A Proposal for Cost Aware Edge-Detectional Dynamic Time Warping for Time Series Classification

Hidetoshi Ito
Faculty of Software and
Information Science
Iwate Prefectural University
Iwate, Japan
Email: g031n020@s.iwate-pu.ac.jp

Basabi Chakraborty
Faculty of Software and
Information Science
Iwate Prefectural University
Iwate, Japan
Email: basabi@iwate-pu.ac.jp

***Abstract*—Dynamic Time Warping (DTW) is a well known algorithm for measuring similarity of two time series and widely used in classification, clustering or regression problems related to time series data. Unlike simple Euclid distance measure, DTW can handle time series of unequal lengths and is able to find an optimal alignment between two time sequences. Though very efficient, the computational cost of DTW is very high. There are several suboptimal variants of DTW for lowering computation, none of them is perfect. In this work, an approach to reduce computational burden of DTW has been proposed from the perspective of removing unimportant portion of the time series from computing, selected by a mask generated by edge detection algorithm commonly used in image processing or computer vision. The proposed Edge-Detectional Dynamic Time Warping (EDDTW) has been compared with original DTW by simulation experiments with 43 publicly available benchmark data sets. The simulation results show that EDDTW outperforms DTW regarding classification accuracy in more than half of the data sets, while reducing on the average 60% of the original time series leading to reduction in computational time.**

I. Introduction

The series of data representing changes across time axis and collected over a period of time is what is known as time series data. If the data correspond to multiple variables or features, the resulting series is multidimensional. Some examples of multidimensional time series (MTS) data are human gait data, online signature data, biosignals in medical field or stock price, exchange rate in financial area. In real world scenario, dynamical systems are abundant. Collection of multidimensional time series data through various sensors is becoming common for analysis of the behavior of the system dynamics. Unlike static data, analysis of time series data is not so simple and it requires special methods for representation and analysis [1].

For classification or clustering of time series data, some sort of similarity or dissimilarity measure is required. For static data, several distance metrics are available for computation of dissimilarity which cannot be used directly for temporal data. The most popular method to compare two time series is warping the time axis in order to achieve an alignment between the data points of the series. The Dynamic Time Warping (DTW) algorithm [2], first being used in speech recognition, has been shown to be an effective solution for measuring the distance between time series [3]. Unlike euclidean distance which is easier to compute, DTW allows a time series to be stretched or compressed to provide a better match with another time series and can handle time series with local time shifting and different lengths. Despite the effectiveness of DTW algorithm, it has a computational cost of $O(N^2)$ which makes it computationally difficult to use for longer time series. Several measures have been introduced to speed up DTW computations as well as to better control the possible routes of the warping path [4], [5], [6].

In this work, a method for reduction of computational cost for DTW has been proposed. The proposal here is based on the concept of selecting the important information rich part of the time series for comparison by edge detection algorithms commonly used in image processing or computer vision area for capturing changing patterns or boundary. Simulation experiments with benchmark data sets from UCR repository have been done to check the efficiency of the proposed algorithm compared to other available algorithms for reduction of computational complexity of DTW. The next section describes DTW and some other variants or modifications of DTW for decreased computation. The following section contains our approach in this paper. Section IV represents simulation experiments and results followed by last section with summarization and conclusion.

II. DTW and Fast Variants of DTW

Though there are various measures evolved from time to time for calculating similarity between two time series [7], Dynamic Time Warping (DTW) is the most popular measure. Euclid distance is the most simple but it only works for equal length time series. DTW takes care of unequal length of the two time series but the computational cost is high. Fast DTW [8], Multiscale DTW [9] or Sparse DTW are some of the popularly used approaches for reduction of computational cost

978-1-5386-5826-0/18/$31.00 ©2018 IEEE

of original DTW. Other proposals for improvement of DTW from the authors include DTW-GA [11] and DTE (Dynamic Translation Error) [12]. In the following subsections original DTW and a few modifications of DTW used for comparison has been presented.

