Constrained DTW preserving shapelets for explainable time-Series clustering

Hussein El Amouri, Thomas Lampert, Pierre Gançarski, Clément Mallet

Highlights

Constrained DTW Preserving Shapelets for Explainable Time-Series Clustering

Hussein EL Amouri, Thomas Lampert, Pierre Ganļarski, Clément Mallet

- Introduces Constrained DTW Preserving shapelets (CDPS).

- A means for semi-supervised learning of shapelets using must-link and cannot-link constraints.

- Introduces Shapelet Cluster Explanation, including several variants, for explaining clustering partitions.

Constrained DTW Preserving Shapelets for Explainable Time-Series Clustering

Hussein EL Amouri[a,*], Thomas Lampert[a], Pierre Ganļarski[a], Clément Mallet[b]

[a] *ICube, University of Strasbourg, Illkirch-Graffenstaden, France*
[b] *Univ Gustave Eiffel, ENSG, IGN, Saint-Mande, France*

Abstract

The analysis of time series is becoming ever more popular due to the proliferation of sensors. A well-known similarity measure for time-series is Dynamic Time Warping (DTW), which does not respect the axioms of a metric. These, however, can be reintroduced through Learning DTW-Preserving Shapelets (LDPS). This article extends LDPS and presents constrained DTW-preserving shapelets (CDPS). CDPS directs the time-series representation to captures the user's interpretation of the data by considering a limited amount of user knowledge in the from of must-link- cannot link constraints. Subsequently, unconstrained algorithms can be used to generate a clustering that respects the constraints without explicit knowledge of them. Out-of-sample data can be transformed into this space, overcoming the limitations of traditional transductive constrained-clustering algorithms. Furthermore, several Shapelet Cluster Explanation (SCE) approaches are proposed that explain the clustering and can simplify the representation while preserving clustering performance. State-of-the-art performance is demonstrated on multiple time-series datasets and an open-source implementation will be made publicly available upon acceptance.

Keywords: Shapelets, Semi-supervised Learning, Constrained Clustering, Time-series, Representation Learning

1. Introduction

Time series data volume and complexity have grown as sensing technology has advanced. This makes obtaining accurate labels challenging, leading to a preference for unsupervised clustering over supervised classification. Unsupervised approaches, however, can be unreliable due to algorithmic bias. In contrast, semi-supervised approaches incorporate limited expert knowledge, in the form of constraints, to produce more reliable results.

Constraints can have many forms and in this work must-link (ML) and cannot-link (CL) constraints are considered. These constraints do not define a class sample, instead they indicate that pairs are similar (must-link), hence belong to the same cluster, or not (cannot-link). This allows meaningful results to be obtained without exhaustive explicit labeling of all the samples.

Generally, time series exhibit phase shift, warping, and distortions [1], rendering Euclidean distance ineffective [2]. Various similarity measures for time series have been proposed, e.g. compression-based measures [3], Levenshtein Distance [4], Longest Common Subsequence [5], and Dynamic Time Warping (DTW) [6, 7]. DTW is often used to overcome distortions by aligning time series and it has been shown to be a good basis for calculating embeddings [8, 2]. However, analysing the results and data distributions based on the DTW similarity is challenging since it is not a metric.

Shapelets [9] are capable of extracting task-specific features that are phase-independent discriminative sub-sequences, and map time series into this representational space. Learning DTW-Preserving Shapelets (LDPS) [10] extends them to learn representations that model DTW similarity. To improve the representational space of LDPS by integrating user knowledge, Constrained DTW Preserving Shapelets (CDPS) [11], presented herein[1], extends LDPS to a semi-supervised approach by integrating ML and CL constraints, see Figure 1. This affects the space by moving ML points closer together and CL points further apart. Any subsequent unconstrained algorithm applied to the embedding therefore fulfills the constraints without explicit knowledge of them. Moreover, it is often desirable that machine learning algorithms provide explanations of their results [12]. Since shapelets are interpretable and define an Euclidean mapping, their importance for clustering results can be determined to provide a clustering explanation.

This article's main contributions are:

- the proposed Constrained DTW-Preserving Shapelets (CDPS) method for time-series representation learning is comprehensively detailed;
- a cluster explanation process that takes advantage of the shapelets' interpretability, called Shapelet Clustering Explanation (SCE).

CDPS's representational embedding, and therefore the constaints, is generalisable to out-of-sample data, overcoming the

*Corresponding author
Email addresses: helamouri@unistra.fr (Hussein EL Amouri), lampert@unistra.fr (Thomas Lampert), gancarski@unistra.fr (Pierre Ganļarski), clement.mallet@ign.fr (Clément Mallet)

[1] This article is an extension of "CDPS: Constrained DTW Preserving shapelets " [11].

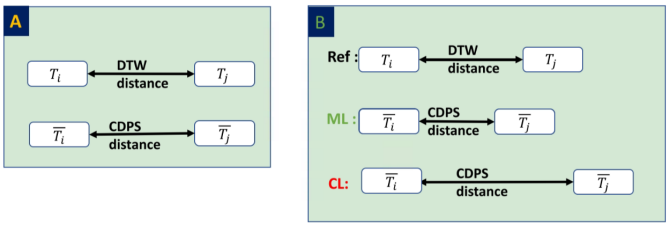


Figure 1: (A) LDPS models the DTW distance between time-series pairs, (B) CDPS extends this by integrating user knowledge to reduce (increase) the distance when points are under must-link (cannot-link) constraint.

restrictions of typical constrained-clustering techniques such as COP-KMeans [13]. Furthermore, such advances allow for the development of novel interactive and active time-series constrained clustering processes, which need vectorial data representations to measure notations of density, consistency [14] and informativeness [15].

This article is structured as follows: Section 2 reviews related work; Section 3 outlines CDPS for explainable clustering; Section 4 compares CDPS to (un)constrained algorithms and presents a study of SCE; Section 5 discusses the results; and Section 6 concludes the study.

2. Related Work

2.1. Dynamic Time Warping - DTW

DTW is a similarity measure that finds an optimal alignment between two time series. It warps the time dimension of one time series to match that of another [16]. The best alignment corresponds to the minimum cumulative distance between corresponding elements of the time series, where an element of T_i (T_j) at instant e (e') is represented as $T_{i,e}$ ($T_{j,e'}$). The DTW similarity is the cost of the optimal alignment path. The cost between any pair of sequences' elements can be written as:

$$DTW(T_{i,e}, T_{j,e'}) = D(T_{i,e}, T_{j,e'}) + \min \begin{cases} DTW(T_{i,e-1}, T_{j,e'-1}), \\ DTW(T_{i,e}, T_{j,e'-1}), \\ DTW(T_{i,e-1}, T_{j,e'}). \end{cases}$$

Efforts to reduce DTW's computational complexity have been made [17].

2.2. Shapelets

Shapelets are sub-sequences of time-series and were originally developed for to discrimination using a tree based classifier [9, 18]. Originally shapelets were sub-sequences of the time-series being studied, however searching for them is expensive and much of the literature focuses on efficiency improvements. Lines et al. [19] and Hills et al. [20] first proposed to separate representation learning from classification, and introduced the shapelet transform, by which the raw data is mapped into a vectorial representation, the bases of which is defined by the shapelets. In order to overcome the exhaustive search for optimal shapelets, Grabocka et al. [21] introduce supervised shapelet learning, in which shapelets are learnt by minimising an objective function. Shah et al. [22] learn more pertinent and representative shapelets for classification by using DTW similarity.

Zakaria et al. [23] introduced u-shapelets, the first shapelet-based approach to clustering time-series. U-shapelets are those that best partition a subset of time series from the dataset, however, they are not learnt and instead are searched for. Zhang et al. [24] propose combining learning shapelets and unsupervised feature selection and, amongst others Paparrizos et al. [2], learning DTW-preserving shapelets (LDPS) [10] broadens the learning paradigm further. LDPS models the distribution of DTW similarity between the time series, which allows for better discrimination. To learn the shapelets, the loss function $\mathcal{L} = \frac{1}{2}(DTW(T_i, T_j) - \beta \cdot \|\overline{T}_i - \overline{T}_j\|)^2$ defined over two time series T_i and T_j is minimised, where $\overline{T}_i$ ($\overline{T}_j$) is the shapelet transform of the time series T_i (T_j) and β scales the embedding time-series distance. During training, $\mathcal{L}$ is averaged over all possible pairs of time series.

2.3. Constrained Clustering

Constrained clustering algorithms enhance the clustering process by integrating expert knowledge. Constraints can be given in different forms, however, this works focuses on instance level constraints that define how samples must and cannot be linked together.

Many constrained clustering algorithms have been proposed, some of which have been adapted to time-series [8]. COP-KMeans is a parameter-free extension to K-means that often offers state-of-the-art performance [8]. Cluster allocations are validated using the constraint set at each iteration to verify that no constraints are violated, if so the algorithm halts (without producing a result). Mixed integer programming k-Means (MIP-KMeans) [25] implements a branch-and-bound search for the optimal cluster assignments while fulfilling the constraints. More recently deep embedded clustering was extended to form Deep Constrained Clustering (DCC) [26] by integrating constraints into the loss function and has been extended to time-series [27]. FeatTS [28] takes a different approach to semi-supervised time-series clustering. It represents the data as a graph and uses a percentage of labeled samples to extract features relevant to calculate its co-occurrence matrix, with which the data is clustered. Other unsupervised methods, e.g. k-shape [29], exist but are beyond the scope of this article.

2.4. Explainable Clustering

The majority of clustering algorithms present in the literature leave it to the expert to explain the partitions. Identifying the necessary features for this task can be slow and subjective, if at all possible. Conceptual Clustering [30] aims to automatically learn an partitioning's explanation by identify concepts, justifications, and explanations [30]. Despite the fact that shapelets are interpretable, there is limited research on using them to explain time-series clustering. Effort has been made, however, to identify shapelets that best retrieve a time series [31] and classifiy wake and sleep states [32].

3. Constrained DTW Preserving Shapelets for Explainable Clustering

The necessary terminology and notations are first defined, the Constrained DTW preserving shapelet (CDPS) algorithm is then presented in Section 3.1, and the Shapelet Cluster Explanation (SCE) in Section 3.2.

Definition 1 (Time-series). An ordered set of L_i real-valued observations $T_i = T_{i,1}, \ldots, T_{i,L_i}$. (uniform length is assumed for simplicity but not required). A set of N univariate[2] time-series is given by $\mathcal{T} = \{T_1, T_2, \ldots, T_N\}$.

Definition 2 (Shapelets). An ordered set of real-valued variables whose length is less than or equal to that of the dataset's smallest time-series. Let S be a shapelet having length L_k. A set of K shapelets is denoted as $\mathcal{S} = \{S_1, \ldots, S_K\}$, such that $S_k = S_{j,1:L_k}$. Although, $\mathcal{S}$ typically contains different length shapelets, for simplicity, uniform shapelets are assumed.

Definition 3 (Euclidean Shapelet Match). Represents the matching score between a shapelet S_k and a time series T_i, such that

$$\overline{T}_{i,k} = \min_{m \in \{1:L_i - L_k + 1\}} D_{i,k,m}. \tag{1}$$

where $D_{i,k,m}$ is the Squared Euclidean Score between S_k and T_i at m [11].

Definition 4 (Shapelet transform). Maps T_i to a new vectorial representation, $\overline{T}_i$, using Euclidean shapelet match with respect to $\mathcal{S}$, such that

$$\overline{T}_i = \{\overline{T}_{i,1}, \ldots, \overline{T}_{i,K}\}. \tag{2}$$

Definition 5 (Constraint Sets). Let C_k be the k^{th} cluster, ML (CL) be a set of time series indexes connected by must-link (cannot-link) constraints. Thus, T_i, T_j such that $i, j \in \{1, \ldots, N\}$ and $i \neq j$, we have

$$\text{ML} = \{(i,j) | \forall\ k \in \{1, \ldots, K\}, T_i \in\ C_k \Leftrightarrow T_j \in C_k\}, \tag{3}$$

$$\text{CL} = \{(i,j) | \forall\ k \in \{1, \ldots, K\}, \neg(T_i \in\ C_k \wedge T_j \in C_k)\}. \tag{4}$$

3.1. CDPS – Constrained DTW Preserving Shapelets

A new objective function is introduced based on contrastive learning [33] that allows an expert to influence the learning process, while preserving DTW similarity and interpretability of the resulting shapelets. The proposed loss between two time-series is defined such that

$$\mathcal{L}(T_i, T_j) = \frac{1}{2}\left(DTW(T_i, T_j) - \beta \cdot Dist_{i,j}\right)^2 + \phi_{i,j}, \tag{5}$$

where $DTW(T_i, T_j)$ is the dynamic time warping similarity between time-series T_i and T_j, $Dist_{i,j} = \|\overline{T}_i - \overline{T}_j\|_2$ is the L_2 distance between T_i and T_j in the embedded space, and β scales the time-series distance in the embedded space to the corresponding DTW similarity. The first term is the LDPS loss and $\phi_{i,j}$ incorporates the expert constraints into the learning process, such that

$$\phi_{i,j} = \begin{cases} \alpha \cdot Dist_{i,j}^2, & \text{if } (i,j) \in \text{ML}, \\ \gamma \cdot \max(0, w - Dist_{i,j})^2, & \text{if } (i,j) \in \text{CL}, \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

where α, γ regularise the ML and CL similarity distances respectively, and w is the minimum embedded distance between samples for them to be considered well separated. To bound this distance, 'well separated' is taken to be the maximum DTW distance between two points in the original dataset, and so this term is calculated such that

$$w = \max_{\forall i, \forall j \neq i} (DTW(T_i, T_j)) + \log\left(\frac{DTW(T_i, T_j)}{\max_{\forall i, \forall j \neq i}(DTW(T_i, T_j)}\right). \tag{7}$$

Consequently, the overall loss function is defined to be

$$\mathcal{L}(\mathcal{T}) = \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N-1} \mathcal{L}(T_i, T_j). \tag{8}$$

Algorithm 1 Constrained Distance Preserving Shapelets

Input: $\mathcal{T}$, ML, CL, $L_{\min}$, $S_{\max}$, n_{epochs}, s_{batch}, c_{batch}
Output: S, Embeddings.

```
ShapeletBlocks ← Get_Shapelet_Blocks(L_min, S_max, L_i)
Shapelets ← Initialize_Shapelets(ShapeletBlocks)
for i ← 0 to n_epochs do
    for 1 to |T|/s_batch do
        minibatch ← Get_Batch(T, ML, CL, s_batch, c_batch)
        Compute the DTW between the T_i's and T_j's in minibatch
        Update the Shapelets and β by descending the gradient ∇L(T_i, T_j)
Embeddings ← Shapelet_Transform(T)
```

Algorithm 1 describes the CDPS approach to learning the representational embedding. ShapeletBlocks is a dictionary containing $S_{\max}$ (maximum number of shapelets blocks) pairs, {shapelet length; shapelet number}, where shapelet length is $L_{\min} \cdot b_{\text{ind}}$, $L_{\min}$ is the minimum shapelet length and $b_{\text{ind}} \in \{1, \ldots, S_{\max}\}$ is the index of the shapelet block. The number of shapelets for each block is calculated using the same approach as LDPS [10]: $10 \times \log(L_i - L_{\min} \cdot b_{\text{ind}})$, where L_i is the length of the time series. Initialize_shapelets initialises the shapelets in a rule-based manner: (1) shapelet prototypes are drawn from the time series; (2) K-means clustering groups them and the centroids form the initial shapelets. Get_Batch($\mathcal{T}$, ML, CL, s_{batch}, c_{batch}) returns batches containing s_{batch} samples including c_{batch} constrained samples. In this way, constrained time-series are frequently observed during training (they are repeated if necessary). To take advantage of GPU acceleration, the above algorithm can be implemented as a 1D CNN in which each layer represents a shapelet block followed by maxpooling in order to

[2] To facilitate clarity, nevertheless, CDPS may be used with multivariate time-series.

obtain the embeddings. The derivation of CDPS' gradient of $\mathcal{L}(\mathcal{T})$, $\nabla\mathcal{L}(T_i, T_j)$ (Algorithm 1, Line 7), is given in Appendix F.

