

# Motif-based Classification using Enhanced Sub-Sequence-Based Dynamic Time Warping

Mohammed Alshehri<sup>1,2</sup>, Frans Coenen<sup>1</sup> and Keith Dures<sup>1</sup>

<sup>1</sup>Department of Computer Science, University of Liverpool, Liverpool, U.K.

<sup>2</sup>Department of Computer Science, King Khalid University, Abha, Saudi Arabia

**Keywords:** Time Series Analysis, Dynamic Time Warping,  $K$ -Nearest Neighbour Classification, Sub-Sequence-Based DTW, Matrix Profile, Motifs.

**Abstract:** In time series analysis, Dynamic Time Warping (DTW) coupled with  $k$  Nearest Neighbour classification, where  $k = 1$ , is the most commonly used classification model. Even though DTW has a quadratic complexity, it outperforms other similarity measurements in terms of accuracy, hence its popularity. This paper presents two motif-based mechanisms directed at speeding up the DTW process in such a way that accuracy is not adversely affected: (i) the Differential Sub-Sequence Motifs (DSSM) mechanism and (ii) the Matrix Profile Sub-Sequence Motifs (MPSSM) mechanism. Both mechanisms are fully described and evaluated. The evaluation indicates that both DSSM and MPSSM can speed up the DTW process while producing a better, or at least comparable accuracy, in 90% of cases.

## 1 INTRODUCTION

Today's technology allows us to collect large amounts of time series data. Examples include stock market data (Ebadati and Mortazavi, 2018), weather data (Karevan and Suykens, 2020) and electrocardiogram data (Phinyomark and Scheme, 2018). Time series analysis is directed at finding and extracting meaningful knowledge from this data. The analysis can take many forms, but a frequently encountered example is time series classification where we wish to build a model of the time series data we have and then use this model to label a time series  $\mathfrak{t}$ , that we have not previously seen, according to a set of classes  $C$ .

Time series classification, whether supervised, unsupervised or somewhere between the two, requires a comparison of time series. The number of comparisons to be undertaken is the main contributing factor to the computational complexity of time series classification. A range of techniques is available to calculate similarity between two time series. Euclidean Distance (ED) and Dynamic Time Warping (DTW) are the most widely used techniques. Although ED is faster, it has been shown to be less accurate than DTW (Rakthanmanon et al., 2012; Silva et al., 2018), and does not support comparison of time series of different length, whilst DTW does. On the other hand, DTW is slower. DTW has a time complexity  $O(x^2)$ ,

compared to a time complexity of  $O(x)$  for ED (where  $x$  is the length of the time series). In the context of supervised time series classification a range of algorithms is available: Decision Trees (Brunello et al., 2018), Artificial Neural Networks (Gamboa, 2017) and Deep Learning (Fawaz et al., 2019). However,  $k$ -Nearest Neighbour ( $k$ NN), with  $k = 1$  and DTW as the similarity measurement, remains the most frequently used algorithm for time series classification (Rakthanmanon et al., 2012; Silva et al., 2018). The work presented in this paper is directed at reducing the time complexity of DTW with a focus on the  $k$ NN algorithm with  $k = 1$ .

DTW was first introduced in the speech recognition community (Sakoe and Chiba, 1978). The main idea was to find the optimal match, the minimum "warping distance",  $wd$ , between two time series,  $S_1 = [p_1, p_2, \dots, p_x]$  and  $S_2 = [q_1, q_2, \dots, q_y]$  (where  $p_i$  and  $q_j$  are individual values in the time series, and  $x$  and  $y$  are the time series lengths). The DTW process can be described as follows. A distance matrix  $M$  of size  $x \times y$  is generated where the value held at each cell  $m_{i,j}$  is calculated as shown in Equation 1 (Niennattrakul and Ratanamahatana, 2009), where  $d_{ij}$  is the ED between the corresponding points  $p_i \in S_1$  and  $p_j \in S_2$ , to which is added the minimum value from the three "previous" cells ( $m_{i,j}$  ( $m_{i-1,j}$ ,  $m_{i-1,j-1}$  or  $m_{i,j-1}$ ) (Alshehri et al., 2019a). At the end of the

