mechanism was proposed, a mechanism designed to speed up the DTW process without adversely affecting effectiveness. The fundamental idea was to divide the time series to be compared into sets of equal-sized subsequences of length ℓ, and then to first compute DTW values for corresponding individual subsequences in a pair of given time series, before deriving an overall DTW similarity measure. This produced some good results, outperforming standard DTW. In [1] it was hypothesised that the equal-sized approach advocated in [2] was not the most appropriate approach and that a certain amount of "fuzziness" should be introduced into the process. More specifically it was proposed that the "cut point" should be wherever the two time series converged, although limited to a range of points defined by a tail parameter t measured backwards from ℓ. However, in [2], the values for ℓ and t were predefined. Building on the work presented in [2] and [1], this paper proposes a variation of the Sub-Sequence-Based DTW mechanism that includes: (i) a novel mechanism for reducing the KNN search space by first applying the computationally less expensive ED measure and then applying DTW to $\mathfrak{r}$ retained time series (in other words a hybrid ED-DTW approach), and (ii) optimisation of the parameter t, ℓ and $\mathfrak{r}$. The proposed mechanism is fully described and evaluated using 15 time series datasets taken from the UEA and UCR (University of East Anglia and University of California Riverside) Time Series Classification Repository [4].

The remainder of this paper is organised as follows. Some background and a review of related work are presented in Section 2. The operation of the proposed hybrid Sub-Sequence-Based DTW mechanism is then presented in Section 3. The theoretical computational complexity of the proposed approach is presented in Section 4. The evaluation of the proposed mechanism is then presented in Section 5, together with a discussion of the results obtained. The paper is concluded in Section 6. For convenience, a symbol table is given in Table 1 listing the symbols frequently used throughout this paper.

2 Background and Previous Work

As noted in the introduction to this paper, the most common similarity measures used in time series analysis are Euclidean Distance (ED) and Dynamic Time Warping (DTW). Euclidean distance has been widely used in time series analysis applications to measure the similarity between time series in terms of the distance between corresponding points within pairs of time series. Given two time series $S = [p_1, p_2, \ldots, p_x]$ and $S = [q_1, q_2, \ldots p_x]$, both of length x, ED similarity d_E is measured as shown in Equation 1, the square root of the sum of the squares of the differences between corresponding points in the two time

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 $S = [p_1, p_2, \ldots]$ ($S = [q_1, q_2, \ldots]$), $S \in D$.
x or y	The 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 $[w_1, w_2, \ldots]$ where $w_i \in M$.
wd	A warping distance derived from WP.
ℓ	The number of points in a subsequence.
s	A number of sub-sequences into which a given time series is to be split.
C	A set of class labels $C = \{c_1, c_2, \ldots\}$.
D	A collection of time series $\{S_1, S_2, \ldots, S_r\}$
r	The number of time series in in a dataset D.
t	The tail measured backwards from ℓ within which the cut is to be applied; thus given $S = [p_0, \ldots, p_\ell]$ the cut will fall between p_ℓ and $p_{\ell-t}$.
w	A time series subsequence $\{p_i, p_{i+1}, \ldots\}$, such that $w \in S$
W	A set of s time series subsequences, $\{w_1, w_2, \ldots w_s\}$ contained in a given time series S
$\mathfrak{r}$	The number of candidates selected from a dataset.
$\mathfrak{t}$	A new previously unseen time series.

series [11]. ED similarity calculation offers the advantage, over DTW, that it is fast; its weakness is in terms of classification accuracy. Moreover, it only works with time series of the same length [14].

$$d_E = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \quad (1)$$

DTW was originally directed at speech recognition applications [15]. The idea was to find the minimum warping distance (wd) between two time series in non-linear alignment. The process of DTW can be described as follows. Given two time series $S_1 = [p_1, p_2, \ldots, p_x]$ and $S_2 = [q_1, q_2, \ldots, q_y]$ a distance matrix M of size $x \times y$ will be constructed such that the value held at each cell $m_{i,j}$ is the distance calculated using Equation 1, between the corresponding points [1]. In other words, the distance value assigned to $m_{i,j}$ is the summation of $d_{i,j}$ and the minimum cumulative distance value held at one of the three "previous" cells to $m_{i,j}$ [13]. At the end of the process, the minimum warping distance (wd) will be held at $m_{x,y}$. Even though DTW has quadratic time complexity, it tends to perform better than ED in term of accuracy; and offers the additional advantage that it works with time series of different length. Figure 1 gives examples of both similarity measurements (taken from [14]).