A. Dynamic Time Warping

The dynamic time warping is one of the algorithms to measure similarity between two series [2]. It calculates the score with dynamic programming, so it takes quadratic time for the computation. DTW belongs to the group of elastic measures and works by optimally aligning the time series in temporal domain so that the accumulated cost of the alignment is minimal.The accumulated cost can be calculated by dynamic programming, recursively applying

$$D_{i,j} = f(x_i, y_j) + \min\left(D_{i,j-1}, D_{i-1,j}, D_{i-1,j-1}\right) \quad (1)$$

where $D_{i,j}$ represents the distance between x_i and y_j , ith point in x time series and jth point in y time series respectively, for $i = 1 \ldots M$ and $j = 1 \ldots N$ where M and N are the length of the time series x and y respectively and $f(x_i, y_j) = |(x_i - y_j)|$.

The pseudocode is shown as following.

Algorithm 1: DTW
Input: Time-series x[0,...,n], y[0,...,m]
Output: DTW(x, y)
Let a two dimensional data matrix S be the score of similarity measures such that S[0,...,n, 0,...,m], and i, j, are loop index, cost is an integer.

S[0, 0] := 0
for $i = 1$ to m **do**
  S[0, i] = ∞
end for
for $i = 1$ to n **do**
  S[i, 0] = ∞
end for
for $i = 1$ to n **do**
  for $j = 1$ to m **do**
    cost = f(x[i], y[j])
    S[i, j] = cost + min(S[i-1, j], S[i, j-1], S[i-1, j-1])
  end for
end for
return S[n, m]
end

B. FastDTW

FastDTW is a modification of DTW for speeding up which has linear time and space complexity. FastDTW uses a multilevel approach that recursively projects a solution from a coarse resolution and refines the projected solution. It is an approximation of original DTW that sacrifices absolute optimality in lieu of space and time. The details of the algorithm can be found in [8]. FastDTW produces results in shorter time with the little change in classification accuracy specially for larger dataset.

C. DTW-GA

DTW-GA is a new similarity measure proposed by one of the authors [11] which is a combination of Genetic Algorithm (GA) and DTW. DTW-GA use masking of time series for each class of time series with the optimum gene of GA as the representation method of time series and DTW as the comparison method of two time series. The main concept behind this new measure is that GA searches the best(optimum) gene for masking so that the most important points of the time series is used for comparison instead of the whole time series as in original DTW to reduce the computational cost. DTW-GA achieves higher accuracy and faster computation than DTW, however there are some problems; mask generation takes long time; parameter adjustment for GA is also difficult. Besides, the algorithm cannot be used for clustering as the mask generation is based on labelled samples.

III. Proposed Approach

In this section, our proposed approach, Edge-Detectional DTW is presented. As in the image processing problems, edge detection is done based on the concept of derivative where the changes take place, here also the concept is that the most changing points in the time series represent the characteristics or the feature of the time series. So one dimensional edge detection algorithm used on time series can act as a mask to extract the most important part of the time series. The details of Edge Detection algorithms are represented in [13]. DTW is used then to calculate similarity between the two reduced time series to reduce computation cost. The proposed algorithm is divided into two parts; pre-processing, and normalization. Let x, y be time-series data, t be a threshold, and k be the kernel (convolution array) for the approach. The notation $x*k$ denotes linear convolution which is used in Edge detection [13].

A. Pre-processing

First, Edge-detectional DTW extracts peaks of the time series data, the area that the amount of change is large, by using Edge-detection. The area represents the feature of the class well. The processes are shown in Fig1-4.

1) Convert the data to a sequence of absolute value of amount of change by convoluting with the kernel

$$x^k = |x * k|, y^k = |y * k|$$

2) Get time indices so that the value is equal or larger than the threshold from the sequence of step1

$$I_x = \{i \in \mathbb{N} | x_i^k > t\}, I_y = \{i \in \mathbb{N} | y_i^k > t\}$$

3) Collect the value from the original time-series data at time indices of step2

$$\overline{x} = \{x_{i \in I_x} \in x\}, \overline{y} = \{y_{i \in I_y} \in y\}$$