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 $S = [p_1, p_2, \dots]$ ( $S = [q_1, q_2, \dots]$ ), $S \in D$ .
$x$ or $y$	The length of a given time series.
$D$	A collection of time series $\{S_1, S_2, \dots, S_r\}$
$r$	The number of time series in in $D$ .
$C$	A set of class labels, $C = \{c_1, c_2, \dots\}$ , associated with a $D$ .
$M$	A distance matrix measuring $x \times y$ (used for DTW).
$m_{i,j}$	The distance value at location $i, j$ in $M$ .
$wd$	A warping distance derived from $M$ .
$\ell$	The number of points in a subsequence.
$w$	A time series subsequence $\{p_i, p_{i+1}, \dots\}$ , such that $w \in S$
$s$	The number of sub-sequences into which a given time series is to be split, $s = x/\ell$ ( $s = y/\ell$ ).
$t$	The tail measured backwards from $\ell$ within which a cut is to be applied to create a sub-sequence; thus given $S = [p_0, \dots, p_\ell]$ the cut will fall between $p_\ell$ and $p_{\ell-t}$ .
$W$	A set of $s$ time series subsequences, $\{w_1, w_2, \dots, w_s\}$ contained in a given time series $S$
$\bar{l}$	The number of points in a motif (used with the DSSM mechanism).
$n$	The length of a window in a Matrix Profile (used with the MPSSM mechanism).
$\dagger$	A new previously unseen time series to be classified (labeled).

process, the minimum  $wd$  will be held at  $m_{x,y}$ . Two time series are identical if  $wd$  equates to zero. As the value of  $wd$  increases, the similarity reduces.

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

There has been previous work directed at reducing the complexity of DTW, typically directed at reducing the size of  $M$ . One example can be found in (Alshehri et al., 2019b) where the *Sub-Sequence-Based DTW* mechanism was proposed. The main idea here was to speed up the DTW process by splitting the two time series to be compared into equally-sized sub-sequences of length  $\ell$ . Consequently, the size of  $M$  was reduced by a factor of  $\ell$ . The process of DTW was then applied in each corresponding sub-sequence simultaneously. Finally, the values held in  $m_{x,y}$  for all sub-sequence was accumulated to give final  $wd$  value. This mechanism produced better results compared to standard DTW, not only in terms of run-time, but also in terms of accuracy and F1-Score. This paper builds on this idea, but instead investigates the potential of using only a limited number of sub-sequences. The idea is akin to the concept of motifs proposed in (Torkamani and Lohweg, 2017). In more detail, this paper proposes two motif-based classification mechanisms founded on the Sub-Sequence-Based DTW idea: (i) the Differential Sub-Sequence Motifs (DSSM) mechanism and (ii) the Matrix Profile Sub-Sequence Motifs (MPSSM) mechanism. The

distinction is how the motifs are identified. Using the DSSM mechanism the set of classes,  $C$ , is used to select motifs that are good differentiators of class. Using the MPSSM mechanism, the matrix profile idea, proposed on (Yeh et al., 2016) is used.

The remainder of this paper is organised as follows. A review of related work is presented in Section 2. The proposed DSSM and MPSSM mechanisms are presented in Section 3. The theoretical computational complexity of the proposed mechanisms is presented in Section 4. The evaluation of the proposed techniques is then presented in Section 5, together with a discussion of the results obtained. The paper is concluded in Section 7. For convenience, a symbol table is given in Table 1 listing the symbols used throughout the paper.

## 2 BACKGROUND AND PREVIOUS WORK

The Sub-Sequence-Based DTW idea, first proposed in (Alshehri et al., 2019b), was directed at speeding up the DTW process by segmenting two time series  $S_1$  and  $S_2$  into sub-sequences. 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  $S_2 = [U_{2_1}, U_{2_2}, \dots, U_{2_s}]$ . DTW was then applied to

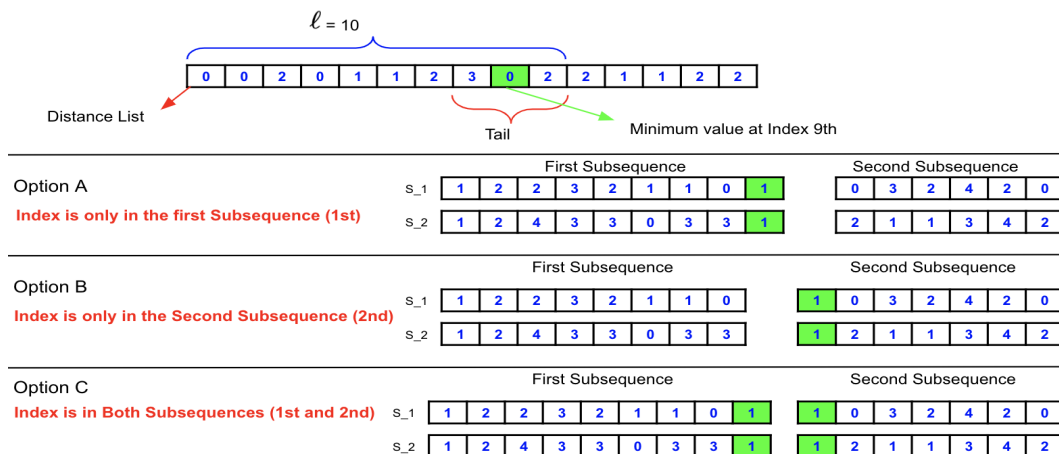


Figure 1: Segmentation examples given two time series  $S_1$  and  $S_2$ , and SPAO options  $A$ ,  $B$  or  $C$  (Alshehri et al., 2019a).