3.2. SCE – Shapelet Clustering Explanation

This process is inspired by previous work on shapelet discovery and categorisation [9] in which information gain is used to measure shapelet quality. Information gain is one of several quality measures, however, it has been shown [20] that there is no significant difference between them. Furthermore, information gain, which is based on entropy, is the only that results in a partitioning of the data into two groups, which models the clustering process, lending itslef to the work described herein.

Definition 6 (Entropy). Measures the randomness of a given set of class labels: homogeneous cluster labels gives zero entropy, and a uniformly random sample gives the maximum value. Assuming a dataset $\mathcal{T}$ with $\updownarrow$ classes and N instances, each class c_i contains n_i samples, and the class probability is $p(c_i)$. With ln the natural logarithm, $\mathcal{T}$'s entropy is defined, such that

$$E(\mathcal{T}) = -\sum_{i=1}^{l} p(c_i)\ln(p(c_i)), \quad s.t. \begin{cases} \sum n_i = N, \\ p(c_i) = \dfrac{n_i}{N}. \end{cases} \tag{9}$$

Definition 7 (Information Gain). Measures the reduction of entropy, i.e. information gained, in a dataset after being partitioned into distinct groups based on a splitting criteria (e.g. a threshold). The Information Gain (IG) of a dataset $\mathcal{T}$ with split ζ is defined such that $IG(\mathcal{T}, \zeta) = E(\mathcal{T}) - E(\mathcal{T})|_\zeta$, where $E(\mathcal{T})|_\zeta$ is the entropy after the split. Suppose D is partitioned into two sets $\mathcal{T}_l$ and $\mathcal{T}_r$ based on the split ζ,

$$E(\mathcal{T})|_\zeta = \frac{N_l}{N}E(\mathcal{T}_l) + \frac{N_r}{N}E(\mathcal{T}_r), \quad s.t. \begin{cases} N_l = |\mathcal{T}_l|, \\ N_r = |\mathcal{T}_r|. \end{cases} \tag{10}$$

The CDPS learned shapelets are ranked using IG according to their ability to reproduce (i.e. explain) a clustering, termed 'Global-Wise' explanation (GE), or a specific cluster, termed 'Cluster-Wise' explanation (CE). The term 'reproduce' refers to the ability to recreate the clustering output (GE) or a specific cluster therein (CE), when the time-series dataset is projected into the space defined by one or more shapelets (ranked according to their IG) by linear separation. More specifically, given a set of N time-series $\mathcal{T}$, a set of K learned shapelets $\mathcal{S}$, and a partitioning of $\mathcal{T}$ into M clusters $\updownarrow = \{C_1, \ldots, C_M\}$ (i.e. the result of an unsupervised clustering algorithm applied to the CDPS learnt representation), the goal is to rank the shapelets in importance for giving the best CE using C_i (GE using $\updownarrow$). In this manner if a certain subset of shapelets is highly representative of C_i ($\updownarrow$), i.e. is highly ranked by IG, it allows the user to visually perceive how the cluster (clustering) came to be formed. Furthermore, once the shapelets have been ranked accordingly, the best d-shapelets, $\mathcal{S}_d$ ($|\mathcal{S}_d| = d$) can be adjusted according to further user needs (e.g. to reduce dimensionality, remove duplicate shapelets, etc). It should be noted that in the following formulations, ζ is an operator that partitions the space into M partitions if GE is employed or two partitions if CE is used, i.e. the partitioned space is denoted by $D = \{D_m\}_{m=1\ldots M}$ such that D_m should be representative of C_m. Thus, with CE, a thresholding criterion is used for ζ. While with GE, Linear Discriminative Analysis (LDA) is used for ζ, in which the cluster assignments $\updownarrow$ are treated as class labels and LDA splits the data according to the linear decision boundaries found.

Several approaches for ranking the shapelets, irrespective of the objective (either CE or GE), can be identified depending on the desired result.

Independent. In which the IG of each shapelet is examined independently. Hence, $\mathcal{S}_d$ is defined such that $\mathcal{S}_d = \{S_j \mid IG(\mathcal{T}, \zeta^j) > IG(\mathcal{T}, \zeta^i)\}$, where $IG(\mathcal{T}, \zeta^i) > IG(\mathcal{T}, \zeta^{i-1})\ \forall S_i, S_j, S_{i-1} \in \mathcal{S},\ S_i \neq S_j$, such that $\zeta^i = \zeta(\mathcal{T}, S_i)$ partitions the dataset $\mathcal{T}$ in the space defined by shapelet S_i into D_i. Similarly $\zeta^j = \zeta(\mathcal{T}, S_j)$ and $\zeta^{i-1} = \zeta(\mathcal{T}, S_{i-1})$ partition the dataset $\mathcal{T}$ in the space defined by shapelets S_j and S_{i-1} into $\mathcal{D}_j$ and $\mathcal{D}_{i-1}$ respectively. In this way, all shapelets are ranked according to their IG.

Combined. Ranking the shapelets independently does not guarantee that the combination of the top-d shapelets maximises IG. Finding this is NP-Hard and the combined approach therefore uses an exhaustive recursive search strategy to find the shapelet that adds the highest IG[3], S_l, at each iteration to build the d-dimensional space, such that

$$\mathcal{S}_d^j = (\mathcal{S}_d^{j-1}, S_l), \quad \text{where} \begin{cases} \mathcal{S}_d^0 = (\phi), \\ IG(\mathcal{T}, \zeta_d^j) > IG(\mathcal{T}, \zeta_d^{j-1,i}), \end{cases} \tag{11}$$

and where $\mathcal{S}_d^j$ is the ordered list of the j best shapelets found so far, such that $\zeta_d^j = \zeta(\mathcal{T}, \mathcal{S}_d^j)$ and $\zeta_d^{j-1,i} = \zeta(\mathcal{T}, (\mathcal{S}_d^{j-1}, S_i))$ partition the dataset $\mathcal{T}$ in the space defined by $\mathcal{S}_d^j$ and $(\mathcal{S}_d^{j-1}, S_i)$ into $\mathcal{D}_j$ and $\mathcal{D}_{j-1,i}$ respectively, $l, i \in \{1, \ldots, K\}$, $i \neq l$, and $S_i, S_l \in \mathcal{S}, S_i \notin \mathcal{S}_d^j$.

Successive. Besides optimising linear separability and modelling the clustering result, the combined strategy does not impose diversity in the shapelet ordering. For example, assume that a shapelet is selected using the combined strategy and has the same IG to the previously discovered set of shapelets, this equality in IG does not necessarily imply that the selected shapelet is diverse nor increases separation in the space. The successive strategy instead calculates the information gain using samples whose partitioning differs or are incorrectly identified, i.e. $\delta_j \subset \mathcal{T}$, after adding, S_l, at each iteration, such that $\delta_j = \Delta_{j,j-1} \cup \iota_j$, $\delta_j' = \Delta_{(j-1,i),j-1} \cup \iota_{j-1,i}$, where Δ indicates the difference between the current partitioning $\mathcal{D}_j$ and the previous partitioning $\mathcal{D}_{j-1}$, i.e.

$$\Delta_{j,j-1} = \bigcup_{m=1}^{M} \mathcal{D}_{j,m} \setminus \mathcal{D}_{j-1,m}, \quad \Delta_{(j-1,i),j-1} = \bigcup_{m=1}^{M} \mathcal{D}_{(j-1,i),m} \setminus \mathcal{D}_{j-1,m}, \tag{12}$$

[3]The information gain is calculated based on the partitioning returned by linear decision boundaries, i.e. hyper-planes.

Algorithm 2 GS-SCE: Global Successive Shapelet Clustering Explanation

Input: *dists*, a distance matrix between (shapelets, time-series). $\updownarrow$, clustering predictions for the time-series.

Output: $\mathcal{S}_d$, Shapelets ordered by IG. IG, Information gain of $\mathcal{S}_d$.

1: $S_j, IG_j^*, \mathcal{D}_{j-1} \leftarrow$ GetBestIndependentShapelet(*dists*, $\updownarrow$)
2: $\mathcal{S}_d,\ IG \leftarrow [S_j],\ [IG_j^*]$
3: $S,\ K, N \leftarrow GetShapeletsIndicesAndSize(dists)$
4: **for** $j = 2, \ldots, K$ **do**
5: $\quad IG_j^* \leftarrow 0$
6: $\quad$ **for** $S_l \in S$ *and* $S_l \notin \mathcal{S}_d$ **do**
7: $\quad\quad$ embvector $\leftarrow$ transpose(*dists*)$[\mathcal{S}_d, S_l]$
8: $\quad\quad \mathcal{D}_j \leftarrow$ get_Partitions_By_Shapelets(embvector, $\updownarrow$)
9: $\quad\quad \delta_j \leftarrow \Delta_{j,j-1} \cup \iota_j$
10: $\quad\quad \mu_j \leftarrow NumberOfCorrectPts(\delta_j)$
11: $\quad\quad IG_j \leftarrow IG(\delta_j, [\mathcal{S}_d, S_l]) \cdot \frac{\mu_j}{N}$
12: $\quad\quad$ **if** $IG_j^* < IG_j$ **then** $S_j,\ D_{j-1},\ IG_j^* \leftarrow S_l, D_j,\ IG_j$
13: $\quad \mathcal{S}_d,\ IG \leftarrow [\mathcal{S}_d, S_j],\ [IG, IG_j^*]$

and ι indicates the miss-identified points in $\mathcal{D}_j$, such that

$$\iota_j = \{t \mid t \in \mathcal{D}_{j,m}, t \notin C_m\}, \quad \iota_{j-1,i} = \{t \mid t \in \mathcal{D}_{(j-1,i),m}, t \notin C_m\}, \tag{13}$$

where $\mathcal{D}_j = \{D_{j,m}\}_{m=1\ldots M}$ is the partitioning of the points in the space defined by $\mathcal{S}_d^j$ (and $\mathcal{D}_0 = \mathcal{T}$) and $\mathcal{D}_{j-1,i} = \{D_{(j-1,i),m}\}_{m=1\ldots M}$ is the partitioning in the space defined by $(\mathcal{S}_d^{j-1}, S_i)$ such that $S_i \notin \mathcal{S}_d^j$, i.e. all the shapelets that do not add the maximal IG when combined with $\mathcal{S}_d^{j-1}$. Therefore, the best shapelet at iteration j can be found such that $\text{IG}(\delta_j, \zeta_\delta^j)\frac{\mu_j}{N} > \text{IG}(\delta_j', \zeta_{\delta'}^{j-1,i})\frac{\mu_{j-1}}{N}$, where $\zeta_\delta^j = \{\delta_{j,m}\}_{m=1\ldots M}$ and $\zeta_{\delta'}^{j-1,i} = \{\delta_{j,m}'\}_{m=1\ldots M}$ are the partitions of δ_j and δ_j' respectively, and μ is the number of points correctly identified in $\mathcal{D}_j$ and incorrectly in $\mathcal{D}_{j-1}$. The scaling factor $\frac{\mu}{N}$ weights the shapelets proportionally to the number of correctly partitioned points. As such, it constructs a space using more diverse shapelets, aiding explainability.

With this in place, several combinations of SCE measures can be proposed, e.g. Global Combined SCE, GC-SCE; Local Independent SCE, LI-SCE, etc. As an example, Algorithm 2 describes the Global Successive SCE (GC-SCE) approach, some specific details are given below.

Line 1 the shapelet that best splits the dataset into $M = |\updownarrow|$ partitions is found based on the independent approach. Its index S_l, corresponding IG^* and partitions $\mathcal{D}_{j-1}$ are stored.

Line 6 The `for` loop iterates through the shapelet indices in S but not in $\mathcal{S}_d$, it therefore iterates from the lowest to the highest embedding dimensions.

Line 7 "embvector" is the embedding derived from the current d-shapelets.

Line 8 GET_PARTITIONS_BY_SHAPELETS returns the best possible partitioning by applying ζ to the data-points obtained via embvector.

Line 9 Stores misidentified points and those whose partition differs.

Line 10 The number of correct points in δ_j is returned.

Line 11 The information gain defined in the successive case is calculated.

In order to achieve the Cluster-Wise explanation (CE), binary labels are used, i.e. the cluster labels are C_i (the cluster being explained) and $\overline{C_i}$ (all other clusters), to rank the shapelets that best represent a specific cluster. When Global-Wise explanation (GE) is concerned the original clustering predictions, $\updownarrow$, are used as the labels. The algorithms of all the SCE approaches (Global Successive, Global Combined, etc) follow the same structure as Algorithm 2 and are presented in Appendix C and the experimental differences are explored in Appendix D (the results derived from Global Successive-SCE will be presented in Section 4.3). Note, that these strategies are applicable to any shapelet-based method, irrespective of the learning process.

4. Evaluation

4.1. Experimental Setup

Unsupervised (K-Means, and DCC without constraints) and semi-supervised clustering algorithms (COP-KMeans [13], MIP-Kmeans [25], and DCC [26]) are used as baselines.Since CDPS takes constraints during training, none are given to the algorithms that uses CDPS' representation.

The study is carried on the UCR repository [34, 35] and normalised Mutual Information (NMI) is used to measure clustering performance, where 0 signifies no mutual information and 1 a perfect correlation with the ground truth labels. Due to the computational complexity of COP-Kmeans and MIP-Kmeans, thirty-five univariate and fifteen multivariate datasets were chosen at random. Since CDPS and DCC are faster, their performance on all the datasets was evaluated and is presented in Appendix E. Details of the datasets are provided in Appendix G

The number of clusters for each algorithm is set to the number of classes, and the datasets are standardised with respect to time.