$$m_{i,j} = d_{i,j} + min\{m_{i-1,j}, m_{i,j-1}, m_{i-1,j-1}\} \quad (2)$$

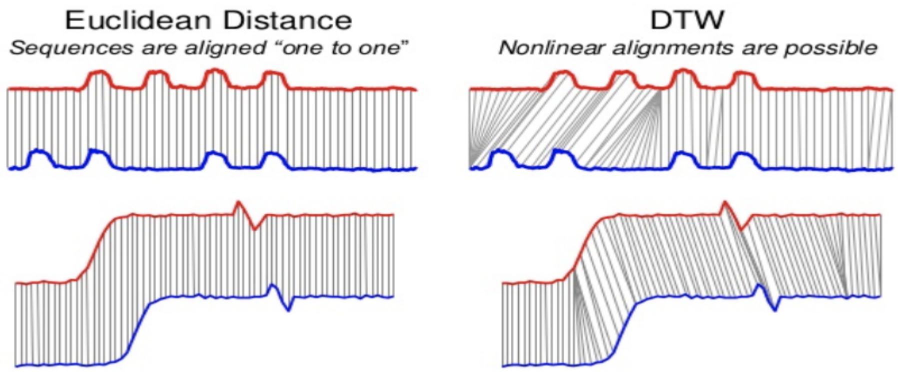

Fig. 1: Example of Euclidean Distance (left) and Dynamic Time Warping (right) [14].

Given the time complexity of DTW, a number of mechanisms have been proposed to address this complexity. This work can be categorised as either being directed at limiting the number of values to be calculated to construct the matrix M, or at limiting the number of comparisons to be considered. Examples of the first can be found in [9, 13, 15, 16], examples of the second can be found in [5, 14, 18]. In [11] an investigation was reported that considered a number of different distance measures (Euclidean distance, Normalised Euclidean distance, Manhatten distance and Canberra distance) for calculating the values to be held in M. Experiments were done for each distance measure using ten datasets and k-Nearest Neighbour Classification. The results demonstrated that Euclidean distance was the most appropriate distance measure to be used to build DTW distance matrices.

The Sub-Sequence-Based DTW idea, first proposed in [2], took a different approach that did not fit well with the above categorisation. As noted in the introduction to this paper, the main idea was to segment each time series into a predefined set of s sub-sequence and apply the DTW process to corresponding pairs of sub-sequences before deriving an overall DTW similarity value. Thus, given two time series S_1 and S_2, these would be divided into s sub-sequences so that we have $S_1 = [U_{1_1}, U_{1_2}, \dots U_{1_s}]$ and $S_1 = [U_{2_1}, U_{2_2}, \dots U_{2_s}]$. 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. This mechanism was shown to improve the DTW calculation runtime significantly compared to alternative approaches, especially given very long time series. However, the fixed sub-sequence size advocated in [2] was conjectured to be a disadvantage with respect to the accuracy of the approach. An enhanced Sub-Sequence-Based DTW mechanism was therefore proposed in [1]. The fundamental idea of the improved mechanism was to find the most appropriate size for s by utilising two parameters: the maximum length of a sub-sequences ℓ and a tail t, measured backwards from ℓ, within which the cut was to be applied. Thus given $S = [p_0, \dots, p_\ell]$ the cut will fall between p_ℓ and $p_{\ell-t}$. Consideration was also given to whether the split point should be included in the first sub-sequence only, in both subsequences or the second subsequences only, Split Point Allocation Options (SPAO) A, B and C respectively as illustrated in Fig 2. Option C provided the best performance and was therefore used with respect to the evaluation presented later in this paper.