each sub-sequence pairing  $U_{1_i}, U_{2_j}$  where  $i = j$ . The minimum  $wd$  values arrived at were then accumulated to give a final  $wd$  value. The approach worked well in some cases, and not so well in other cases; this was found to be because of the use of a fixed value for  $s$  which suited some applications, but not all. Therefore, an improved approach, Enhanced Sub-Sequence-Based DTW, was proposed in (Alshehri et al., 2019a) which featured a more flexible way of dividing a time series into sub-sequence by finding the most appropriate size for  $s$  by using two parameters: the maximum length of a sub-sequences  $\ell$  and a tail  $t$ , measured backwards from  $\ell$ , within which the cut was to be applied (Alshehri et al., 2019a). Thus, given a time series  $U = [p_0, \dots, p_\ell, \dots]$ , the first “cut” will fall between  $p_\ell$  and  $p_{\ell-t}$ ; and so on for further cuts. In addition, a Split Point Allocation Option (SPAO) was considered. This featured three alternative ways of including the split point, identified as  $A$ ,  $B$  and  $C$ , as illustrated in Figure 1. Option  $A$  was found to provide the best performance and was therefore used with respect to the evaluation presented later in this paper.

The MPSSM mechanisms presented later in this paper is founded on the concept of matrix profiles, as first introduced in (Yeh et al., 2016), for finding motifs within time series. Motifs are repeating patterns in a time series. The Matrix Profile technique has two main components: (i) a distance profile and (ii) a profile index. The distance profile is constructed using a sliding window technique and holds “distance” values. The profile index holds indexes to sub-sequences referenced in the distance profile. The similarity is measured using Euclidean Distance. Only one parameter is used, the sliding window size  $n$ . The distance profile is used to identify frequently occurring distances which are referenced to the index profile which in turn references individual sub-sequences in the in-

put which are then identified as motifs. The distance profile idea is central to the MPSSM mechanism described later in this paper. Figure 2 gives an example of the matrix profile generated from an original time series.

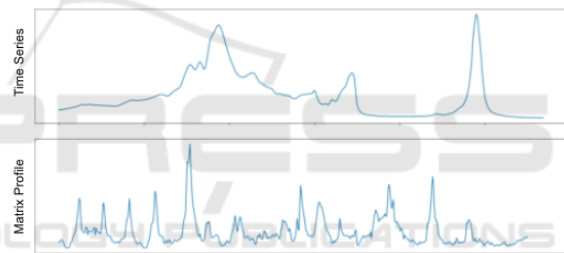


Figure 2: Matrix Profile generation. Top: the original time series. Bottom: the resulting Matrix Profile.

### 3 PROPOSED TECHNIQUES

In this section, the two proposed time series classification mechanisms, the Differential Sub-Sequence Motifs (DSSM) mechanism and the Matrix Profile Sub-Sequence Motifs (MPSSM), are presented. Recall that the objective is to speed up the DTW process by reshaping time series to a form based on the concept of motifs. Then, Enhanced Sub-Sequence-Based DTW is applied to the reshaped time series data. The DSSM mechanism is presented in Sub-section 3.1 and the MPSSM mechanism in Sub-section 3.2 respectively.

#### 3.1 Differential Sub-Sequence Motifs

This sub-section presents the proposed DSSM mechanism. The pseudo code for the DSSM mechanism

Algorithm 1: Differential Sub-Sequence Motifs.

---

```

1: input  $D, |C|, \bar{l}$ 
2:  $D' = \{\langle S'_1, c_1 \rangle, \langle S'_2, c_2 \rangle, \dots, \langle S'_r, c_r \rangle\}, S_i = \emptyset \triangleright D$ 
   Reshaped
3:  $A =$  Temporary set of sets length  $|C|$  to hold sets
   of time series
4: for  $\forall \langle S_j, c_j \rangle \in D$  do  $\triangleright$  Populate  $A$ 
5:    $A_i = A_i \cup \langle S_j, c_j \rangle$  ( $i = j$ )
6: end for
7: for  $\forall A_i \in A$  do  $\triangleright$  Populate  $B$ 
8:    $B =$  Temporary array  $[dist_1, dist_2, \dots, dist_r],$ 
      $dist_i = 0$ 
9:   for  $S_j \in A_i, j = 0$  to  $j = r - 1$  do
10:    for  $\forall p_x \in S_j$  and  $\forall q_x \in S_{j+1}$  do
11:       $dist_x \in B = dist_x + abs(p_x - q_x)$ 
12:    end for
13:  end for
14:   $W = [w_1, w_2, \dots, w_s],$  array of sub-sequences
     in  $B$ , each of length  $\bar{l}$ 
15:   $I = [i_1, i_2, \dots, i_s],$  array of indexes to  $S_1 \in A_i$ 
     (one-to-one match
     with  $W$ )
16:   $E =$  Temporary array  $[dist_1, dist_2, \dots, dist_s]$ 
17:  for  $\forall w_j \in W$  do  $\triangleright$  Populate  $E$ 
18:     $dist_j \in E = \sum_{i=0}^{i=\bar{l}} p_i \in w_j$ 
19:  end for
20:   $F =$  Temporary array  $[count_1, count_2, \dots]$ 
     holding frequency counts
     for each  $dist_j \in E$ 
21:   $S'_i \in D' =$  sub-sequence in  $S_1 \in A_i$  associated
     with  $w_i$  that has the
     highest frequency count in  $F$ 
22: end for
23: return  $D'$ 