Two use-cases are evaluated. In the first, termed Transductive Clustering, the training and test sets (as given by the repository) are combined, this reflects the real-world case in which a dataset is to be explored and knowledge extracted. In the second, termed Inductive Clustering, the embedding is learnt on the training set and its generalisation is evaluated on the test set. This inductive use-case is not normally possible with constrained clustering algorithms since clustering is a transductive operation. This highlights a key contribution of CDPS, its ability to generalise constraints to unseen data.

The algorithms are evaluated with varying constraint levels, represented as a percentage of constrained samples: 5%, 15%, 25%. These represent a very small fraction of the total possible constraints, $\frac{1}{2}N(N-1)$. Experiments are repeated 10 times with different random constraint sets, and each algorithm is repeated 10 times per constraint set (i.e. 100 repetitions for each con-

straint percentage[4]). Constraints are generated by randomly selecting two samples, and adding a constraint (ML/CL) depending on their class labels until the desired number of constraints is reached.

Training is carried out using mini-batch gradient descent, $s_{\text{batch}} = 64$ and $c_{\text{batch}} = 16$ for the transductive setting, and $s_{\text{batch}} = 32$ and $c_{\text{batch}} = 8$ for the inductive setting (since there are fewer samples). Experiments were performed on the high-performance computing cluster provided by the University of Strasbourg. The influence of α and γ on accuracy were evaluated and the algorithm showed stability in most cases, leading to a fixed value of 2.5 for both. The minimum shapelet length $L_{min} = 0.15 \cdot L_i$, and the maximum number of shapelets $S_{max} = 3$ are taken following those in LDPS [10]. Appendix B presents a detailed explanation of the parameters' sensitivity. All models are trained for 500 epochs using the Adam optimiser. The DCC implementation and parameter values were taken from [27].

4.2. Clustering

4.2.1. Transductive Clustering

Figure 2 summarises the Tansductive results (see Table E.2 for details). It shows the NMI scores for CDPS (Euclidean K-means performed on the CDPS embeddings) compared to the other algorithms applied to the raw time-series (COP-Kmeans, MIP-Kmeans, and DCC), LDPS (CDPS with no constraints reduces to LDPS) and (unconstrained) K-means are presented as a basis for constrained algorithms to study how constraints benefit each.

As expected LDPS and K-means offer similar performance, see Figure 2a, although some datasets do favour the unconstrained K-means algorithms, outperforming LDPS. Nevertheless, CDPS efficiently uses the information given by constraints to outperform the other algorithms in almost all the different constraint fractions and datasets. Its performance increases with the number of constraints, whereas COP-KMeans and MIP-Kmeans tends to stagnate. This can be seen as the cloud of points move upwards (CDPS' NMI score increases) as more constraints are given. DCC tends to work better for the datasets that have low NMI scores for COP-Kmeans and MIP-Kmeans. Indicating complex or similar structures with which DTW struggles to capture the detailed differences. In contrast, DCC is able to find more discriminative features beyond the structural similarity of the time series, resulting in improved performance. We can also observe that for some datasets the constrained algorithms behave similarly with 5% constraints, where COP-Kmeans and MIP-Kmeans have almost similar performance while DCC is a bit worse. Again CDPS benefits the most from increasing the number of constraints and significantly outperforms the comparison algorithms with larger constraint percentages.

Overall COP(MIP)-Kmeans perform similarly, however MIP-Kmeans was able to converge on datasets for which COP-Kmeans was not. Furthermore, CDPS exploits constraints better than other algorithms, and DCC seems to exploit them the least. For some datasets all algorithms perform equally (see top right and bottom left corners) which can be due to the fact that the datasets are either very difficult (hence low NMI) or very easy to cluster.

4.2.2. Inductive Clustering

Figure 3 summarises the Inductive results (see Table E.3 for details), i.e. embedding learnt on the training set, generalised to test set. Note that for the same constraint percentage, there are significantly fewer constraints than in the transductive setting. It can therefore be concluded that even with fewer data and constraints, CDPS is still able to learn a generalisable representation and attain (within a certain margin) the same clustering performance than with the merged datasets. This is probably explained by the fact that having fewer samples and constraints means they are repeated in the mini-batches (see Section 3.1). Allowing CDPS to focus on learning shapelets that are highly discriminative and preserve DTW rather than shapelets that model larger numbers of time series. Thus, the resulting representational space better adheres to the constraints, allowing better clustering of unseen time-series. It can also be observed that COP-Kmeans tends to struggles with a small number of constraints, since it increases the risk of constraint violations, while MIP-Kmeans is able to overcome this. In this setting, DCC outperforms both COP-KMeans and MIP-KMeans for some of the challenging datasets (low NMI – lower left corner), and shows similar performance to the other algorithms on less complex data (top right corner).

4.3. Shapelet Cluster Explanation

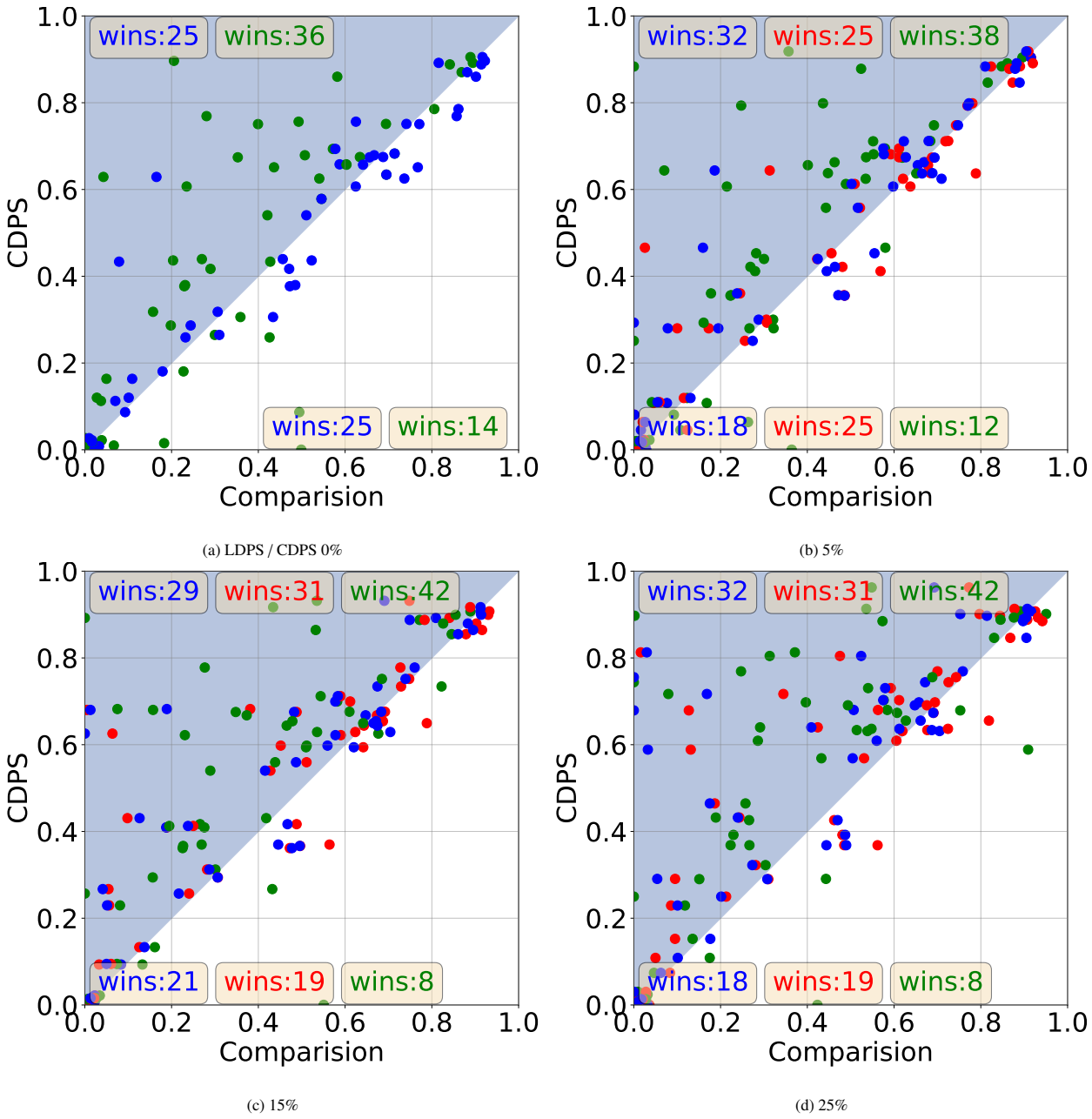
A case study of GS-SCE is presented herein (Appendix D presents the results for GI-SCE and GC-SCE). The Synthetic Control dataset is used, which contains 600 samples, representing synthetically generated control charts of length 60 and six categories: 1-Normal, 2-Cyclic, 3-Increasing trend, 4-Decreasing trend, 5-Upward shift, and 6-Downward shift. Figure 4 shows two instances from each category. The Transductive approach is used, and the aim is to provide insight and explanation of the clustering result. The GS-SCE algorithm, described in Algorithm 2, takes clustering labels and transformed time series as input and returns a shapelet ranking based on the information added by each.

Figure 5 presents the normalised cumulative information gain (NCIG) (scaled between 0 and 1). After a certain number of shapelets NCIG plateaus, indicating that any further shapelets do not improve the space relative to the clustering partitioning In this case, dimensionality reduction up to the number of shapelets indicated by the elbow can be considered. In some cases, information gain may not plateau, indicating that all shapelets are necessary.

Figure 6 presents the best 3 shapelets, the order-lines[5] for each shapelet, and the scatter plots when used with 2 and 3 shapelets.These 3 shapelets represent an additive information

[4]Despite multiple initialisations, some COP-KMeans results did not converge when applied to certain constraints sets, resulting in fewer repetitions.

[5]An order-line is simply a representation of the distance between the time-series and the shapelet ordered from lowest to highest distance.



(a) LDPS / CDPS 0%

(b) 5%

(c) 15%

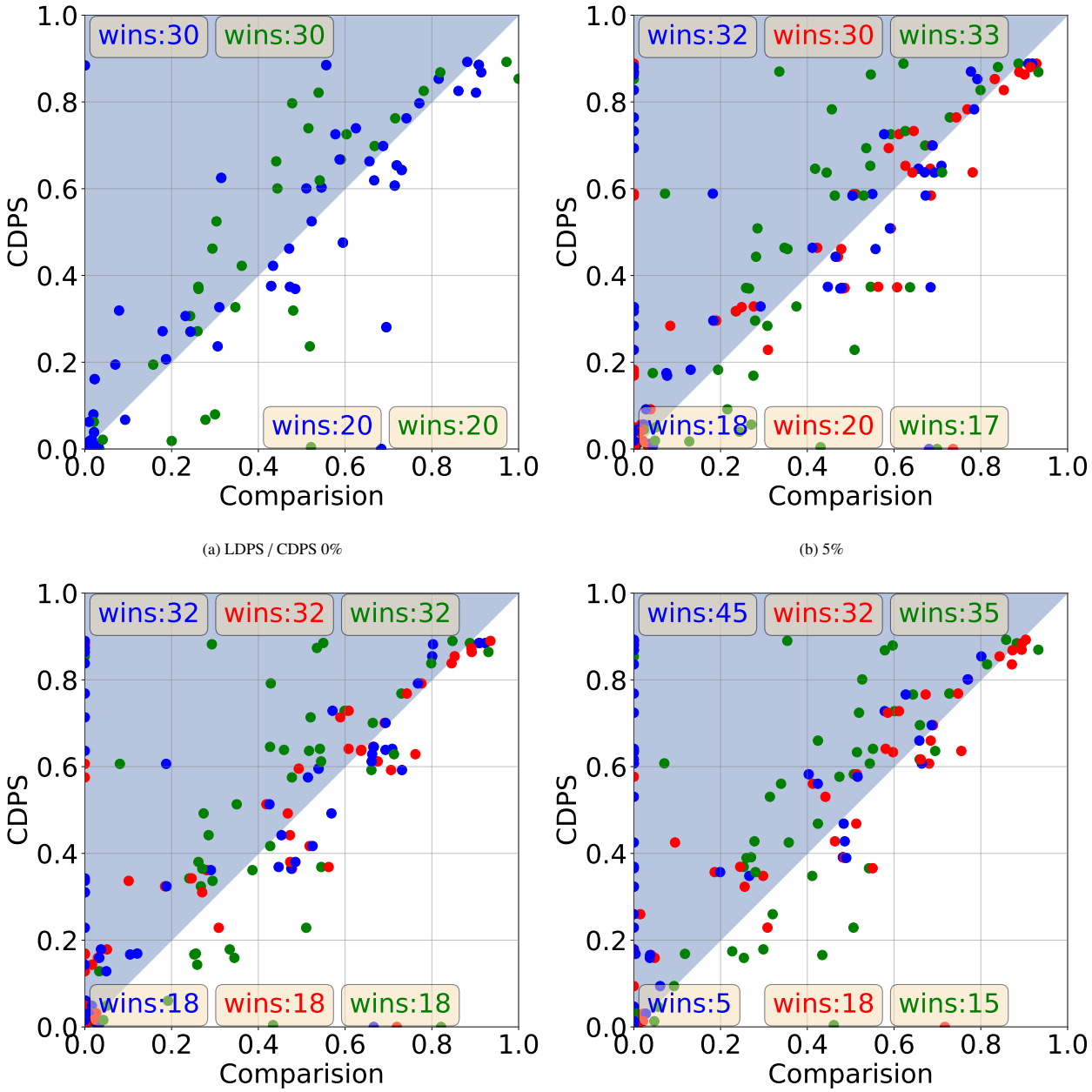
(d) 25%

Figure 2: A Transductive comparison between CDPS+Kmeans (y-axis) and Raw-TS+constrained-algorithms (x-axis) with different constraint fractions. Each point represents a dataset, blue for COP-Kmeans, red for MIP-Kmeans, and green for DCC. Points in the blue triangle are those for which CDPS outperforms the competing algorithms.

gain of 0.61, 0.64, 0.036 respectively, together capturing 1.29 of the total cumulative IG (0.97 of NCIG). Figure 6d presents the order-line of the most informative shapelet (S_{54}), which demonstrates that despite having well separated-clusters , not all categories can be distinguished. It is clear that cluster two (green) and cluster zero (blue) are well separated and the other categories slightly overlap except for cluster one (orange) and four (violet), which overlap by a considerable amount. Shapelet S_{54} represents an increasing trend in the data and therefore has a low distance to clusters two and three, and a high distance to clusters zero and five (having increasing trends). Clusters one and four, however, exhibit both upward and downward trends, hence their overlap in feature space.

Shapelet S_{32} represents a downward trend, therefore adding complementary information to the first shapelet. Adding this shapelet improves separability (see Figure 6e) since clusters zero, two, three, four and five have high density and less overlap, which is not the case for clusters one and six (although they are better separated than only with S_{54}). Shapelet S_{50} represents a cyclic trend, which complements the information previously extracted to better separate clusters one, four and five, as can be seen in Figure 6f (this is better visualised through the projections on the xy, xz, and zy planes).