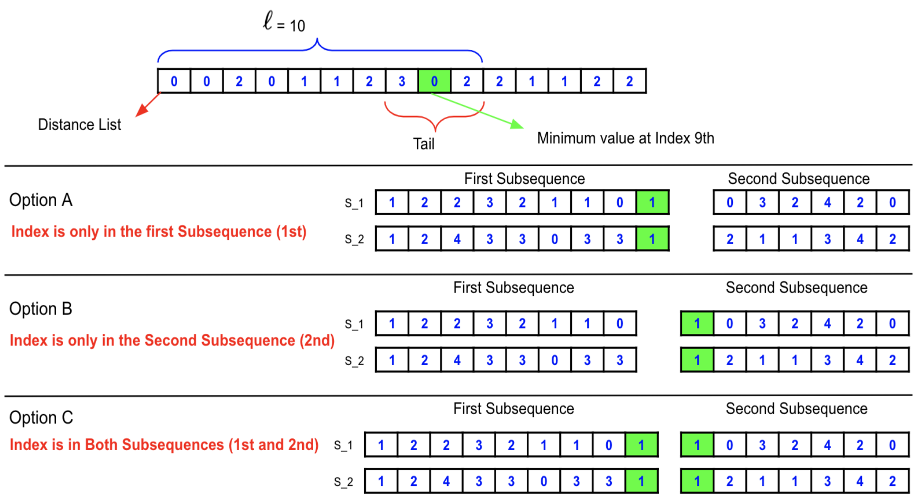

Fig. 2: Segmentation examples given two time series S_1 and S_2, and SPAO options A, B or C [1].

3 Enhanced Sub-Sequence-Based DTW

A block-diagram outlining the proposed Sub-Sequence-Based DTW process is presented in Figure 3. The process commences with a Database D of r time series $D = \{S_1, S_2, \ldots, S_r\}$. The first stage is to identify the most appropriate values for the parameters ℓ, t and Γ; rather than adopting the parameter pre-specification approach advocated in [2] and [1]. We thus have a three-dimensional search space $|I| \times |T| \times |R|$ where $I = \{\ell_1, \ldots, \ell_{|I|}\}$, $T = \{t_1, \ldots, t_{|T|}\}$ and $R = \{\Gamma_1, \ldots, \Gamma_{|R|}\}$. Preliminary experiments indicated that, typically, there was no global "peak" in this space, but instead many local maxima with, again typically, one that was better than the rest. This precluded any form of "hill-climbing" strategy. An exhaustive search strategy was therefore adopted. For the evaluation discussed in the following section, F1 score was used as the parameter to be maximised.

Once the most appropriate parameters settings have been identified, given a particular application domain as represented by the time series in the database, these can be used to translate the time series in the database so that each $S_i \in D$

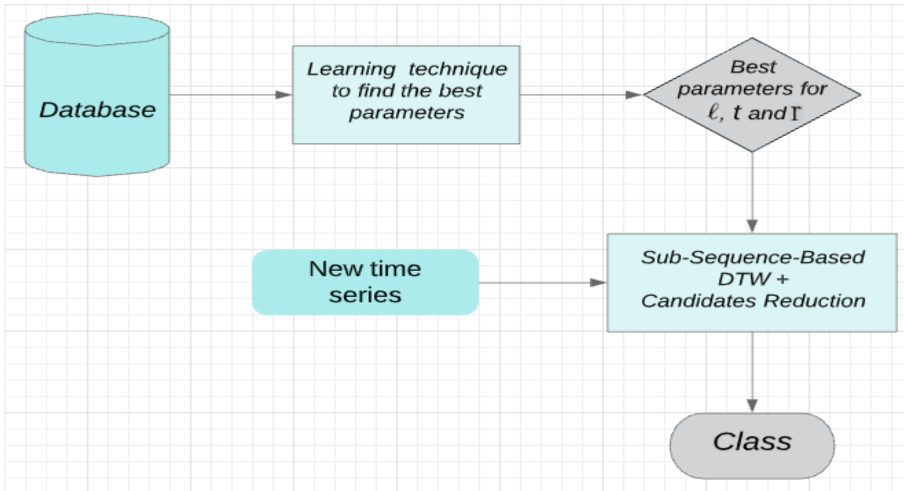

Fig. 3: Proposed Sub-Sequence-Based DTW Process with parameter optimisation and a hybrid ED-DTW similarity mechanism

is represented as a set W_i of s time series subsequences $W_i = \{w_1, w_2, \ldots w_s\}$. The sub-sequence represented time series then form a kNN "bank", each associated with a class label, with which to label a previously unseen time series $\mathfrak{t}$. Note that $\mathfrak{t}$ is first recast, using the learnt parameter values ℓ and t, so that it also comprises a set of subsequences. However, instead of comparing $\mathfrak{t}$ to every set of subsequences in the kNN bank the idea is to first reduce the search space by applying a "candidate reduction" process founded on ED similarity measurement. The motivation here is that ED outperforms DTW in terms of runtime. Using this approach $\mathfrak{r}$ candidates were retained to which DTW was applied, because it had been shown to be more accurate than ED.