```

---

is given in Algorithm 1. The inputs (line 1) are: (i) the data set  $D = \{\langle S_1, c_1 \rangle, \langle S_2, c_2 \rangle, \dots, \langle S_r, c_r \rangle\}$ , where each  $S_i$  is a time series and  $c_i$  is the associated class label taken from the set of classes  $C$  ( $c_i \in C$ ), (ii) the number of classes in  $C$  and (iii) a sub-sequence length  $\bar{l}$ . The first step (line 2) is to declare the reshaped data set  $D'$  which is to be populated as the process progresses. Next (lines 3 to 6) the time series in  $D$  are grouped according to their associated class and placed in a set of sets  $A = \{A_1, A_2, \dots, A_{|C|}\}$ , where set  $A_i$  holds the collection of time series associated with class  $c_i$ . This set of sets is then processed, lines 7 to 22, so that a reshaped input set is produced (stored in  $D'$ ). For each set  $A_i \in A$ , associated with a particular class  $c_i$ , a temporary array  $B$  of length  $r$  is generated (lines 8 to 13), which holds the accumulated distances for each value in the time series in  $A_i$ . Thus the accumulated distances between the time se-

ries for time point 1, time point 2, and so on up to time point  $r$  (the assumption is that the input time series are all of the same length). The array  $B$  is then, line 14, divided into a set of non-overlapping sub-sequences,  $W = \{w_1, w_2, \dots, w_s\}$ , each of length  $\bar{l}$ . An array of indexes is also created (line 15) that links the start of each sub-sequence in  $W$  back to the corresponding sub-sequence in time series  $S_1 \in A_i$ ; the significance is that one of these sub-sequences will be selected as the motif to represent class  $c_i$ . A temporary array  $E$  is then created, lines 16 to 18, to hold the accumulated distances (sum of distances) held in each sub-sequence  $w_j \in W$ . Note that there is a one-to-one correspondence between  $W$  and  $E$ . A third temporary array  $F$  is then created (line 20) to hold the frequency count of each distance  $dist_i \in E$ ; the length of  $F$  will depend on the number of unique distances held in  $E$ . The sub-sequence in  $S_1 \in A_i$  which is associated with the distance  $w_j \in W$  that has the highest frequency count as listed in  $F$ , is then selected as the motif for class  $c_i$  to be included in  $D'$  (line 21); this is facilitated by the array of indexes  $I$  created earlier (line 15). At the end of the process  $D'$  will be populated with a set of sub-sequences, representative of the input time series, one sequence per class.

Effective Sub-Sequence DTW, described earlier, will then be used to label previously unseen time series using  $kNN$  with  $k = 1$ . Given a new time series to be labeled,  $\ddagger$ , this is first segmented into a sequences of sub-sequences, each of length  $\bar{l}$ . The sub-sequences in  $\ddagger$  will then be compared to the motifs in  $D'$  and the class associated with the most similar motif in  $D'$  adopted as the label for  $\ddagger$ .

### 3.2 Matrix Profile Sub-Sequence Motifs

The second motif-based mechanism considered in this paper is the MPSSM mechanism. This uses the matrix profile technique from (Yeh et al., 2016) as outlined in Section 2; although unlike the technique described in (Yeh et al., 2016) an index profile is not used. Instead the the distance profile is used as the reshape input data  $D'$ . The pseudo code for the MPSSM mechanism is given in Algorithm 2. The input (line 1) is the data set  $D = \{\langle S_1, c_1 \rangle, \langle S_2, c_2 \rangle, \dots, \langle S_r, c_r \rangle\}$  and a window size  $n$ . The first step (lines 2), as in the case of the DSSM mechanism, is to declare the reshaped data set  $D'$  which is to be populated as the process progresses. This next step, lines 3 to 10, is to reshape the input  $D$  into a distance matrix which will be stored in  $D'$ . Each time series  $S_i \in D$  is segmented (line 4) into  $x - n + 1$  sub-sequences each of length  $n$  (recall that  $x$  is the length of the time series in  $D$ ). We then (line 5) create a comparator time series

$T$  comprised of the first  $n$  points in  $S$ . Then (lines 6 to 8), for each sub-sequence  $w_j \in W$  we determine the Euclidean Distance between  $w_j$  and  $T$  and add this to  $S'_i \in D'$ . At the end of the process we have a distance profile held in  $D'$ .