In light of this, one can ask whether SCE can also help to



(a) LDPS / CDPS 0%

(b) 5%

(c) 15%

(d) 25%

Figure 3: Inductive comparison between CDPS+Kmeans (y-axis) and Raw-TS+constrained-algorithms (x-axis) with different constraint fractions. Each point represents a dataset, blue for COP-Kmeans, red for MIP-Kmeans, and green for DCC. Points in the blue triangle are those for which CDPS outperforms the competing algorithms.

achieve equivalent clustering with fewer, informative shapelets. As such, KMeans clustering is applied to the embeddings obtained using each subset returned by GS-SCE and compared to the results of the original clustering. As expected, NMI follows the same trend as IG, see Figure 7, and it can be observed that using just 20% of the shapelets (10 shapelets) results in approximately the same performance as the original clustering. As such the NCIG can be used to threshold the number of shapelets in a similar way to the eigenvalues in principal component analysis. For example, taking a threshold of 80% NCIG, would result in two shapelets (S_{54} and S_{32}), accumulating 0.94 of the NCIG, while retaining an NMI of 0.8 relative to the original clustering. Adding a third shapelet (S_{50}) increases the NCIG to 0.96 and results in an NMI of 0.9. It should be noted, however, that the reduced dimensional space will not model DTW distance, as will be explored in Section 5.

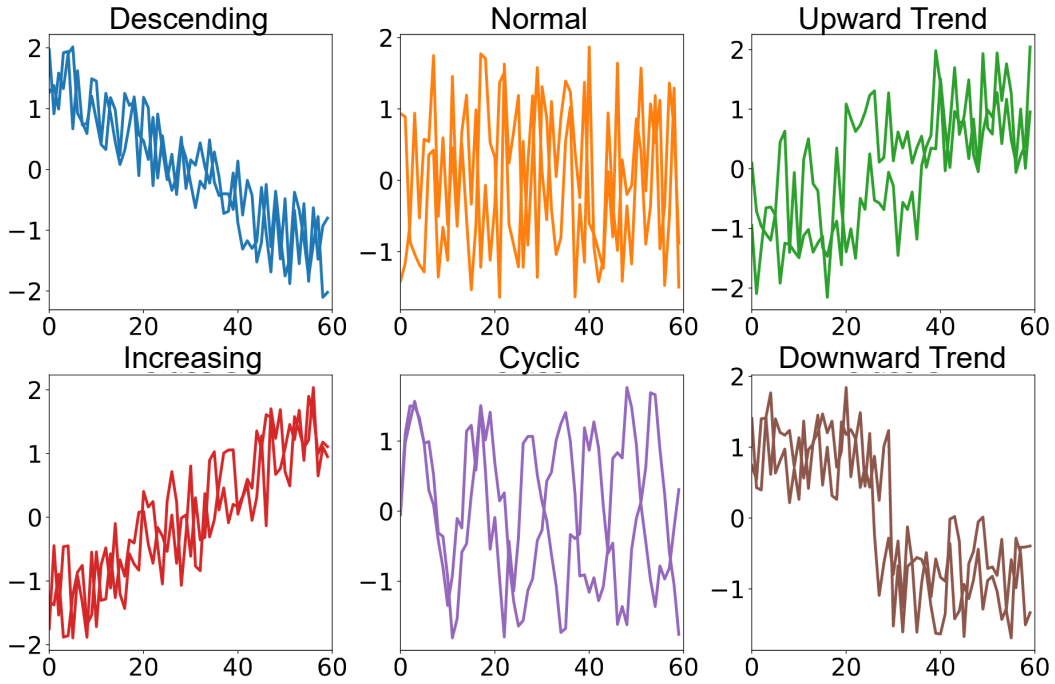


Figure 4: Time-series from each category of the Synthetic Control dataset. Colours correspond to the clustering in Figure 6.

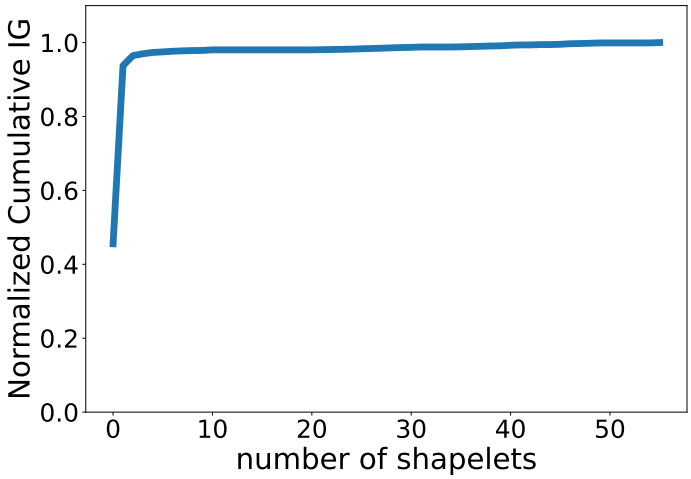


Figure 5: The normalised cumulative information gain calculated using GS-SCE.

5. Discussion

Due to the fact that LDPS only models DTW distance, its induced clustering performs approximately equal to K-means, as shown in Figure 3a) in which the results are clustered around the diagonal. However, both LDPS (and CDPS) result in a metric space, which is beneficial for further analysis.

CDPS is more capable of exploiting the information contained in the constraints when they are introduced, giving more accurate clustering results overall. Furthermore, CDPS avoids the problem of non-convergence that can arise with hard-constrained clustering techniques such as COP-KMeans, which is significantly more challenging when using an elastic measurement such as DTW. In these experiments, all constraints can be considered coherent since they are generated from the ground truth data. However, in real-world situations, this problem would be exacerbated by inconsistent constraints, particularly considering time-series since these are very hard to label. CDPS does not suffer from such limitations. Although it was included in this study for comparison, using constrained clustering algorithms in an inductive use-case is not usual practice. It was simulated by providing the constrained algorithms with the combined 'training' dataset, its constraints, and the test data to be clustered. The necessity to store and access this data can prevent such use-cases and also leads to very high computational cost. CDPS, on the other hand, offers a truly inductive approach in which new data can be projected into the resulting space, which inherently models user constraints. This demonstrates the new possibilities of 'constrained representations'. Moreover, CDPS' embedding space can be used for other tasks such as, classification, generation, etc.

Overall, the CDPS algorithm leads to better clustering results since it is able to exploit the information brought to the learning process by the constraints. In absolute terms, CDPS's performance tends to incease as cosntraints are introduced, while the performance of other algorithms either stagnate or decrease. These constraints bias CDPS to find shapelets that define a representation that respects both DTW and the constraints. Although the focus of this article is not to evaluate whether clustering on these datasets benefits from constraints, it can be observed that generally better performance is achieved when constraints are added.

It is also conceivable to use a constrained clustering algorithm with CDPS' embedding. Although this was not studied, it would allow another mechanism to integrate constraints after

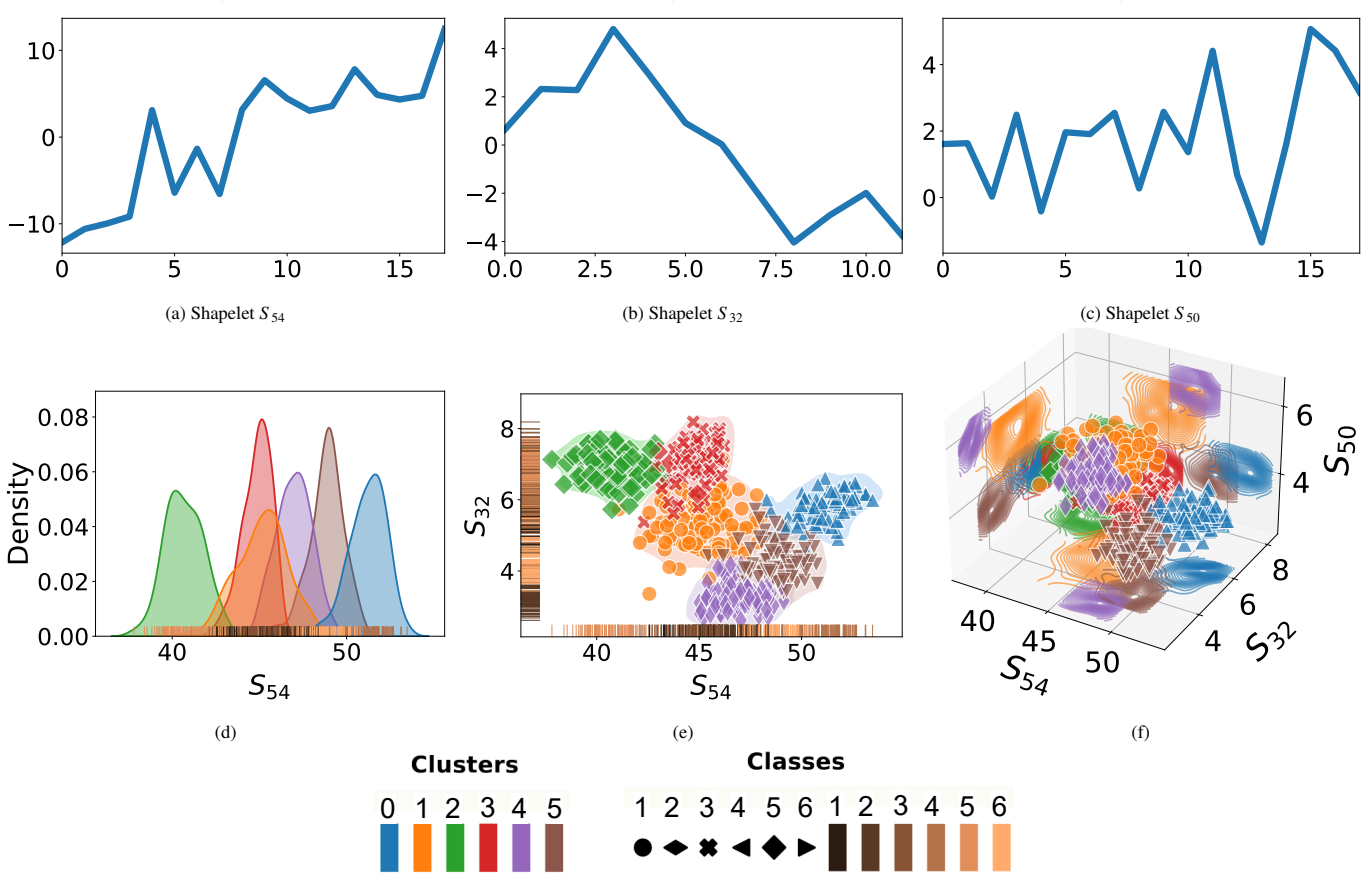


Figure 6: Successive global explanation (GS-SCE). The top three shapelets are presented with the corresponding scatter plots of the data projected into each space, defined by adding each shapelet in succession.

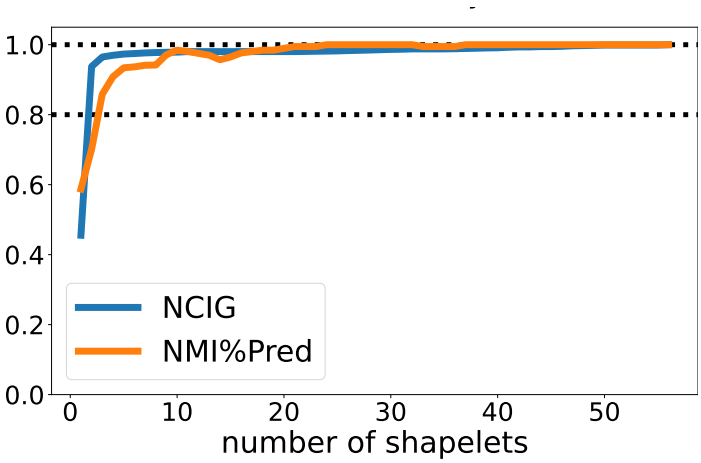


Figure 7: NMI of K-means clustering relative to the number of shapelets used to form the representation (ordered by GS-SCE). Clustering labels are used as class labels for NMI.

the embedding has been learnt.

5.1. Shapelet Selection

Section 4.3 illustrated cluster explanation using GS-SCE and the potential of fewer shapelets yielding equivalent results to all learnt shapelets. Besides providing insights into clustering results, this demonstrated further possibilities for the explanation. We further illustrate this with 10 datasets in transductive clustering. Table 1 shows the clustering score obtained using shapelet subsets that give an 80% cumulative information gain.

Table 1
Clustering scores for different datasets using all shapelets (100% NCIG) and a subset of the shapelets (thresholded to 80% NCIG).

Dataset	#Shapelets (Total)	NCIG	NMI
BME	1 (68)	96%	0.77±0.000
CBF	2 (68)	100%	0.86±0.000
ECG200	1 (64)	96%	0.70±0.000
GunPoint	1 (70)	99%	0.88±0.000
GunPointAgeSpan	2 (70)	98%	0.72±0.000
Herring	1 (89)	86%	0.66±0.000
MoteStrain	1 (62)	88%	0.64±0.001
OSULeaf	12 (86)	80%	0.67±0.006
Plane	2 (70)	99%	0.82±0.003
Symbols	2 (85)	96%	0.82±0.002

For the majority of the datasets, reducing the dimensionality results in more than 0.7 NMI compared to clustering in the overall space, despite only retaining around 2% of the shapelets.

However, the removed shapelets contain necessary information to preserve the original DTW distances.

Figure 8 shows an example of distance maps obtained using DTW similarity (Figure 8a), CDPS space (Figure 8b), and reduced CDPS space (Figure 8c) for the Plane dataset. It can be seen that DTW and CDPS produce similar distance maps, but reduced CDPS is very different as it loses DTW approximation (yet retains cluster separation). Thus, one can either use all shapelets to retain a DTW approximation while respecting constraints or opt for fewer shapelets for speed and simplicity.

5.2. Strengths and Limitations

Unlike other transformation approaches for time series, CDPS integrates the expert to guide it toward their expectation. When a CDPS representation is given as an input to any downstream algorithm, the results achieved will inherently include the expert knowledge as shown in Section 4. In addition to this, the proposed shapelet cluster explanation enables a user to choose the best shapelets in a successive manner, such that new information is added with each shapelet, and obtain comparable clustering results to that of the full shapelet space.

In contrast, CDPS can be limited by DTW's accuracy. For some datasets the performance of DTW can be poor depending on the dataset domain, type, and task, and since CDPS approximates DTW it can reflect this. Depending on the weight and number of constraints provided, this can be compensated for to a certain degree. Also, since CDPS does not take into account missing values it cannot be used with sparse time series and, because the shapelet transform is time-independent, CDPS is not suitable for time-dependent feature analysis. Since CDPS's loss function does not inherently force dissimilarity between the learnt shapelets, they may be similar but this is mitigated by the proposed shapelet ranking approach, shapelet cluster explanation (SCE).