4 Time Complexity

In this section, the time complexity of the proposed mechanism is presented. In time series classification, when using standard DTW, the complexity of the comparison between two time series, S_1 and S_2 is dependent on the size of the distance matrix M. The time complex thus is given by $O(x \times y)$ where x and y are the lengths of S_1 and S_2 respectively [1]. For the experiments reported in the evaluation section below, each evaluation data set featured time series of the same length, the calculation of standard DTW complexity, $DTW_{compStand}$, thus simplifies to:

$$DTW_{compStand} = O\left(x^2\right) \tag{3}$$

When using ED the complexity, in terms of the number of similarity calculations, will be:

$$ED_{compStand} = O\left(x\right) \quad (4)$$

When using the Enhanced Sub-Sequence-Based DTW, proposed in this paper, the DTW time complexity reduces to:

$$DTW_{compSplit} = O\left(\frac{x^2}{x \div \ell}\right) \quad (5)$$

As note earlier, kNN classification was used with respect to the evaluation reported below, with $k = 1$ because this is the most commonly used value for k for time series classification [2, 3], the new unseen time series t needs to be compared with all records r in the dataset D in order to be classified. The time complexity for comparing a single record using 1NN will be:

$$O\left(r \times complexity\right) \quad (6)$$

where complexity can be measured using either: (i) $DTW_{compStand}$, (ii) $ED_{compStand}$ or (iii) $DTW_{compSplit}$.

If there are $|\mathsf{t}|$ new time series to be classified ($|\mathsf{t}| > 1$) the complexity will become:

$$O\left(r \times complexity \times |\mathsf{t}|\right) \quad (7)$$

When using the proposed, Candidate Reduction, the time complexity is given by:

$$O\left(|\mathsf{t}| \times \left(\left(\mathsf{r} \times DTW_{compSplit}\right) + \left(\left(r - \mathsf{r}\right) \times ED_{compStand}\right)\right)\right) \quad (8)$$

where r is the number of time series retained after candidate reduction.

5 Evaluation

In this section, the evaluation of the proposed mechanism is presented. Experiments were conducted using: (i) Standard DTW (the benchmark), (ii) Enhanced Sub-Sequence-Based DTW as described in [1] and using $\ell = 40$ and $t = 2$ and Option C (the parameters that produced the best results), (iii) Enhanced Sub-Sequence-Based DTW with parameter learning but without candidate reduction and (iv) Enhanced Sub-sequence-Based DTW with parameter learning and candidate reduction (the proposed mechanism). As noted earlier, the evaluation was performed using the kNN classification algorithm. Fifteen datasets from the UEA and UCR Time Series Classification repository [4] were used. Further detail regarding the datasets used is given in Sub-section 5.1. The evaluation objectives were:

1. To review the operation of the proposed Enhanced Sub-sequence-Based DTW with parameter learning and candidate reduction in terms of the parameters used

2. To evaluate the run-time of the proposed approach in comparison with the other approaches considered.
3. To evaluate the accuracy of the proposed approach in comparison with the other approaches considered.

Each is considered in turn in the following three sub-sections, Sub-section 5.1 to Sub-section 5.3.

For the experiments, a desktop computer with a 3.5 GHz Intel Core i5 processor and 16 GB, 2400 MHz, DDR4 of primary memory was used. The reported values are Ten Cross Validation (TCV) average values.

5.1 Parameter Settings

This section presents an overview of the data sets used and the learnt parameter settings. A total of fifteen datasets were taken form the UEA and UCR repository [4]. The datasets were chosen so that a range of datasets of different sizes, in terms of the number of records and time series lengths, were considered and different numbers of classes. The lengths varied between 8 and 2000 points; the number of records varied between 60 to 10992. An overview of the fifteen datasets, ordered according to ascending order of time series length (x), is given in Table 2. Column 3, x, gives the time series length (number of points) for each dataset. The number of records r for each dataset is given in Column 4, and the number of classes in Column 5. The parameter values learnt using the proposed Sub-Sequence-Based DTW approach, for ℓ, t and ɼ, are given in Columns 6, 7 and 8 respectively. The runtime to learn the parameters is given in Column 9. From the table, it can be seen that each dataset has its own values for the parameters; although it should be noted that $\ell = 40$ is the best length for almost 50% of the datasets used in the experiments. For runtime, as expected, the size of the dataset $(x \times r)$ plays an important role to determine the required time for learning the parameters.