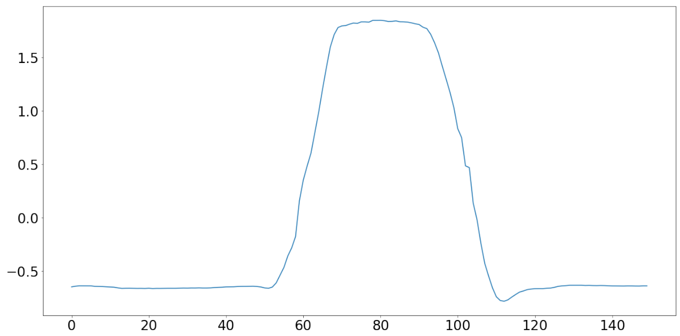

Fig. 1: Original time-series data

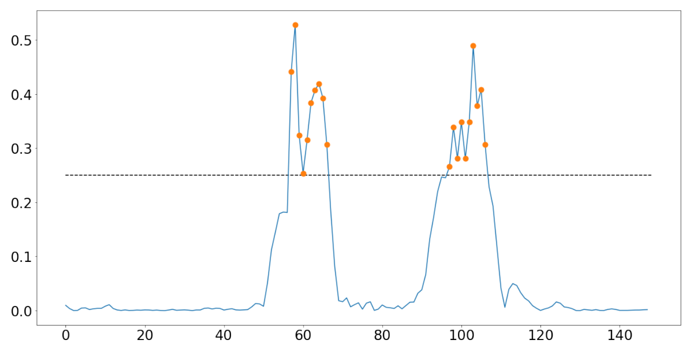

Fig. 2: Kernel convoluted data

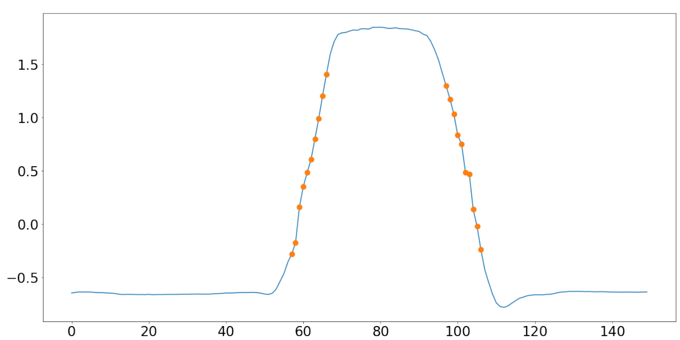

Fig. 3: Original time-series data with highlighting the peaks of change

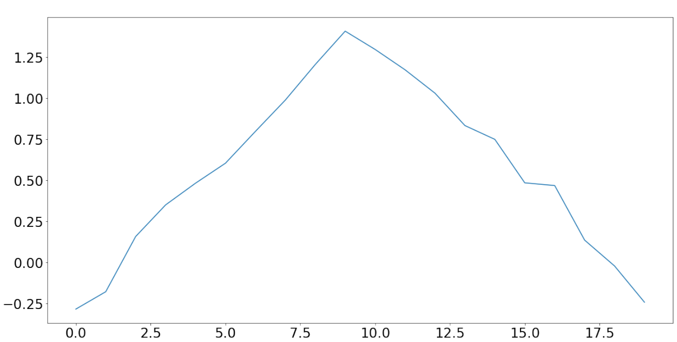

Fig. 4: Pre-processed data

B. *Normalization*

Normalize the DTW score by dividing the length of the longer sequence.

$$\frac{d(\overline{x}, \overline{y})}{max(len(\overline{x}), len(\overline{y}))}$$

The pseudocode of the algorithm is given below:

Algorithm 2: Edge-detectional DTW

Input: Time-series x[0,...,n], y[0,...,m], threshold t, kernel k

Output: EDDTW(x, y)

Vector Xk and Yk are peak-emphasized data

```
Sx = conv(x, k), Sy = conv(y, k)
for i = 0 to n do
    Sx[i] = abs(Sx[i])
end for
for i = 0 to m do
    Sy[i] = abs(Sy[i])
end for
for i = 0 to n do
    if Sx[i] > t then
        Xk.push(x[i])
    end if
end for
for i = 0 to m do
    if Sy[i] > t then
        Yk.push(y[i])
    end if
end for
return DTW(Xk, Yk) / max(len(Xk), len(Yk))
```

end

IV. Simulation Experiments

Simulation Experiments with benchmark data sets have been done with two main objectives of evaluation of our proposed measure. One of the objectives is to check the percentage reduction of the time series which is not needed for comparison and the change in classification accuracy between original DTW and our proposed measure.