6. Conclusions

This article presented Constrained DTW Preserving Shapelets (CDPS), an approach for learning shapelets constrained by expert knowledge while approximating DTW similarity between time series. It has also presented Shapelet Cluster Explanation (SCE), an approach for explaining anunderstanding the clustering output produced by CDPS. The user-provided constraints are in the form of must-link and cannot-link pairs of samples. In order to guarantee that the constraints have an effect, the mini-batch approach was adopted, in which constrained information is included in each batch. The resulting space removes many drawbacks of the DTW similarity measure for time-series, including lack of interpretability, inability of constraint analysis nor sample density analysis. Instead, the CDPS algorithm generates a general-purpose Euclidean space representation, making it suitable for use with any subsequent machine learning technique. The SCE framework, with IG as it core, was developed for cluster explanation. Three different search method was developed to rank shapelets: independent, combined, and successive. Then, this can be used to find global (overall clustering explanation) and cluster-wise (cluster-vs-all explanation) explanations. By evaluating the proposed method on multiple uni(multi)-variate) public time-series datasets, it was found that using unconstrained K-means on CPDS representations outperforms COP-KMeans, MIP-Kmeans, and Deep constrained clustering. It was also shown that CDPS's representation is generalisable, something that is not possible with classic constrained clustering algorithms and when applied to unseen data, CDPS outperforms both constrained versions of Kmeans. In an demonstration of the SCE framework, top three shapelets were visualised. It was shown that using fewer shapelets (roughly 80% of the cumulative information gain) could still produce results that were comparable those obtained using all the shapelets. CDPS was shown to be stable to the number and length of the shapelets, as well as to its parameters alpha and gamma, where the details of this study is presented in Appendix B.

These developments pave the way for research on constraint proposition and evaluation. Ultimately this will allow the integration of active learning to result in an algorithm where the clustering process and the expert exchange information until the desired result is obtained while minimising effort.

Acknowledgements

This work was supported by the HIATUS and HERELLES (ANR-18-CE23-0025, ANR-20-CE23-0022) ANR projects. We thank the Nvidia Corporation, the *Centre de Calcul de l'Université de Strasbourg*, and HPC resources of IDRIS under the allocation 2021-A0111011872 made by GENCI.

References

[1] R. Sperandio, Recherche de séries temporelles à l'aide de DTW-preserving shapelets, Ph.D. thesis, Université Rennes 1, 2019.

[2] J. Paparrizos, et al., Debunking four long-standing misconceptions of time-series distance measures, in: SIGMOD, 2020, pp. 1887–1905.

[3] E. Keogh, et al., Towards parameter-free data mining, in: SIGKDD, 2004, pp. 206–215.

[4] V. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals, Soviet physics doklady 10 (1966) 707–710.

[5] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, E. Keogh, Indexing multidimensional time-series, The VLDB Journal 15 (2006) 1–20.

[6] H. Sakoe, S. Chiba, Dynamic-programming approach to continuous speech recognition, in: ICA, 1971, pp. 65–69.

[7] H. Sakoe, S. Chiba, Dynamic programming algorithm optimization for spoken word recognition, in: ICA, 1978, pp. 43–49.

[8] T. Lampert, et al., Constrained distance based clustering for time-series: a comparative and experimental study, DMKD 32 (2018) 1663–1707.

[9] L. Ye, E. Keogh, Time series shapelets: a new primitive for data mining, in: SIGKDD, 2009, pp. 947–956.

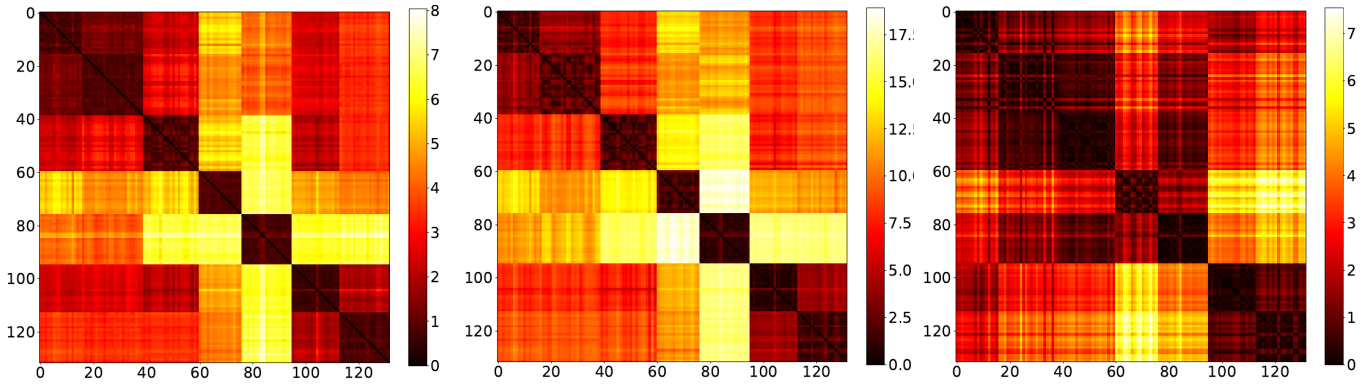
[10] A. Lods, et al., Learning DTW-preserving shapelets, in: IDA, 2017, pp. 198–209.

[11] H. El Amouri, et al., CDPS: Constrained DTW-preserving shapelets, in: ECML PKDD, 2022, pp. 21–37.

[12] G. Vilone, L. Longo, Notions of explainability and evaluation approaches for explainable artificial intelligence, Inf. Fusion 76 (2021) 89–106.

[13] K. Wagstaff, et al., Constrained k-means clustering with background knowledge, in: ICML, 2001, pp. 577–584.

[14] K. L. Wagstaff, S. Basu, I. Davidson, When is constrained clustering beneficial, and why?, Ionosphere 58 (2006) 62–63.



(a) (b) (c)

Figure 8: Distance map comparison between the actual DTW (a), approximate DTW using all shapelets (b) and a subset of shapelets (c) for the Plane dataset. The distance value is represented by the color-bar , brighter the larger the distance.

[15] I. Davidson, S. Ravi, Identifying and generating easy sets of constraints for clustering, in: AAAI, volume 6, 2006, pp. 336–341.

[16] M. Cuturi, M. Blondel, Soft-dtw: a differentiable loss function for time-series, in: International conference on machine learning, PMLR, 2017, pp. 894–903.

[17] M. Herrmann, G. I. Webb, Early abandoning and pruning for elastic distances including dynamic time warping, DMKD 35 (2021) 2577–2601.

[18] L. Ye, E. Keogh, Time series shapelets: a novel technique that allows accurate, interpretable and fast classification, DMKD 22 (2011) 149–182.

[19] J. Lines, et al., A shapelet transform for time series classification, in: SIGKDD, 2012, pp. 289–297.

[20] J. Hills, et al., Classification of time series by shapelet transformation, DMKD 28 (2014) 851–881.

[21] J. Grabocka, et al., Learning time-series shapelets, in: SIGKDD, 2014, pp. 392–401.

[22] M. Shah, et al., Learning DTW-shapelets for time-series classification, in: Conference on Data Science, 2016, pp. 1–8.

[23] J. Zakaria, et al., Clustering time series using unsupervised-shapelets, in: ICDM, 2012, pp. 785–794.

[24] Q. Zhang, et al., Unsupervised feature learning from time series., in: IJCAI, 2016, pp. 2322–2328.

[25] B. Babaki, MIPKmeans, 2017.

[26] H. Zhang, et al., A framework for deep constrained clustering, DMKD (2021) 593–620.

[27] B. Lafabregue, J. Weber, P. Gançarski, G. Forestier, Deep constrained clustering applied to satellite image time series, in: ECML/PKDD Workshop on Machine Learning for Earth Observation Data (MACLEAN), 2019.

[28] D. Tiano, A. Bonifati, R. Ng, Feature-driven time series clustering, in: EDBT, 2021, pp. 349–354.

[29] J. Paparrizos, L. Gravano, k-shape: Efficient and accurate clustering of time series, in: SIGMOD, 2015, pp. 1855–1870.

[30] A. Pérez-Suárez, et al., A review of conceptual clustering algorithms, Artificial Intelligence Review 52 (2019) 1267–1296.

[31] R. C. Sperandio, et al., Time series retrieval using DTW-preserving shapelets, in: Int. Conf. Comput. Commun. Appl., 2018, pp. 257–270.

[32] D. Geng, et al., Personalized recognition of wake/sleep state based on the combined shapelets and k-means algorithm, Biomed. Signal Process. Control 71 (2022) 103132.

[33] R. Hadsell, S. Chopra, Y. LeCun, Dimensionality reduction by learning an invariant mapping, in: CVPR, volume 2, 2006, pp. 1735–1742.

[34] A. Bagnall, et al., The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances, DMKD 31 (2017) 606–660.

[35] H. A. Dau, et al., The UCR Time Series Classification Archive, 2018.

Hussein El Amouri is a doctoral student at the University of Strasbourg and ICube research laboratory. His research focuses on learning representation for time series data and constrained clustering algorithms. He completed his masters degree at the Grenoble INP university and his engineering degree at the Lebanese University Faculty of Engineering.

Thomas Lampert received a PhD from the University of York, UK, and an Habilitation from the University of Strasbourg, France, where he now holds the Chair of Data Science and Artificial Intelligence. His research focuses on Machine Learning, particularly deep learning, unsupervised approaches, domain adaptation, representation learning, and clustering, with application to remote sensing and medical imaging.

Clément Mallet received the Ph.D in image and signal processing at Telecom ParisTech, France in 2010 and the Habilitation in Geographic Information Sciences at the University Paris Est, France, in 2016. He is director of the LASTIG lab. of IGN/Univ Gustave Eiffel, France. He is interested in geospatial computer visionwith an emphasis in the dynamics of urban and natural territories.

Pierre Ganlarski received a Ph.D. and Habilitation in computer science from the University of Strasbourg, France, in 1989 and 2007, respectively. He is currently a full professor of computer science at the University of Strasbourg. His current research interests include collaborative multistrategical clustering with applications to complex data mining, and remote sensing image and time-series analysis

Appendix A. Model Selection

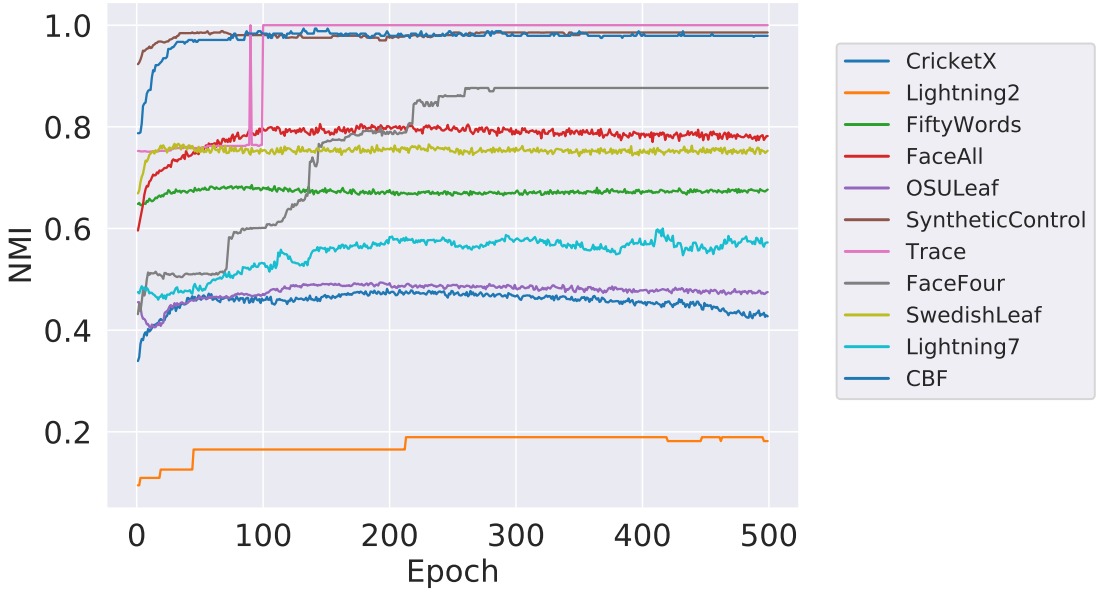


Figure A.9: Clustering quality (NMI) as a function of the number of epochs for each dataset, using a constraint fraction of 30%.

When performing clustering, there is no validation data with which to determine a stopping criteria. It is therefore important to analyse the behaviour of CDPS during training to give some general recommendations. Figure A.9 presents the CDPS clustering quality (NMI) as a function of the number of epochs for each dataset (using 30 % constraints). It demonstrates that generally most of the models converge within a small number of epochs, with FaceFour taking the most epochs to converge. Moreover, the quality of the learnt representation does not deteriorate as the number of epochs increases, i.e. neither the DTW preserving aspect nor the constraint influence dominate the loss and diminish the other as epochs increase.

Figure A.10 presents scatter-plots of the NMI and CDPS loss (both normalised between 0 and 1) for several datasets. In addition to the total loss, both ML and CL losses have been included. The general trend observed is that a lower overall loss equates to a higher NMI.

These show that the loss can be used as a model selection criterion without any additional knowledge of the dataset. For practical application, the embedding can be trained for a fixed, large enough number of epochs (as done in this study) or until stability is

Appendix B. Sensitivity Study

This section will discuss the sensitivity of CDPS with respect to the different parameters, α, γ, and shapelets length and number of shapelets which is dependent on L_{min} and S_{max}.

Appendix B.1. Effect of α and γ

It is common that only a small fraction of the dataset is under constraint, therefore their influence on the learning process is relatively small. The parameters α and γ weight the relevance of these must-link and cannot-link constraints (respectively).

To test the stability of the learning process to different values of these parameters, a grid search was performed over the range 0, 0.5, 1, 1.5, 2, 2.5 (with 10 repetitions of one set of constraints for each pair of values). Both univariate and multivariate datasets from the UCR archive are chosen at random. Figure B.11 presents these results, in which the clustering scores are presented for each pair of values (note that brighter colour equals better performance).

It can be observed in some datasets (Fish, ECG200, BME, for example) that higher parameter values increase performance, in others these parameters do not have a large effect (other datasets) As such, taking $\alpha = \gamma = 2.5$ (as in the previous experiments) is reasonable since they result in high scores over the majority of datasets. This means that the user does not need to optimise their values, nor needs to have in-depth knowledge of their meaning to achieve state-of-the-art performance.