5.2 Run Time performance

In this sub-section, the runtime performance with respect to the labelling of a single previously unseen time series is presented. The runtime results are presented in Table 3; again, the data sets are ordered according to ascending order of time series length. The columns equate to the four alternatives considered in the evaluation as listed earlier. From the table, it can be seen that using the proposed approach, with parameter learning and candidate reduction, significant efficiency gains are earned. The runtimes recorded using the proposed approach are almost the same. The same results are plotted in Figure 4, where the x-axis represents the dataset names, see Table 2, and the y-axis represents the runtime in seconds. From the figure, it can again be seen that the runtime using the proposed mechanism (red line) is almost constant regardless of the nature of the data set used. Also, the runtime for the Enhanced Sub-Sequence-Based DTW and Enhance parameter Learning are almost identical.

Table 2: Evaluation Time Series Datasets Used and Learnt Parameters ℓ, t and ɼ.

ID No.	Dataset Name	Length (x)	Num. records (r)	Num. Classes	Parameters ℓ	Parameters t	Parameters ɼ	Runtime (sec)
1.	**PenDigits**	8	10992	10	5	2	1	8867
2.	**SmoothSubspace**	15	300	3	7	3	20	236
3.	**ItalyPowerDemand**	24	1096	2	10	4	5	1131
4.	**Libras**	45	360	15	10	2	5	886
5.	**SyntheticControl**	60	600	10	30	2	32	2187
6.	**GunPoint**	150	200	2	40	2	8	1630
7.	**OliveOil**	570	60	4	40	9	4	1854
8.	**Trace**	275	200	4	80	2	10	2449
9.	**ToeSegment2**	343	166	2	40	9	20	2558
10.	**Car**	577	120	4	60	6	13	3018
11.	**Lightning2**	637	121	2	80	4	10	3564
12.	**ShapeletSim**	500	200	2	40	2	22	6032
13.	**DiatomSizeRed**	345	322	4	40	2	1	6198
14.	**Adiac**	176	781	37	40	8	13	9213
15.	**HouseTwenty**	2000	159	2	40	5	12	11098

Table 3: Runtime (sec) Results for a Single Record.

ID No.	Data set	Standard DTW (B'mark)	Enhanced Sub-Seq. Based DTW	Enhanced Param. Learning	Enhanced Param. Learn Cand. Reduct.
1.	**PenDigits**	19.00	4.40	4.50	0.27
2.	**SmoothSubspace**	1.33	0.45	0.45	0.26
3.	**ItalyPowerDemand**	2.20	0.70	0.75	0.26
4.	**Libras**	1.40	0.45	0.50	0.26
5.	**SyntheticControl**	1.65	0.60	0.55	0.25
6.	**GunPoint**	1.20	0.40	0.35	0.25
7.	**OliveOil**	1.30	0.36	0.40	0.25
8.	**Trace**	1.40	0.52	0.52	0.26
9.	**ToeSegment2**	1.55	0.54	0.60	0.25
10.	**Car**	1.90	0.51	0.52	0.26
11.	**Lightning2**	1.85	0.50	0.55	0.26
12.	**ShapeletSim**	2.11	0.65	0.72	0.28
13.	**DiatomSizeRed**	2.39	0.79	0.85	0.26
14.	**Adiac**	2.30	1.10	1.22	0.25
15.	**HouseTwenty**	17.01	1. 43	1.70	0.28

5.3 Accuracy of performance

In term of the effectiveness of the proposed technique comparisons were conducted using accuracy and F1 score. The results are presented in Table 4, standard deviation values are given in parenthesis. The presented are average values derived using Ten Cross Validation (TCV). From the table, it can be seen that the accuracy and F1 score are either improved or remain unchanged using the proposed mechanism (and with runtime gains). In some datasets, such as SmoothSubspace, Libras, GunPoint, Car, and ShapeletSim have improved significantly in comparison with the fundamental DTW. Whilst the Enhanced Sub-Sequence-Based DTW and Enhanced Parameters Learning has similar performance in term of accuracy and F1 score.