These algorithms are evaluated with classification task using 1- Nearest Neighbor classifier. For Edge-detectional DTW, we chose the parameters $c = 0.25, 0.5, 1, 2$, and following kernels:

$$k_{Prewitt} = [-1,\ 0,\ 1]$$

$$k_{Sobel} = [-2,\ 0,\ 2]$$

$$k_{Laplacion} = [1,\ -4,\ 1]$$

A. *Experimental Setup*

All experiments were carried out on a Ubuntu 16.04 operated PC with a i7-6700K (4.00 GHz) processor and 64 GB main memory. The algorithm were implemented in Python 3 with Numpy.

B. Datasets

We have used following 43 datasets from the UCR Time Series Archive 2015 [14] for simulation experiments. The Table I shows the datails of the datasets.

C. Simulation Results

The experimental results are shown in Table II. 24 out of 43 datasets show improvement in classification accuracy with our proposed approach compared to DTW. The column 4 shows differences of the classification accuracy with DTW and the proposed approach EDDTW, and 5th column of Table II represents the ratio of the reduced time-series length to the length of the original time series. The length of the time series is reduced for all the data sets and classification accuracy has found better than DTW for more than half of the data sets with reduced computational time because of reduced length of the time series.

TABLE I: Overview of Datasets

Dataset	Train	Test	Time series Length
50words	450	455	270
Adiac	390	391	176
Beef	30	30	470
CBF	30	900	128
ChlorineConcentration	467	3840	166
CinC_ECG_torso	40	1380	1639
Coffee	28	28	286
Cricket_X	390	390	300
Cricket_Y	390	390	300
Cricket_Z	390	390	300
DiatomSizeReduction	16	306	345
ECG200	100	100	96
ECGFiveDays	23	861	136
FaceAll	560	1690	131
FaceFour	24	88	350
FacesUCR	200	2050	131
FISH	175	175	463
Gun_Point	50	150	150
Haptics	155	308	1092
InlineSkate	100	550	1882
ItalyPowerDemand	67	1029	24
Lighting2	60	61	637
Lighting7	70	73	319
MALLAT	55	2345	1024
MedicalImages	381	760	99
MoteStrain	20	1252	84
OliveOil	30	30	570
OSULeaf	200	242	427
SonyAIBORobotSurfaceII	27	953	65
SonyAIBORobotSurface	20	601	70
StarLightCurves	1000	8236	1024
SwedishLeaf	500	625	128
Symbols	25	995	398
synthetic_control	300	300	60
Trace	100	100	275
TwoLeadECG	23	1139	82
Two_Patterns	1000	4000	128
uWaveGestureLibrary_X	896	3582	315
uWaveGestureLibrary_Y	896	3582	315
uWaveGestureLibrary_Z	896	3582	315
wafer	1000	6164	152
WordsSynonyms	267	638	270
yoga	300	3000	426

According to the statistical test by Wilcoxon signed-rank test, the null hypothesis, "there is no difference between our approach and DTW about data length" was rejected in two-sided 95% confidence interval. So even for the data sets where classification accuracy is lower than that of original DTW, the change is not statistically significant. By adjusting parameters of our proposed approach, we could find better results for some of the data sets as shown in Table III.