Algorithm 2: Matrix Profile Sub-Sequence Motifs.

---

```

1: input  $D, n$ 
2:  $D' = \{\langle S'_1, c_1 \rangle, \langle S'_2, c_2 \rangle, \dots, \langle S'_r, c_r \rangle\}, S_i = \emptyset \quad \triangleright D$ 
   Reshaped
3: for  $\forall S_i = [p_1, p_2, \dots, p_x] \in D$  do  $\triangleright$  Create
   distance profile
4:    $W =$  a list of time series sub-sequence, of
     length  $x - n + 1$ , generated by moving
     a window, of length  $n$  along  $S_i$ 
5:    $T = [p_1, \dots, p_n]$ 
6:   for  $\forall w_j \in W$  do
7:      $d =$  distance between  $T$  and  $w_j$ 
8:      $S'_i = S'_i \cup d$ 
9:   end for
10: end for
11: return  $D'$ 

```

---

Given a previously unseen time series,  $\dagger$ , this will be compared with the contents of  $D'$  using Effective Sub-Sequence-Based DTW. To do this it has to be reshaped in the same manner as the “training” data to give  $\dagger'$ . This is done by repeating lines 4 to 9 of the pseudo code given in Algorithm 2, but with  $S_i$  replaced with  $\dagger$ , and  $S - i'$  replaced with  $\dagger'$ .

## 4 TIME COMPLEXITY

The time complexity of the two proposed mechanisms are presented in this section. When comparing two time series  $S_1$  and  $S_2$ , using standard DTW, the time complexity ( $DTW_{compStand}$ ) depends on the size of the distance matrix  $M$ . The time complex is thus given by  $O(x \times y)$  where  $x$  and  $y$  are the lengths of  $S_1$  and  $S_2$  respectively (Alshehri et al., 2019a). If both  $S_1$  and  $S_2$  are of the same length (number of points in each time series), the time complexity can be simplified to:

$$DTW_{compStand} = O(x^2) \quad (2)$$

With respect to Sub-Sequence-Based DTW (Alshehri et al., 2019a) as first proposed in (Alshehri et al., 2019b), the DTW time complexity,  $DTW_{compSubS}$ , reduces to:

$$DTW_{compSubS} = O\left(\frac{x^2}{x \div \ell}\right) \quad (3)$$

The time complexity using the DSSM mechanism

for comparing two time series,  $DTW_{compDSSM}$ , where one time series has been reduced to a single motif of length  $\ell$ , and the other has been segmented into  $s$  sub-sequences of length  $\ell$ , is then given by:

$$DTW_{compDSSM} = O(\ell^2 \times s) \quad (4)$$

With respect to MPSSM mechanism, where we are comparing a reshaped time series (a row in a distance profile) with another reshaped time series, each of length  $x - n + 1$ . the time complexity will be:

$$DTW_{compMPSSM} = O((x - n + 1) \times (x - n + 1)) \quad (5)$$

where:  $n$  is the window size and  $x$  is the length of a time series (assuming all time series are of the same length).

When using  $k$ -nearest neighbour ( $k$ NN) classification with  $k = 1$ , the most frequently used time series classification model (Bagnall et al., 2017; Silva et al., 2018), a new time series  $\dagger$  to be classified will need to be compared to all records  $r \in D$ . The complexity when using standard DTW or sub-sequence DTW will be:

$$O(r \times complexity \times |\dagger|) \quad (6)$$

where: (i)  $r$  is the number of records in the  $k$ NN “bank”, (ii)  $complexity$  is either  $DTW_{compStand}$ ,  $DTW_{compSubS}$  or  $DTW_{compMPSSM}$ , and (iii)  $|\dagger|$  is the number of previously unseen time series to be labeled. Using the DSSM mechanism, where we have only one motif per class in the  $k$ NN bank, this reduces to:

$$O(|C| \times DTW_{compDSSM} \times |\dagger|) \quad (7)$$

## 5 EVALUATION

The evaluation of the proposed DSSM and MPSSM mechanism is presented in this section. Their operation was compared with: (i) Standard DTW (the Benchmark) and (ii) Enhanced Sub-Sequence-Based DTW with  $\ell = 40$  and  $t = 2$  and Option A as recommended in (Alshehri et al., 2019a) (see Section 2 for detail). The evaluation was conducted using  $k$ NN with  $k = 1$ . Ten datasets taken from the UEA and UCR Time Series repository (Bagnall et al., 2017) were used for the evaluation reported here. Datasets of different size of data were considered, ranging from  $x = 150$  to  $x = 2000$  (time series length) and from  $r = 60$  to  $r = 781$  (number of records). An overview of the data sets used is given in Table 2. In the table the datasets are listed according to  $x$  (Column 3). Column 5, shows the number of classes ( $|C|$ ). The evaluation objectives were:



1. **Parameter Settings:** To identify the best parameters settings for the two proposed mechanisms.
2. **Runtime:** To compare the operation of the proposed mechanisms, in terms of run time, with the operation of Standard DTW and Effective Sub-Sequence-Based DTW.
3. **Accuracy:** To compare the operation of the proposed mechanisms, in terms of accuracy, with the operation of Standard DTW and Effective Sub-Sequence-Based DTW.