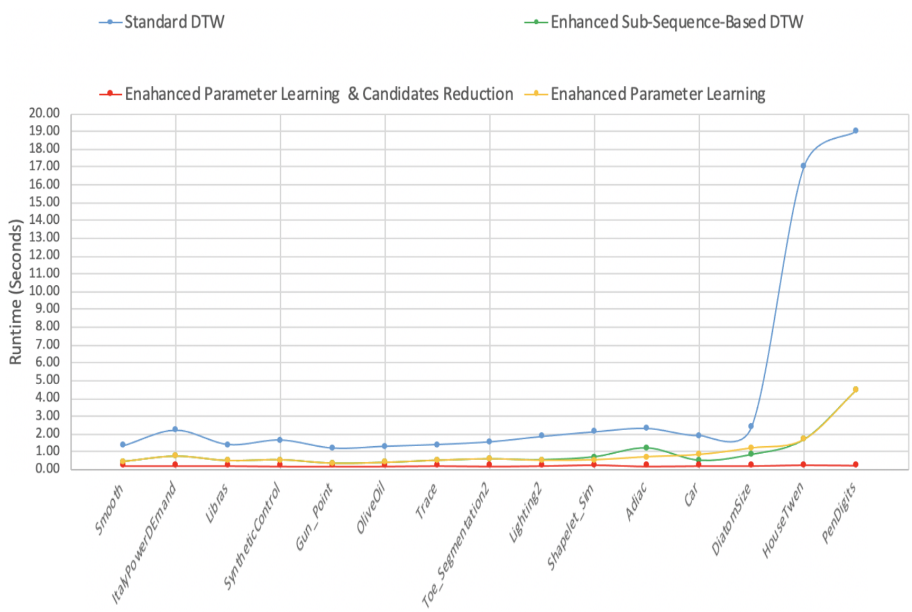

Fig. 4: Average TCV runtime results (seconds) to classify a single record.

6 Conclusion

In this paper, an enhanced Sub-Sequence-Based DTW approach, the Enhanced Sub-sequence-Based DTW with parameter learning and candidate reduction mechanism, has been presented. The operation of the proposed approach was compared with Standard DTW, Enhanced Sub-Sequence-Based DTW and Enhanced Sub-Sequence-Based DTW with parameter learning but no candidate reduction. For the experiments the k-Nearest Neighbour classification algorithm, with $k = 1$, was used, coupled with the Ten Cross Validation (TCV) technique, with respect to 15 datasets taken from the UEA and UCR (University of East Anglia and University of California Riverside) Time Series Classification Repository [4]. A comparison was conducted in terms of runtime, and accuracy and F1 score. The runtimes recorded for the four mechanisms were presented, these

Table 4: Best accuracy and F1 results, overall best accuracies and F1 values highlighted in bold font.