For the length of all original time-series, it was reduced

TABLE II: Experimental Results

Dataset	Proposed	DTW	Diff	Ratio
SonyAIBORobotSurface	0.8220	0.7205	0.1015	0.0623
ECGFiveDays	0.8420	0.7840	0.0581	0.2044
Gun_Point	0.9333	0.8800	0.0533	0.1232
TwoLeadECG	0.9614	0.9219	0.0395	0.5078
Haptics	0.3864	0.3474	0.0390	0.9112
Coffee	1.0000	0.9643	0.0357	0.3423
Beef	0.6000	0.5667	0.0333	0.6006
ECG200	0.8400	0.8100	0.0300	0.6089
FaceAll	0.7953	0.7680	0.0272	0.6770
SwedishLeaf	0.8224	0.7984	0.0240	0.8138
Lighting2	0.8197	0.8033	0.0164	0.3062
Cricket_Y	0.7564	0.7462	0.0103	0.8781
Adiac	0.6010	0.5908	0.0102	0.4330
Trace	1.0000	0.9900	0.0100	0.0796
MALLAT	0.9237	0.9147	0.0090	0.1049
OSULeaf	0.6405	0.6322	0.0083	0.9011
ChlorineConcentration	0.6315	0.6250	0.0065	0.3424
SonyAIBORobotSurfaceII	0.8353	0.8300	0.0052	0.8349
wafer	0.9849	0.9838	0.0011	0.9421
Lighting7	0.7671	0.7671	0.0000	0.8663
FaceFour	0.8523	0.8523	0.0000	0.6445
DiatomSizeReduction	0.9608	0.9608	0.0000	0.7352
OliveOil	0.8333	0.8333	0.0000	0.7734
Two_Patterns	1.0000	1.0000	0.0000	0.8145
uWaveGestureLibrary_X	0.7278	0.7281	-0.0003	0.9096
MoteStrain	0.8874	0.8882	-0.0008	0.9060
ItalyPowerDemand	0.9466	0.9485	-0.0019	0.6393
CBF	0.9967	1.0000	-0.0033	0.9151
yoga	0.8363	0.8407	-0.0043	0.9116
uWaveGestureLibrary_Z	0.6533	0.6588	-0.0056	0.9046
FISH	0.8571	0.8629	-0.0057	0.9327
FacesUCR	0.9210	0.9268	-0.0059	0.8302
synthetic_control	0.9633	0.9733	-0.0100	0.9035
WordsSynonyms	0.6693	0.6803	-0.0110	0.9046
50words	0.7033	0.7143	-0.0110	0.9059
Cricket_Z	0.7718	0.7846	-0.0128	0.5019
Cricket_X	0.7564	0.7692	-0.0128	0.8964
Symbols	0.9377	0.9528	-0.0151	0.9176
MedicalImages	0.7342	0.7513	-0.0171	0.8921
InlineSkate	0.3473	0.3782	-0.0309	0.8019
CinC_ECG_torso	0.6486	0.6913	-0.0428	0.7831
StarLightCurves	0.7776	0.8864	-0.1088	0.0158
uWaveGestureLibrary_Y	0.4958	0.6468	-0.1510	0.1659

TABLE III: Parameters Adjusted Results

Dataset	Proposed	DTW	Diff	Ratio
CBF	1.0000	1.0000	0.0000	0.9288
ItalyPowerDemand	0.9466	0.9485	-0.0019	0.6393
FacesUCR	0.9195	0.9268	-0.0059	0.8302
MoteStrain	0.8842	0.8882	-0.0008	0.9060
FISH	0.8686	0.8629	0.0057	0.9711
StarLightCurves	0.8531	0.8864	-0.0333	0.0798
Cricket_Z	0.7846	0.7846	0.0000	0.9549
Cricket_X	0.7821	0.7692	0.0128	0.7175
MedicalImages	0.7618	0.7513	0.0105	0.5167
50words	0.7099	0.7143	-0.0044	0.8895
CinC_ECG_torso	0.6783	0.6913	-0.0130	0.9135
InlineSkate	0.3691	0.3782	-0.0091	0.9597

about 60% in average. Table IV shows combinations of the best threshold and kernel for each dataset.

TABLE IV: Best parameters

Dataset	Threshold	Kernel
SonyAIBORobotSurface	2.0	Prewitt
ECGFiveDays	0.25	Prewitt
Gun_Point	0.25	Prewitt
TwoLeadECG	1.0	Laplacian
Haptics	0.25	Laplacian
Coffee	0.25	Sobel
Beef	0.5	Laplacian
ECG200	1.0	Laplacian
FaceAll	0.5	Sobel
SwedishLeaf	0.5	Laplacian
Lighting2	0.25	Sobel
Cricket_X	0.1	Sobel
MedicalImages	0.1	Sobel
Cricket_Y	0.25	Laplacian
Adiac	2.0	Laplacian
Trace	0.25	Sobel
MALLAT	0.25	Sobel
OSULeaf	0.25	Laplacian
ChlorineConcentration	2.0	Laplacian
FISH	0.1	Laplacian
SonyAIBORobotSurfaceII	0.25	Laplacian
wafer	0.5	Laplacian
Lighting7	0.25	Laplacian
CBF	0.2	Laplacian
Cricket_Z	0.1	Laplacian
FaceFour	0.25	Sobel
DiatomSizeReduction	1.0	Laplacian
OliveOil	0.5	Laplacian
Two_Patterns	0.5	Laplacian
uWaveGestureLibrary_X	0.25	Laplacian
MoteStrain	0.25	Laplacian
ItalyPowerDemand	0.25	Prewitt
yoga	0.25	Laplacian
50words	0.3	Laplacian
uWaveGestureLibrary_Z	0.25	Laplacian
FacesUCR	0.25	Sobel
InlineSkate	0.1	Laplacian
synthetic_control	0.25	Laplacian
WordsSynonyms	0.25	Laplacian
CinC_ECG_torso	0.1	Laplacian
Symbols	0.25	Laplacian
StarLightCurves	0.1	Sobel
uWaveGestureLibrary_Y	0.25	Sobel