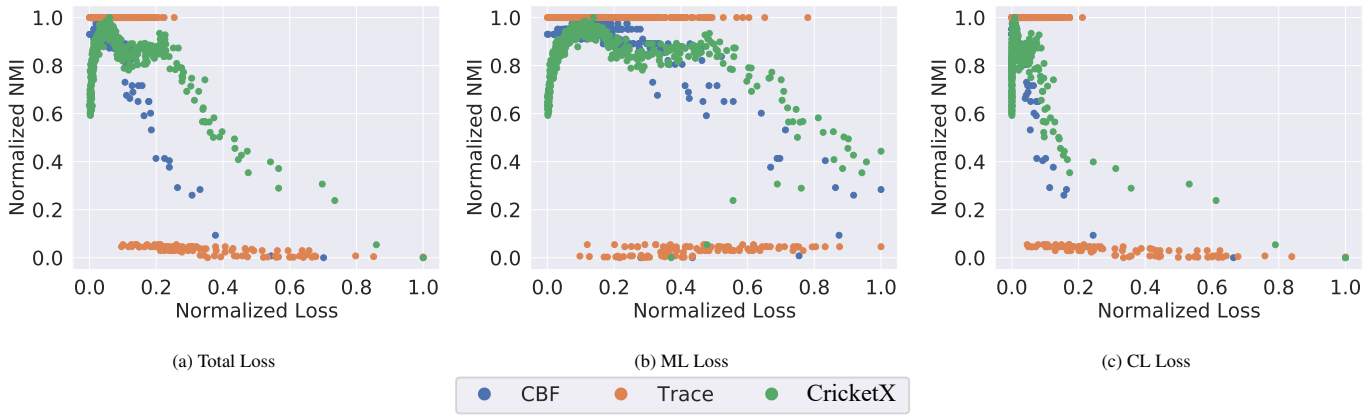


Figure A.10: Relationship between NMI and CDPS Loss for each dataset. To highlight the relationship between datasets, both loss and NMI have been scaled between 0 and 1.

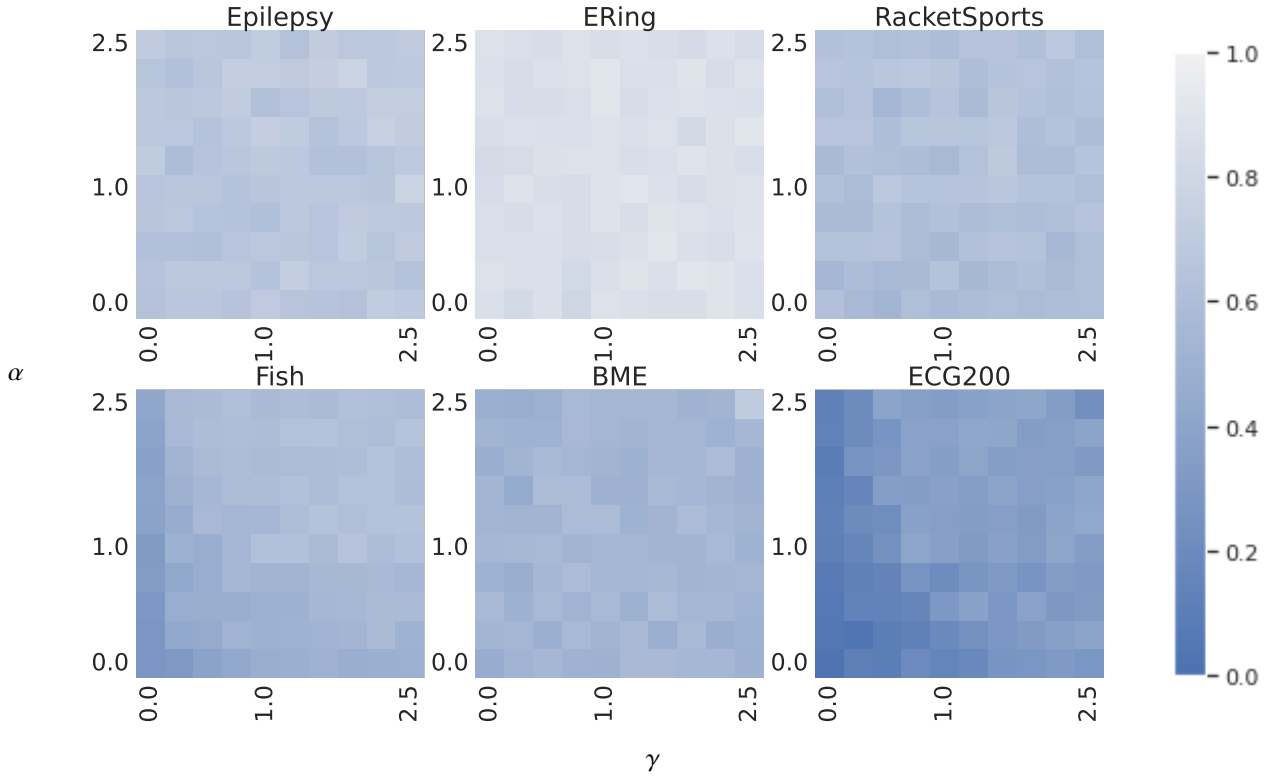


Figure B.11: Heatmaps showing CDPS sensitivity to α and γ parameters, using 6 random univariate datasets (brighter colours indicate better performance).

Appendix B.2. Effect of Shapelet Parameters

The shapelets parameters are: number of shapelet blocks, $S_{\max}$, minimum shapelet length, $L_{\min}$, and a multiplier to specify the number of shapelets per block scale. This study aims to investigate the impact of the shapelet parameters, which are typically user provided, by determining if there is a saturation in the clustering performance induced by the CDPS representation after a specific set of parameter values. As mentioned in Section 3.1, the parameters of the shapelets are specified using the same rule proposed by LDPS [10]. That is, the number of shapelets per block are determined according to the following rule $\log(L_i - L_{\min} \cdot b_{\text{ind}}) \cdot \text{scale}$, where $b_{\text{ind}} \in \{1, \ldots, S_{\max}\}$ indicating the block number. In order to evaluate the sensitivity of the model to the parameters, we explore $S_{\max}$ and scale using the values $\{2, 4, 3\}$ and $\{1, 2, 4, 6, 8, 10\}$ respectively and we fix $L_{\min}$ to $0.15 \cdot L_i$ (since shapelet length is directly related to the block number).

Figure B.12 shows the clustering scores of $S_{\max}$ vs scale. Note that brighter colour means better performance.

It can first be observed that for the majority of datasets the clustering performance is often the same for multiple parameter combinations. This implies that we can choose a smaller multiplier for the shapelets per block as well as a smaller value for the number of shapelet blocks and still guarantee a similar clustering score. For instance, clustering on Epilepsy produced similar

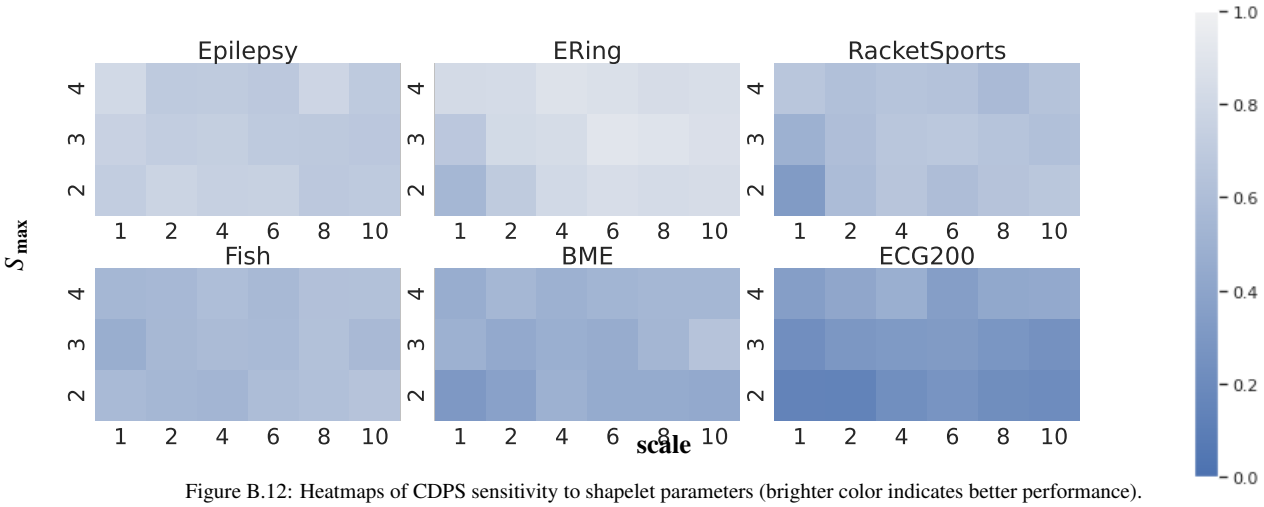


Figure B.12: Heatmaps of CDPS sensitivity to shapelet parameters (brighter color indicates better performance).

results for (4,1) and (4,8), but since (4,1) offers less complexity, it would be preferred. In addition, we can see that for some datasets, the performance degrades as the number of shapelets increase (whether in terms of S_{smax} or scale), as seen in the cases of BME, RacketSports, and EGC200.

Taking this study into the account, we can infer that choosing $S_{\text{max}} = 3$ and scale = 10 generally results in performances on par with the state of the art regardless of the dataset, even though fine tuning these parameters might lead to better results.

Appendix C. Shapelet Cluster Explanation (SCE) Algorithms

The algorithms for the Global Independent (GI-SCE) and Global Combined (GC-SCE) cluster explanation approaches, introduced in Section 3.2, are presented in Algorithm 3 and Algorithm 4 respectively. As noted in the Section 3.2, the Cluster-Wise explanations (CI-SCE, CC-SCE, and CS-SCE), that explain the clustering of a specific cluster rather than the global clustering, can be obtained using the global algorithms but by providing $\updownarrow$ as binary labels that are true for samples contained in the cluster to be explained and false for all others.

Algorithm 3 GI-SCE: Global Independent Shapelet Clustering Explanation

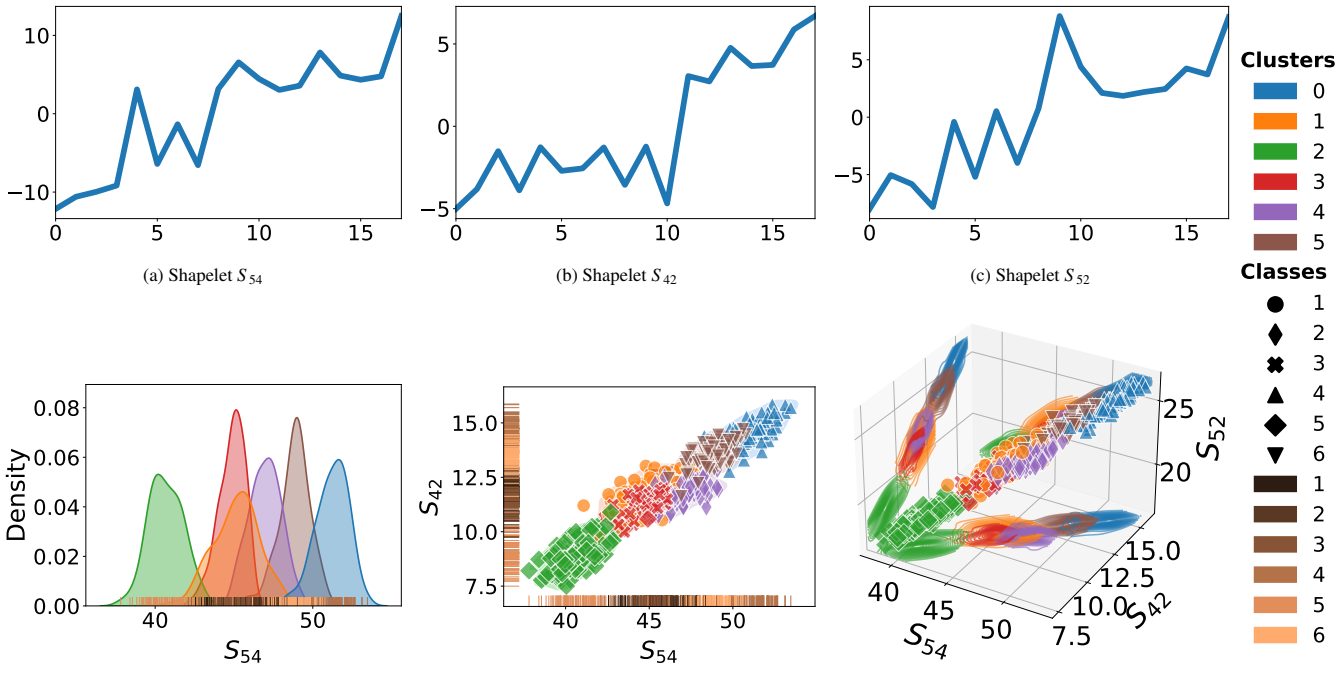
Input: $dists$, a distance matrix between (shapelets, time-series). C, clustering predictions for the time-series.
Output: $\mathcal{S}_d$, Shapelets ordered by information gain. IG, Information gain related to $\mathcal{S}_d$.

$\mathcal{S}_d \leftarrow []$
$IG \leftarrow []$
$S,\ K, N \leftarrow GetShapeletsIndicesAndSize(dists)$
for $j = 1, \ldots, K$ **do**
    $IG^*_j \leftarrow 0$
    for $S_l \in S$ and $S_l \notin \mathcal{S}_d$ **do**
        embvector $\leftarrow$ transpose($dists$)[S_l]
        $\mathcal{D}_j \leftarrow$ get_Partitions_By_Shapelets(embvector, C)
        $IG_j \leftarrow IG(\mathcal{D}_j, S_l)$
        if $IG^*_j < IG_j$ **then**
            $S_j \leftarrow S_l$
            $D_{j-1} \leftarrow D_j$
            $IG^*_j \leftarrow IG_j$
    $\mathcal{S}_d \leftarrow [\mathcal{S}_d, S_j]$
    $IG \leftarrow [IG, IG^*_j]$

Algorithm 4 GC-SCE: Global Combined Shapelet Clustering Explanation

Input: $dists$, a distance matrix between (shapelets, time-series). C, clustering predictions for the time-series.
Output: $\mathcal{S}_d$, Shapelets ordered by information gain. IG, Information gain related to $\mathcal{S}_d$.