ID #	Data set	Bencmark Standard DTW		Enhanced Sub-Sequenc Based DTW		Enhanced Param. Learning		Sub-Sequence B0.98ased DTW and Cand. Reduct.	
		Acc	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**
1	**PenDigits**	85.50 (0.01)	0.85 (0.01)	88.48 (0.01)	0.88 (0.01)	88.48 (0.01)	0.88 (0.01)	**89.28** (0.01)	**0.89** (0.01)
2	**Smooth Subspace**	91.00 (0.04)	0.91 (0.04)	98.33 (0.03)	**0.99** (0.03)	98.33 (0.03)	**0.99** (0.03)	**98.67** (0.01)	**0.99** (0.01)
3	**ItalyPower Demand**	95.70 (0.02)	**0.96** (0.02)	96.34 (0.01)	**0.96** (0.01)	96.34 (0.02)	**0.96** (0.02)	**96.62** (0.01)	**0.96** (0.01)
4	**Libras**	62.59 (0.10)	0.60 (0.11)	66.51 (0.14)	0.64 (0.15)	67.22 (0.12)	0.65 (0.12)	**68.00** (0.11)	**0.66** (0.11)
5	**Synthetic Control**	98.00 (0.01)	**0.98** (0.01)	98.33 (0.01)	**0.98** (0.01)	98.33 (0.01)	0.98 (0.01)	**98.50** (0.01)	**0.98** (0.01)
6	**GunPoint**	93.97 (0.04)	0.94 (0.05)	99.00 (0.02)	**0.99** (0.02)	99.00 (0.02)	**0.99** (0.02)	**99.50** (0.01)	**0.99** (0.01)
7	**OilveOil**	89.52 (0.15)	0.88 (0.16)	90.12 (0.10)	0.89 (0.12)	90.12 (0.10)	0.89 (0.12)	**91.54** (0.11)	**0.91** (0.11)
8	**Trace**	99.00 (0.03)	**0.99** (0.03)	96.50 (0.04)	0.97 (0.04)	99.00 (0.03)	**0.99** (0.03)	**99.50** (0.01)	**0.99** (0.01)
9	**Toe Segmentation2**	89.07 (0.09)	0.88 (0.10)	92.26 (0.03)	0.92 (0.04)	90.56 (0.06)	0.90 (0.07)	**92.30** (0.04)	**0.92** (0.04)
10	**Car**	80.83 (0.07)	0.80 (0.09)	82.50 (0.10)	0.81 (0.11)	81.67 (0.11)	0.80 (0.12)	**88.33** (0.08)	**0.88** (0.09)
11	**Lightin2**	87.74 (0.09)	0.87 (0.08)	87.40 (0.08)	0.87 (0.09)	89.26 (0.06)	0.89 (0.07)	**91.00** (0.09)	**0.91** (0.09)
12	**DiatomSize Reduction**	99.36 (0.01)	0.99 (0.01)	**100.00** (0.00)	**1.00** (0.00)	**100.00** (0.00)	**1.00** (0.00)	**100.00** (0.00)	**1.00** (0.00)
13	**ShapleletSim**	82.37 (0.09)	0.81 (0.11)	**93.00** (0.04)	**0.93** (0.04)	92.00 (0.06)	0.92 (0.06)	**93.00** (0.04)	**0.93** (0.04)
14	**Adiac**	64.63 (0.03)	0.62 (0.04)	64.98 (0.03)	0.62 (0.04)	65.43 (0.02)	0.63 (0.03)	**66.70** (0.02)	**0.66** (0.03)
15	**HouseTwenty**	**93.75** (0.04)	**0.94** (0.04)	91.17 (0.07)	0.91 (0.07)	91.17 (0.07)	0.91 (0.07)	**93.75** (0.04)	**0.94** (0.04)

demonstrated that the proposed approach outperformed the other models significantly. In addition, as the number of records, or time series length, was increased, the runtime advantage becomes more evident. With respect to the recored accuracy and F1 scores, the results demonstrated that the proposed mechanism, incorporating candidate reduction, produced better performance compared to other mechanisms considered.

References

1. Alshehri, M., Coenen, F., Dures, K.: Effective sub-sequence-based dynamic time warping. In: International Conference on Innovative Techniques and Applications of Artificial Intelligence. pp. 293–305. Springer (2019)
2. Alshehri, M., Coenen, F., Dures, K.: Sub-sequence-based dynamic time warping. In: KDIR. pp. 274–281 (2019)
3. Bagnall, A., Lines, J.: An experimental evaluation of nearest neighbour time series classification. arXiv preprint arXiv:1406.4757 (2014)
4. 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)
5. 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)
6. Brunello, A., Marzano, E., Montanari, A., Sciavicco, G.: A novel decision tree approach for the handling of time series. In: International Conference on Mining Intelligence and Knowledge Exploration. pp. 351–368. Springer (2018)
7. Gamboa, J.C.B.: Deep learning for time-series analysis. arXiv preprint arXiv:1701.01887 (2017)
8. Huang, S.F., Guo, M., Chen, M.R.: Stock market trend prediction using a functional time series approach. Quantitative Finance **20**(1), 69–79 (2020)
9. Itakura, F.: Minimum prediction residual principle applied to speech recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing **23**(1), 67–72 (1975)
10. Kocian, A., Chessa, S.: Auto regressive integrated moving average modeling and support vector machine classification of financial time series. In: International Symposium on Distributed Computing and Artificial Intelligence. pp. 1–8. Springer (2018)
11. Kulkarni, N.: Effect of dynamic time warping using different distance measures on time series classification. International Journal of Computer Applications **975**, 8887 (2017)
12. Liu, Z., Jiang, P., Zhang, L., Niu, X.: A combined forecasting model for time series: Application to short-term wind speed forecasting. Applied Energy **259**, 114137 (2020)
13. Niennattrakul, V., Ratanamahatana, C.A.: Learning dtw global constraint for time series classification. arXiv preprint arXiv:0903.0041 (2009)
14. 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)
15. 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)

16. 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)
17. 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)
18. Zhou, M., Wong, M.H.: Boundary-based lower-bound functions for dynamic time warping and their indexing. Information Sciences **181**(19), 4175–4196 (2011)