Each is discussed in turn in the following three sub-sections, Sub-section 6 to Sub-section 6.2. For the experiments, a desktop computer with a 3.5 GHz Intel Core i7 processor and 28 GB, 2400 MHz, DDR3 of primary memory was used.

## 6 PARAMETER SETTINGS

The results from experiments to identify best parameter settings for DSSM and MPSSM are given in Tables 3 and 4 respectively. Recall that the distinction between the two is that DSSM uses a parameter  $\bar{l}$  (the number of points in a motif) while MPSSM uses a parameter  $n$  (window size). The final column in each table (Column 6) gives the relative run time (seconds). The results are averages obtained using cross validation. The range of test values used for  $\ell$  and  $t$  were the same as those used in (Alshehri et al., 2019a). For the parameter  $\bar{l}$ , this was defined in terms of a percentage of the overall length of a time series, from 5% to 95% incrementing in steps of 5%,  $\{5\%, 10\%, \dots, 95\%\}$ .

Table 2: Evaluation Time Series Datasets.

ID No.	Dataset Name	Len. ( $x$ )	Num. recs. ( $r$ )	Num. Classes
1.	<b>GunPoint</b>	150	200	2
2.	<b>OliveOil</b>	570	60	4
3.	<b>Trace</b>	275	200	4
4.	<b>ToeSegment2</b>	343	166	2
5.	<b>Car</b>	577	120	4
6.	<b>Lightning2</b>	637	121	2
7.	<b>ShapeletSim</b>	500	200	2
8.	<b>DiatomSizeRed</b>	345	322	4
9.	<b>Adiac</b>	176	781	37
10.	<b>HouseTwenty</b>	2000	159	2

### 6.1 Run Time Performance

In this sub-section, the runtime performance for the two proposed mechanisms is considered. Table 5 gives the runtime results for the four mechanisms considered. The runtimes for DSMM and MPSSM,

Table 3: DSSM Best Parameters,  $\ell$ ,  $t$  and  $\bar{l}$ .

ID No.	Dataset Name	Parameters			R'time (sec)
		$\ell$	$t$	$\bar{l}$	
1.	<b>GunPoint</b>	10	1	35%	3.00
2.	<b>OliveOil</b>	40	2	90%	2.30
3.	<b>Trace</b>	70	1	80%	6.50
4.	<b>ToeSegment2</b>	100	1	80%	7.90
5.	<b>Car</b>	60	2	35%	2.85
6.	<b>Lightning2</b>	40	2	60%	5.70
7.	<b>ShapeletSim</b>	90	2	95%	20.60
8.	<b>DiatomSizeRed</b>	20	2	20%	6.00
9.	<b>Adiac</b>	10	2	95%	120
10.	<b>HouseTwenty</b>	300	5	45%	45.00

Table 4: MPSSM Best Parameters,  $\ell$ ,  $t$  and  $n$ .

ID No.	Dataset Name	Parameters			Runtime (sec)
		$\ell$	$t$	$n$	
1.	<b>GunPoint</b>	40	2	30	5.18
2.	<b>OliveOil</b>	80	2	20	2.30
3.	<b>Trace</b>	50	2	40	6.51
4.	<b>ToeSegment2</b>	50	2	15	9.05
5.	<b>Car</b>	50	6	35	9.10
6.	<b>Lightning2</b>	110	2	45	9.60
7.	<b>ShapeletSim</b>	70	2	5	20.50
8.	<b>DiatomSizeRed</b>	20	2	40	25.11
9.	<b>Adiac</b>	10	5	70	115
10.	<b>HouseTwenty</b>	400	2	20	130

the last two columns, are taken from Tables 3 and 4 respectively. From the table, it can be seen that the runtime of the two proposed techniques is faster in nine of the ten cases; with DSMM providing the best performance. The exception was the Adiac data set; the reason for this may have something to do with this being the largest data set in terms of number of records. With respect to the DiatomSizeReduction and HouseTwenty; the runtime using DSSM was more than 10 times faster when compared to the standard DTW. Figure 3 shows the same runtime results as given in Table 5, but in graph form with the datasets arranged along the x-axis the run time on the y-axis.