$S_j, IG_j^*, \mathcal{D}_{j-1} \leftarrow$ GetBestIndependentShapelet($dists, C$)
$\mathcal{S}_d \leftarrow [S_j]$
$IG \leftarrow [IG_j^*]$
$S, K, N \leftarrow GetShapeletsIndicesAndSize(dists)$
for $j = 2, \ldots, K$ **do**
    $IG_j^* \leftarrow 0$
    for $S_l \in S$ and $S_l \notin \mathcal{S}_d$ **do**
        embvector $\leftarrow$ transpose($dists$)$[\mathcal{S}_d, S_l]$
        $\mathcal{D}_j \leftarrow$ get_Partitions_By_Shapelets(embvector, C)
        $IG_j \leftarrow IG(\mathcal{D}_j, [\mathcal{S}_d, S_l])$
        if $IG_j^* < IG_j$ **then**
            $S_j \leftarrow S_l$
            $D_{j-1} \leftarrow D_j$
            $IG_j^* \leftarrow IG_j$
    $\mathcal{S}_d \leftarrow [\mathcal{S}_d, S_j]$
    $IG \leftarrow [IG, IG_j^*]$



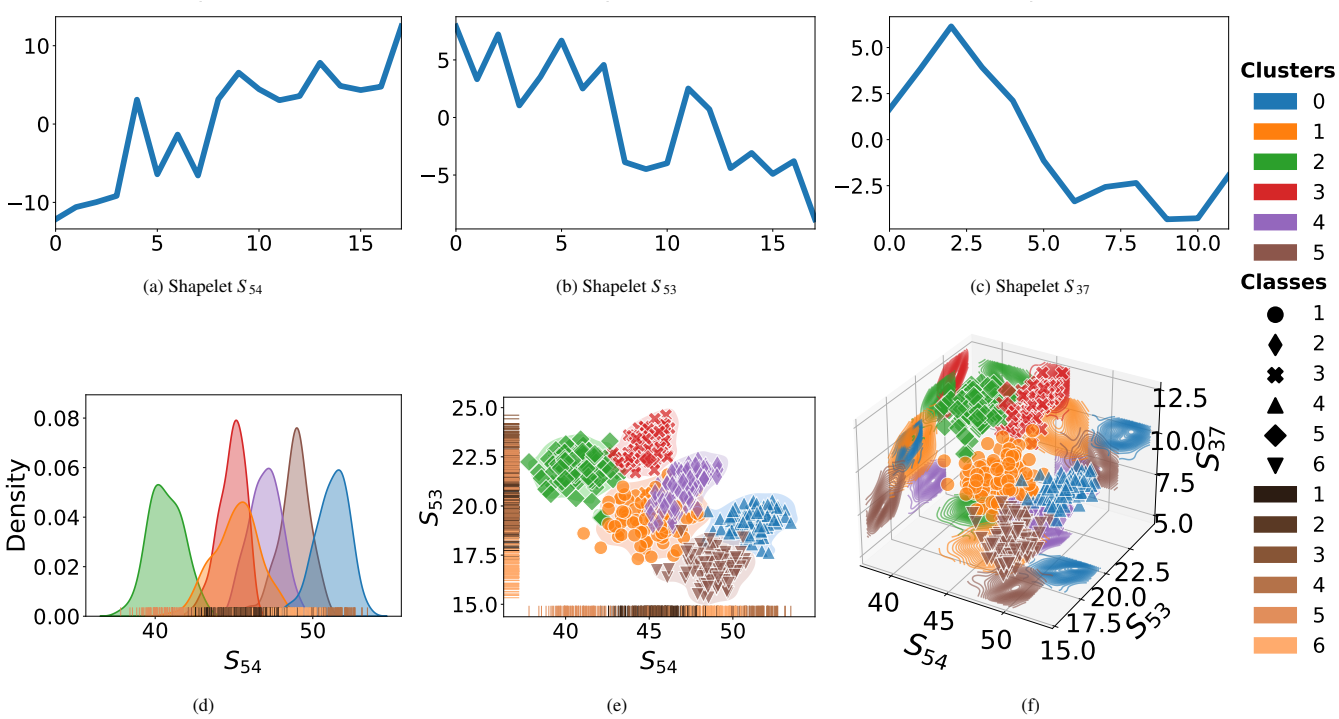
(a) Shapelet S_{54} (b) Shapelet S_{42} (c) Shapelet S_{52}

(d) (e) (f)

Figure D.13: Independent global explanation (GI-SCE). The top three shapelets are presented with the corresponding scatter plots of the data projected into each space, defined by adding each shapelet in succession.

Appendix D. Visualisation of GI-SCE and GC-SCE

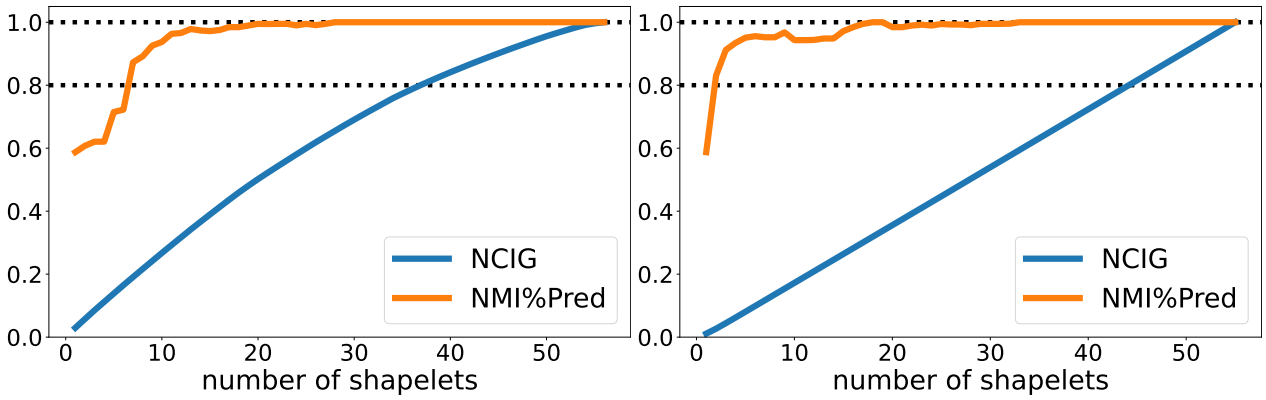
In order to better understand the motivation behind the proposed GS-SCE explanation approaches and to highlight the difference, this section provides visual examples of GI-SCE and GC-SCE. The first three best shapelets are used to infer the difference between the approaches presented in Subsection 3.2. It is clear that in the **Independent** case, a linear space of the points is built (Figure D.13), since there is no requirement for the ordering of the shapelets to reflect complementary information, they are highly correlated. Contrarily, the **Combined** approach tries to find diverse shapelets (Figure D.14), however, the 2nd and 3rd ranked shapelets are still relatively correlated (both extract general downward trends), whereas all three shapelets selected by **Successive** (see Section 3.2) extract complementary information and are diverse.



(a) Shapelet S_{54} (b) Shapelet S_{53} (c) Shapelet S_{37}

(d) (e) (f)

Figure D.14: Combined global explanation (GC-SCE). The top three shapelets are presented with the corresponding scatter plots of the data projected into each space, defined by adding each shapelet in succession.



(a) GI-SCE (b) GC-SCE

Figure D.15: NMI of K-means clustering relative to the number of shapelets used to from the representation ordered by GI-SCE (Figure D.15a) and by GC-SCE (Figure D.15b). The labels used for NMI are the clustering labels output by CDPS.

This is reflected in their Normalised Cumulative Information Gain (NCIG) plots, Figure D.15, in which the Successive approach quickly finds the shapelets that maximise information gain. These figures also include the NMI score of a k-means clustering performed in the subspace relative to the number of shapelets (plotted in orange). It becomes clear that the information gain found by GI-SCE (Figure D.15a) and GC-SCE (Figure D.15b) do not represent NMI, and would therefore result in an overestimation in the required number of shapelets. On the other hand, GS-SCE (Figure 7, Section 4.3) results in a compact representation (fewer shapelets) with a comparable NMI score.

Appendix E. Results

The following are the full NMI clustering scores that are summarised in Figures 2 and 3, Section 4.

Table E.2
Transductive NMI results. KM, CKM, MKM represents KMeans, COP-KMeans, MIP-KMeans respectively.

	0%			5%				15%				25%			
Dataset	LDPS	KM	DCC	CDPS	CKM	MKM	DCC	CDPS	CKM	MKM	DCC	CDPS	CKM	MKM	DCC
Adiac	0.69	0.58	0.57	0.69	0.58	0.61	0.58	0.7	0.58	0.61	0.58	0.7	0.57	0.61	0.58
ArticularyWordRecognition	0.91	0.92	0.89	0.9	0.92	0.92	0.9	0.91	0.91	0.93	0.89	0.91	0.92	0.93	0.89
AtrialFibrillation	0.12	0.1	0.03	0.11	0.08	0.05	0.17	0.09	0.08	0.03	0.13	0.11	0.1	0.05	0.18
BME	0.44	0.52	0.2	0.45	0.55	0.46	0.28	0.68	0.48	0.49	0.35	0.8	0.52	0.48	0.31
BasicMotions	0.77	0.86	0.28	0.79	0.77	0.77	0.25	0.78	0.76	0.73	0.28	0.77	0.76	0.7	0.25
Beef	0.26	0.31	0.3	0.3	0.29	0.31	0.32	0.31	0.29	0.28	0.3	0.32	0.27	0.28	0.3
BirdChicken	0.01	0.02	0.07	0.05	0.02	0.12	0.11	0.23	0.05	0.06	0.08	0.23	0.1	0.09	0.12
CBF	0.75	0.77	0.4	0.8	0.77	0.78	0.44	0.89	0.75	0.78	0.77	0.9	0.75	0.8	0.95
Car	0.18	0.18	0.23	0.28	0.19	0.17	0.27	0.41	0.19	0.19	0.28	0.46	0.18	0.19	0.26
Coffee	0.63	0.7	0	0.71	0.62	0.72	0.55	0.93	0.69	0.75	0.53	0.96	0.69	0.77	0.55
Cricket	0.9	0.92	0.21	0.92	0.9	0.91	0.36	0.92	0.91	0.89	0.43	0.91	0.91	0.88	0.54
CricketX	0.38	0.49	0.23	0.36	0.47	0.48	0.22	0.36	0.48	0.47	0.23	0.39	0.49	0.48	0.23
CricketY	0.42	0.47	0.29	0.42	0.46	0.48	0.27	0.42	0.47	0.49	0.27	0.43	0.47	0.46	0.27
CricketZ	0.38	0.47	0.23	0.36	0.49	0.49	0.22	0.37	0.5	0.5	0.23	0.37	0.49	0.49	0.22
ECG200	0.02	0.01	0.18	0.06	0.03	0.02	0.26	0.27	0.04	0.05	0.43	0.29	0.05	0.1	0.44
Epilepsy	0.63	0.17	0.04	0.64	0.19	0.31	0.07	0.68	0.19	0.38	0.08	0.72	0.17	0.34	0.08
EthanolConcentration	0.0	0	0.0	0.0	0	0.01	0.0	0.0	0	0.0	0.0	0.01	0	0.01	0.0
FaceAll	0.67	0.66	0.35	0.66	0.65	0.68	0.4	0.67	0.65	0.67	0.37	0.7	0.66	0.69	0.4
FaceFour	0.58	0.55	0	0.61	0.5	0.51	0.49	0.6	0.56	0.45	0.51	0.68	0.51	0.56	0.58
FacesUCR	0.68	0.67	0.51	0.66	0.67	0.67	0.46	0.65	0.67	0.69	0.48	0.69	0.65	0.68	0.49
FiftyWords	0.67	0.69	0.63	0.67	0.69	0.69	0.62	0.68	0.68	0.69	0.61	0.67	0.69	0.69	0.61
Fish	0.31	0.43	0.36	0.44	0.42	0.42	0.3	0.54	0.42	0.43	0.29	0.64	0.41	0.42	0.29
Fungi	0.89	0.82	0.89	0.88	0.81	0.82	0.0	0.89	0.81	0.84	0.0	0.9	0.81	0.84	0.0
GunPoint	0.0	0.0	0.0	0.08	0.0	0.0	0.09	0.68	0.01	0.0	0.16	0.81	0.03	0.02	0.37
GunPointAgeSpan	0.0	0.01	0.5	0.0	0.03	0.02	0.36	0.0	0.01	0.0	0.55	0.0	0.03	0.04	0.42
HandMovementDirection	0.02	0.02	0.04	0.02	0.02	0.02	0.04	0.02	0.02	0.02	0.03	0.02	0.02	0.03	0.03
Handwriting	0.44	0.46	0.27	0.41	0.44	0.57	0.28	0.37	0.45	0.56	0.27	0.37	0.44	0.56	0.27
Heartbeat	0.01	0.03	0.0	0.01	0.03	0.0	0.0	0.01	0.02	0.0	0.0	0.01	0.01	0.0	0.0
Herring	0.03	0.01	0.0	0.02	0.01	0.02	0.01	0.01	0.02	0.01	0.01	0.03	0.0	0.03	0.01
Libras	0.63	0.74	0.54	0.62	0.71	0.62	0.53	0.63	0.7	0.62	0.54	0.63	0.7	0.62	0.54
Lightning2	0.11	0.07	0.04	0.11	0.05	0.06	0.04	0.09	0.05	0.06	0.07	0.07	0.06	0.08	0.05
Lightning7	0.54	0.51	0.42	0.56	0.52	0.52	0.44	0.56	0.49	0.51	0.44	0.57	0.5	0.53	0.43
Mallat	0.87	0.88	0.87	0.88	0.87	0.89	0.85	0.88	0.88	0.9	0.83	0.89	0.9	0.91	0.85
Meat	0.76	0.62	0.49	0.67	0.63	0.61	0.54	0.59	0.62	0.64	0.51	0.64	0.61	0.72	0.55
MoteStrain	0.09	0.09	0.49	0.47	0.16	0.03	0.58	0.63	0	0.06	0.68	0.68	0	0.13	0.75
NATOPS	0.65	0.77	0.44	0.64	0.69	0.68	0.45	0.64	0.67	0.64	0.47	0.63	0.69	0.68	0.51
OSULeaf	0.29	0.24	0.2	0.36	0.24	0.24	0.18	0.41	0.24	0.25	0.19	0.43	0.24	0.24	0.19
PenDigits	0.68	0.71	0	0.71	0.68	0.72	0.68	0.73	0.67	0.73	0.82	0.74	0.67	0.73	0
Phoneme	0.32	0.31	0.16	0.29	0	0.31	0.16	0.29	0.31	0.31	0.16	0.29	0.31	0.31	0.15
Plane	0.89	0.91	0.84	0.89	0.88	0.92	0.86	0.9	0.91	0.93	0.85	0.89	0.9	0.93	0.88
PowerCons	0.43	0.08	0.43	0.28	0.08	0.1	0.32	0.43	0.13	0.1	0.42	0.59	0.03	0.13	0.91
RacketSports	0.61	0.62	0.23	0.61	0.6	0.64	0.21	0.62	0.58	0.59	0.23	0.61	0.56	0.61	0.29
Rock	0.26	0.23	0.43	0.25	0.27	0.26	0.0	0.26	0.22	0.24	0.0	0.25	0.2	0.21	0.0
ScreenType	0.02	0.02	0.02	0.02	0.02	0.02	0.03	0.02	0.01	0.02	0.03	0.01	0.02	0.02	0.03
ShapesAll	0.75	0.74	0.69	0.75	0.75	0.74	0.69	0.75	0.74	0.75	0.68	0.76	0	0.74	0.69
StandWalkJump	0.16	0.11	0.05	0.12	0.13	0.12	0.12	0.13	0.14	0.13	0.16	0.15	0.18	0.1	0.14
SwedishLeaf	0.66	0.59	0.6	0.68	0.58	0.59	0.55	0.71	0.58	0.59	0.54	0.73	0.58	0.59	0.54
Symbols	0.79	0.86	0.81	0.85	0.89	0.87	0.82	0.85	0.86	0.88	0.85	0.85	0.91	0.87	0.83
SyntheticControl	0.86	0.9	0.58	0.88	0.88	0.87	0.52	0.86	0.9	0.92	0.53	0.88	0.9	0.94	0.57
UWaveGestureLibrary	0.66	0.64	0.6	0.64	0.66	0.79	0.65	0.65	0.67	0.79	0.64	0.66	0.66	0.82	0.63

Table E.3
Inductive NMI results. KM, CKM, MKM represents KMeans, COP-KMeans, MIP-KMeans respectively.