V. Conclusion

The growing necessity of time series data analysis in different field of applications e.g medical signal processing to weather forcasting or financial market analysis to biometric authentication techniques has led to the development of efficient time series matching algorithms. Dynamic Time Warping belonging to the group of elastic measures, is the most popular approach though the algorithm has a computational cost of quadratic order. There are many variants and modifications of DTW which tries to reduce the computation time to linear order while sacrificing the optimality of alighnemt of the two time series. In this work, a method for reduction of computational time of DTW has proposed which is by concept different from those approaches. The proposed approach EDDTW is based on the concept of removing the unimportant part while keeping the discriminatory part of the original time series to reduce computation. The selection is based on generating a mask according to the concept of edge detection in the area of image processing. This approach is similar in concept to our earlier apprroach of DTW-GA where GA (Genetic Algorithm) was exploited to generate the mask. DTW-GA has lesser computational time for classification but poorer in classification accuracy compared to EDDTW. DTW-GA also takes longer time for proper parameter selection than that of EDDTW. In the next step, proper parameter selection and further improvement in classification accuracy for larger number of data sets is to be taken care of.

References

[1] H. Kantz, T. Schreiber, *Nonlinear Time Series Analysis*, Cambridge University Press, Cambridge, UK, 2004.

[2] H. Sakoe and S. Chiba, "Dynamic program- ming algorithm optimization for spoken word recognition," in IEEE Transactions on Acous- tics, Speech, and Signal Processing, vol. 26, no. 1, pp. 43-49, Feb 1978.

[3] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series",*Proc. AAAI Workshop on Knowledge Discovery in Databases*, pp. 359–370, 1994.

[4] S. Chu, E. Keogh, D. Hart and M. Pazzani, "Iterative deepening dynamic time warping for Time series", *Proc. 2nd SIAM International Conference on Data Mining*, 2002.

[5] S. Salvador and P. Chan, " FastDTW: Toward accurate dynamic time warping in linear time and space", *Intelligent Data Analysis.*, Vol.11(5), pp. 561–580, 2007.

[6] Basabi Chakraborty and Sho Yoshida, Time series Classification with New Similarity Measure : An Application for Automatic Detection of Drivers Distraction, in Proceedings of NOLTA 2017, December, 2017.

[7] J. Serra and J. Arcios, "An empirical evaluation of similarity measures for time series classification", Knowledge based systems, Vol 67, pp. 305-314, September, 2014.

[8] S. Salvador and P. Chan, "FastDTW : Toward Accurate Dynamic Time Warping in Linear Time and Space," Intell. Data Anal., vol. 11, no. 5, pp. 561-580, 2007.

[9] E. J. Keogh and M. J. Pazzani, "Scaling up dynamic time warping for datamining applications," Proceedings of KDD, pp. 285-289, 2000.

[10] Al-Naymat, G., S. Chawla, and J. Taheri, "SparseDTW: A Novel Approach to Speed up Dynamic Time Warping", The 2009 Australasian Data Mining, vol. 101, Melbourne, Australia, ACM Digital Library, pp. 117-127, 12/2009

[11] Basabi Chakraborty and Sho Yoshida, "A novel genetic algorithm based similarity measure for time series classification", Proceedings of ITISE 2017, pp. 536–547, 2017.

[12] Basabi Chakraborty and Sho Yoshida, "Proposal of a new similarity measure for time series classification", *Proc. ITISE 2016*, Spain, 2016.

[13] J. Canny, "A Computational Approach to Edge Detection," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, no. 6, pp. 679-698, Nov. 1986.

[14] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bag- nall, A. Mueen and G. Batista (2015). The UCR Time Series Classification Archive. www.cs.ucr.edu/ eamonn/time_series_data/