### 6.2 Accuracy of Performance

In this sub-section, the performance of the two proposed mechanism is considered in terms of accuracy and F1 score. The results are given in Table 6; standard deviation values are given in parenthesis. The values given in the table are the average of Ten Cross Validations (TCVs). Best results are highlighted in bold font. From the Table, it can be seen that with respect to DSSM the accuracy improved, over standard DTW and Enhanced Sub-sequence based DTW, in five cases, and remained the same in a further three cases. Only in two cases, ShapeletSim and Adiac, did DSSM produce a worse performance. With re-

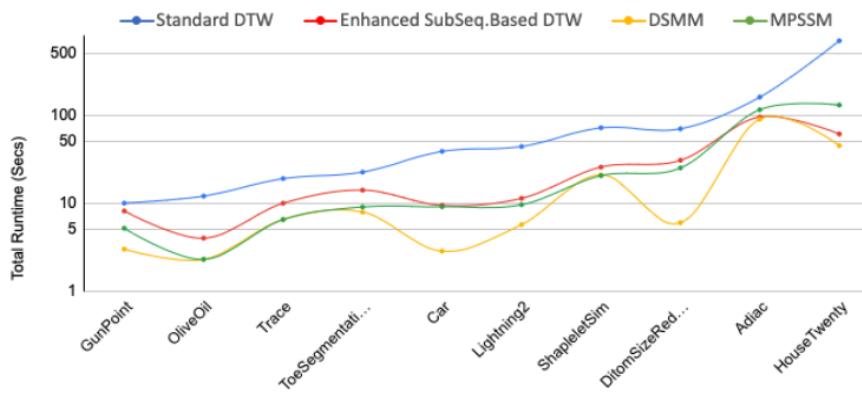


Figure 3: Total TCV runtime results (seconds) to classify each datasets.

Table 5: Total Runtime Results (seconds) using Ten Cross Validation, best results in bold font.

ID No.	Data Set	Standard DTW (B'mark)	Enhanced Sub-Seq. Based DTW	DSMM	MPSSM
1.	<b>GunPoint</b>	10.00	8.16	<b>3.00</b>	5.18
2.	<b>OliveOil</b>	12.02	4.00	<b>2.30</b>	2.30
3.	<b>Trace</b>	18.98	9.96	<b>6.50</b>	6.51
4.	<b>ToeSegment2</b>	22.49	14.05	<b>7.90</b>	9.05
5.	<b>Car</b>	38.69	9.45	<b>2.85</b>	9.10
6.	<b>Lightning2</b>	43.90	11.33	<b>5.70</b>	9.60
7.	<b>ShapeletSim</b>	71.69	25.65	<b>20.60</b>	20.50
8.	<b>DiatomSizeRed</b>	69.88	30.66	<b>6.00</b>	25.11
9.	<b>Adiac</b>	160.11	<b>94.94</b>	120.00	115.00
10.	<b>HouseTwenty</b>	696.88	60.89	<b>45.00</b>	130

Table 6: Best accuracy and F1 results, overall best accuracy and F1 values highlighted in bold font.

ID #	Data set	Benchmark Standard DTW		Enhanced Sub-Sequence Based DTW		DSSM		MPSSM	
		Acc	F1	Acc	F1	Acc	F1	Acc	F1
1	<b>GunPoint</b>	93.97 (0.04)	0.94 (0.05)	<b>99.00</b> (0.02)	<b>0.99</b> (0.02)	<b>99.50</b> (0.01)	<b>0.99</b> (0.01)	98.00 (0.02)	0.98 (0.02)
2	<b>OliveOil</b>	89.52 (0.15)	0.88 (0.16)	<b>90.00</b> (0.10)	<b>0.89</b> (0.12)	<b>90.00</b> (0.10)	<b>0.89</b> (0.12)	<b>90.00</b> (0.08)	<b>0.89</b> (0.10)
3	<b>Trace</b>	99.00 (0.03)	0.99 (0.03)	96.50 (0.04)	0.97 (0.04)	99.00 (0.02)	0.99 (0.02)	<b>100.00</b> (0.00)	<b>1.00</b> (0.00)
4	<b>Toe Segmentation2</b>	89.07 (0.09)	0.88 (0.10)	92.26 (0.03)	0.92 (0.04)	<b>95.11</b> (0.05)	<b>0.95</b> (0.03)	88.53 (0.05)	0.88 (0.07)
5	<b>Car</b>	80.83 (0.07)	0.80 (0.09)	82.50 (0.10)	0.81 (0.11)	<b>86.67</b> (0.11)	<b>0.86</b> (0.12)	83.33 (0.09)	0.82 (0.10)
6	<b>Lightin2</b>	87.74 (0.09)	0.87 (0.08)	87.40 (0.08)	0.87 (0.09)	<b>89.26</b> (0.06)	<b>0.89</b> (0.07)	75.26 (0.13)	0.75 (0.13)
7	<b>DiatomSize Reduction</b>	99.36 (0.01)	0.99 (0.01)	<b>100.00</b> (0.00)	<b>1.00</b> (0.00)	<b>100.00</b> (0.00)	<b>1.00</b> (0.00)	<b>100.00</b> (0.00)	<b>1.00</b> (0.00)
8	<b>ShapeletSim</b>	82.37 (0.09)	0.81 (0.11)	93.00 (0.04)	0.93 (0.04)	89.00 (0.05)	0.89 (0.06)	<b>99.50</b> (0.01)	<b>0.99</b> (0.01)
9	<b>Adiac</b>	64.63 (0.03)	0.62 (0.04)	<b>64.98</b> (0.03)	<b>0.62</b> (0.04)	64.63 (0.03)	0.62 (0.04)	53.78 (0.05)	0.52 (0.06)
10	<b>HouseTwenty</b>	93.75 (0.04)	0.94 (0.04)	91.17 (0.07)	0.91 (0.07)	<b>96.25</b> (0.05)	<b>0.96</b> (0.05)	94.29 (0.06)	0.94 (0.06)

spect to MPSSM the accuracy improved, over standard DTW and Enhanced Sub-sequence based DTW, in five cases, and remained the same in a further

two cases. Only in three cases, ToeSegmnetation2, Lightin2 and Adiac, did MPSSM produce a worse performance. Again, both DSSM and MPSSM did

not work well with respect to the Asiac data set (possibly because of its size in terms of number of records). In general, the proposed DSSM and MPSSM mechanisms, performed better or as well as Standard DTW or Enhanced Sub-sequence based DTW, but with an improved run time (much improved in some cases). From the results presented in Table 6 an argument can be made that DSSM produced the best results.

## 7 CONCLUSION

In this paper, two DTW mechanisms have been proposed founded on the concept of motifs: (i) the Differential Sub-Sequence Motifs (DSSM) mechanism and (ii) the Matrix Profile Sub-Sequence Motifs (MPSSM) mechanism. Both were directed at speeding up the DTW process without adversely affecting accuracy. The operation of the proposed mechanisms was compared with Standard DTW and Enhanced Sub-sequence based DTW, using a  $k$ NN classification model with  $k = 1$  and ten time series data sets of a variety of sizes; taken from the UEA and UCR (University of East Anglia and University of California Riverside) Time Series Classification Repository (Bagnall et al., 2017). The evaluation demonstrated that the proposed mechanisms outperformed the comparator mechanisms in nine out of ten cases with respect to run time without adversely affecting classification accuracy. Out of the two proposed mechanisms DSSM gave the best performance.

## REFERENCES

- Alshehri, M., Coenen, F., and Dures, K. (2019a). Effective sub-sequence-based dynamic time warping. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 293–305. Springer.
- Alshehri, M., Coenen, F., and Dures, K. (2019b). Sub-sequence-based dynamic time warping. In *KDIR*, pages 274–281.
- Bagnall, A., Lines, J., Bostrom, A., Large, J., and Keogh, E. (2017). 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.
- Brunello, A., Marzano, E., Montanari, A., and Sciacicco, G. (2018). A novel decision tree approach for the handling of time series. In *International Conference on Mining Intelligence and Knowledge Exploration*, pages 351–368. Springer.
- Ebadati, O. and Mortazavi, M. (2018). An efficient hybrid machine learning method for time series stock market forecasting. *Neural Network World*, 28(1):41–55.
- Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.-A. (2019). Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963.
- Gamboa, J. C. B. (2017). Deep learning for time-series analysis. *arXiv preprint arXiv:1701.01887*.
- Karevan, Z. and Suykens, J. A. (2020). Transductive lstm for time-series prediction: An application to weather forecasting. *Neural Networks*, 125:1–9.
- Niennattrakul, V. and Ratanamahatana, C. A. (2009). Learning dtw global constraint for time series classification. *arXiv preprint arXiv:0903.0041*.
- Phinyomark, A. and Scheme, E. (2018). 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)*, pages 5236–5240. IEEE.
- Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., Zakaria, J., and Keogh, E. (2012). 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*, pages 262–270. ACM.
- Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1):43–49.
- Silva, D. F., Giusti, R., Keogh, E., and Batista, G. E. (2018). Speeding up similarity search under dynamic time warping by pruning unpromising alignments. *Data Mining and Knowledge Discovery*, pages 1–29.
- Torkamani, S. and Lohweg, V. (2017). Survey on time series motif discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7(2):e1199.
- Yeh, C.-C. M., Zhu, Y., Ulanova, L., Begum, N., Ding, Y., Dau, H. A., Silva, D. F., Mueen, A., and Keogh, E. (2016). Matrix profile i: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In *2016 IEEE 16th international conference on data mining (ICDM)*, pages 1317–1322. Ieee.