	0%			5%				15%				25%			
Dataset	LDPS	KM	DCC	CDPS	CKM	MKM	DCC	CDPS	CKM	MKM	DCC	CDPS	CKM	MKM	DCC
Adiac	0.73	0.58	0.6	0.73	0.58	0.61	0.59	0.73	0.57	0.61	0.6	0.73	0.58	0.61	0.6
ArticularyWordRecognition	0.89	0.91	0	0.89	0.92	0.93	0.89	0.88	0.92	0.91	0.89	0.88	0	0.89	0.88
AtrialFibrillation	0.16	0.02	0	0.17	0.08	0	0.28	0.17	0.1	0	0.25	0.18	0	0	0.3
BME	0.52	0.52	0.3	0.46	0.56	0.48	0.35	0.42	0.53	0.52	0.43	0.47	0.48	0.51	0.42
BasicMotions	0.88	0.56	0	0.87	0.78	0	0.34	0.88	0.8	0	0.29	0.89	0	0	0.35
Beef	0.33	0.31	0.35	0.33	0.29	0.28	0.37	0.36	0.29	0.28	0.39	0.35	0.27	0.3	0.41
BirdChicken	0.08	0.02	0.3	0.09	0.03	0.04	0.22	0.18	0.04	0.05	0.33	0.16	0.04	0.05	0.25
CBF	0.8	0.77	0.48	0.78	0.78	0.77	0.46	0.79	0.77	0.78	0.43	0.8	0.77	0.77	0.53
Car	0.27	0.18	0.26	0.3	0.18	0.19	0.28	0.32	0.19	0.18	0.27	0.36	0.2	0.19	0.28
Coffee	0.28	0.7	0	0.37	0.68	0.61	0.64	0.59	0.73	0.71	0.66	0.77	0.63	0.67	0.64
Cricket	0.88	0	0	0.89	0.91	0	0.62	0.89	0.91	0	0.55	0.88	0	0	0.6
CricketX	0.37	0.49	0.26	0.37	0.48	0.48	0.27	0.36	0.48	0.48	0.27	0.39	0.48	0.48	0.27
CricketY	0.46	0.47	0.29	0.44	0.47	0.47	0.28	0.44	0.45	0.47	0.29	0.43	0.49	0.46	0.28
CricketZ	0.37	0.47	0.26	0.37	0.48	0.49	0.26	0.38	0.49	0.47	0.26	0.39	0.49	0.49	0.26
ECG200	0.02	0.01	0.2	0.06	0.02	0.02	0.27	0.16	0.03	0.03	0.34	0.17	0.04	0.04	0.43
Epilepsy	0.62	0.31	0	0.59	0.18	0	0.07	0.61	0.19	0	0.08	0.61	0	0	0.07
EthanolConcentration	0.01	0	0	0.01	0.04	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0	0.01	0.01
FaceAll	0.66	0.66	0.44	0.65	0.66	0.68	0.42	0.65	0.67	0.66	0.43	0.66	0.66	0.68	0.42
FaceFour	0.6	0.55	0	0.59	0.55	0.51	0.51	0.6	0.54	0.49	0.54	0.58	0.4	0.51	0.51
FacesUCR	0.62	0.67	0.54	0.58	0.67	0.68	0.53	0.61	0.66	0.68	0.54	0.61	0.66	0.68	0.54
FiftyWords	0.7	0.69	0.67	0.7	0.69	0.69	0.67	0.7	0.69	0.69	0.66	0.7	0.69	0.69	0.66
Fish	0.42	0.43	0.36	0.46	0.41	0.42	0.35	0.51	0.43	0.42	0.35	0.56	0.42	0.41	0.34
Fungi	0.85	0.82	1.0	0.85	0.79	0.83	0.0	0.85	0.8	0.85	0.0	0.85	0.8	0.84	0.0
GunPoint	0.0	0.0	0.03	0.02	0.0	0.0	0.13	0.06	0.0	0.0	0.19	0.17	0.0	0.0	0.12
GunPointAgeSpan	0.0	0.01	0.52	0.0	0.0	0.03	0.43	0.0	0.03	0.0	0.43	0.0	0.0	0.01	0.46
HandMovementDirection	0.04	0.02	0	0.05	0.02	0	0.05	0.05	0.02	0	0.05	0.05	0	0	0.06
Handwriting	0.38	0.43	0	0.37	0.45	0.56	0.55	0.37	0.45	0.56	0.55	0.37	0	0.55	0.54
Heartbeat	0.01	0.03	0	0.01	0.04	0.0	0.0	0.01	0.03	0.0	0.0	0.01	0	0.0	0.0
Herring	0.06	0.01	0.02	0.04	0.01	0.02	0.02	0.03	0.01	0.03	0.02	0.03	0.03	0.02	0.01
Libras	0.65	0.72	0	0.65	0.71	0.63	0.55	0.64	0.71	0.61	0.54	0.64	0	0.58	0.55
Lightning2	0.19	0.07	0.16	0.17	0.08	0	0.04	0.13	0.05	0	0.03	0.09	0.06	0	0.09
Lightning7	0.6	0.51	0.44	0.58	0.5	0	0.46	0.58	0.51	0	0.48	0.58	0.52	0	0.47
Mallat	0.89	0.88	0.97	0.87	0	0.89	0.93	0.86	0	0.89	0.93	0.87	0	0.89	0.93
Meat	0.74	0.62	0.52	0.73	0	0.65	0.63	0.64	0	0.64	0.52	0.62	0	0.66	0.66
MoteStrain	0.07	0.09	0.28	0.04	0	0.01	0.24	0.14	0	0.02	0.26	0.26	0	0.01	0.32
NATOPS	0.64	0.73	0	0.64	0.69	0.64	0.44	0.64	0.69	0.64	0.46	0.63	0	0.6	0.51
OSULeaf	0.27	0.24	0.24	0.32	0	0.24	0.23	0.34	0	0.25	0.24	0.37	0	0.25	0.25
PenDigits	0	0.68	0	0	0.68	0.74	0.7	0	0.67	0.72	0.82	0	0	0.72	0
Phoneme	0.24	0.31	0.52	0.23	0	0.31	0.51	0.23	0	0.31	0.51	0.23	0	0.31	0.51
Plane	0.87	0.91	0.82	0.88	0	0.91	0.84	0.89	0	0.94	0.85	0.89	0	0.9	0.86
PowerCons	0.32	0.08	0.48	0.28	0	0.08	0.31	0.34	0	0.1	0.29	0.43	0	0.1	0.36
RacketSports	0.48	0.59	0	0.51	0.59	0.59	0.29	0.49	0.57	0.47	0.27	0.53	0	0.44	0.31
Rock	0.31	0.23	0.24	0.33	0	0.25	0.0	0.31	0	0.27	0.0	0.32	0	0.26	0.0
ScreenType	0.02	0.02	0.04	0.02	0	0.02	0.05	0.02	0	0.02	0.04	0.01	0	0.02	0.05
ShapesAll	0.76	0.74	0.72	0.76	0	0.74	0.73	0.77	0	0.74	0.73	0.77	0	0.75	0.73
StandWalkJump	0.21	0.19	0	0.18	0.13	0	0.19	0.17	0.12	0	0.26	0.17	0	0	0.23
SwedishLeaf	0.67	0.59	0.59	0.69	0	0.59	0.54	0.71	0	0.59	0.52	0.72	0	0.59	0.52
Symbols	0.83	0.86	0.78	0.83	0	0.85	0.8	0.84	0	0.85	0.8	0.84	0	0.87	0.81
SyntheticControl	0.82	0.9	0.54	0.86	0	0.9	0.55	0.87	0	0.89	0.54	0.87	0	0.87	0.58
UWaveGestureLibrary	0.61	0.71	0	0.64	0.67	0.78	0.71	0.63	0.66	0.76	0.71	0.64	0	0.76	0.69

Appendix F. Derivation of the CDPS Loss Gradient

This section derives the gradient of the loss function shown in Section 3.1. Let

$$Dist_{i,j} = \|\overline{T}_i - \overline{T}_j\|_2, \qquad \mathcal{L}(T_i, T_j) = \frac{1}{2}\psi + \phi_{i,j},$$

where,

$$\psi = \frac{1}{2}\left(DTW(T_i, T_j) - \beta Dist_{i,j}\right)^2,$$

and

$$\phi_{i,j} = \begin{cases} \alpha Dist_{i,j}^2, & \text{if } (i,j) \in ML, \\ \gamma \max(0, w - Dist_{i,j})^2, & \text{if } (i,j) \in CL, \\ 0, & \text{otherwise,} \end{cases}$$

such that w is a predefined constant.

Derivation with Respect to β

$$\frac{\partial \mathcal{L}(T_i, T_j)}{\partial \beta} = \frac{1}{2}\frac{\partial \psi}{\partial \beta} + \frac{\partial \phi}{\partial \beta} = \frac{1}{2}\frac{\partial \psi}{\partial \beta} = -\beta[DTW(T_i, T_j) - \beta Dist_{i,j}].$$

Derivation with Respect to the Shapelets

$$\frac{\partial \mathcal{L}(T_i, T_j)}{\partial S_{k,l}} = \frac{1}{2}\frac{\partial \psi}{\partial S_{k,l}} + \frac{\partial \phi}{\partial S_{k,l}}.$$

The derivations of ψ and ϕ with respect to the shapelets $S_{k,l}$ will be presented separately.

Using the chain rule, the derivation with respect to ψ can be written as such that

$$\frac{\partial \psi}{\partial S_{k,l}} = \frac{\partial \psi}{\partial Dist_{i,j}} \frac{\partial Dist_{i,j}}{\partial \Delta_{i,j,k}} \frac{\partial \Delta_{i,j,k}}{\partial S_{k,l}},$$

where $\Delta_{i,j,k} = \overline{T}_{i,k} - \overline{T}_{j,k}$. The derivation of each term is straight-forward:

$$\frac{\partial \psi}{\partial Dist_{i,j}} = -\beta(DTW(T_i, T_j) - \beta Dist_{i,j}),$$

$$\frac{\partial Dist_{i,j}}{\partial \Delta_{i,j,k}} = \frac{\Delta_{i,j,k}}{Dist_{i,j}}, \qquad \text{where } Dist_{i,j} \neq 0,$$

and

$$\frac{\partial \Delta_{i,j,k}}{\partial S_{k,l}} = \frac{\partial \overline{T}_{i,k}}{\partial S_{k,l}} - \frac{\partial \overline{T}_{j,k}}{\partial S_{k,l}},$$

where

$$\frac{\partial \overline{T}_{i,k}}{\partial S_{k,l}} = \frac{\partial \min(D_{i,k,m})}{\partial S_{k,l}} = \sum_m \frac{\partial \overline{T}_{i,k}}{\partial D_{i,k,m}} \frac{\partial D_{i,k,m}}{\partial S_{k,l}}.$$

Following the approximation used in LDPS [10] which gives $\frac{\partial \overline{T}_{i,k}}{\partial D_{i,k,m}} = \delta_{m,m^*}$ the above can be written as:

$$\frac{\partial \overline{T}_{i,k}}{\partial S_{k,l}} = \sum_m \delta_{m,m^*} \frac{\partial D_{i,k,m}}{\partial S_{k,l}},$$

$$\frac{\partial \phi}{\partial S_{k,l}} = \frac{\partial \phi}{\partial Dist_{i,j}} \frac{\partial Dist_{i,j}}{\partial \Delta_{i,j,k}} \frac{\partial \Delta_{i,j,k}}{\partial S_{k,l}},$$

where

$$\frac{\partial \phi}{\partial Dist_{i,j}} = \begin{cases} 2\alpha Dist_{i,j}, & \text{if } (i,j) \in ML, \\ -2\gamma(w - Dist_{i,j}), & \text{if } (i,j) \in CL, \\ 0, & \text{otherwise,} \end{cases}$$

where w is a predefined constant.

Appendix G. Datasets

Table G.4
List of UCR datasets used in the main study.

Dataset	Train size	Test size	Length	No. of Classes	No. of Dimensions
FaceAll	560	1690	131	14	1
MoteStrain	20	1252	84	2	1
Symbols	25	995	398	6	1
PenDigits	7494	3498	8	10	2
ScreenType	375	375	720	3	1
BasicMotions	40	40	100	4	6
ShapesAll	600	600	512	60	1
BME	30	150	128	3	1
Fungi	18	186	201	18	1
CBF	30	900	128	3	1
RacketSports	151	152	30	4	6
NATOPS	180	180	51	6	24
FiftyWords	450	455	270	50	1
Handwriting	150	850	152	26	3
OSULeaf	200	242	427	6	1
CricketY	390	390	300	12	1
SyntheticControl	300	300	60	6	1
Rock	20	50	2844	4	1
CricketX	390	390	300	12	1
GunPoint	50	150	150	2	1
Fish	175	175	463	7	1
Cricket	108	72	1197	12	6
Libras	180	180	45	15	2
BirdChicken	20	20	512	2	1
FaceFour	24	88	350	4	1
UWaveGestureLibrary	120	320	315	8	3
FacesUCR	200	2050	131	14	1
Phoneme	214	1896	1024	39	1
StandWalkJump	12	15	2500	3	4
Epilepsy	137	138	206	4	3
CricketZ	390	390	300	12	1
Mallat	55	2345	1024	8	1
AtrialFibrillation	15	15	640	3	2
EthanolConcentration	261	263	1751	4	3
PowerCons	180	180	144	2	1
HandMovementDirection	160	74	400	4	10
Lightning7	70	73	319	7	1
Plane	105	105	144	7	1
Lightning2	60	61	637	2	1
Adiac	390	391	176	37	1
Meat	60	60	448	3	1
SwedishLeaf	500	625	128	15	1
Heartbeat	204	205	405	2	61
ECG200	100	100	96	2	1
Car	60	60	577	4	1
Coffee	28	28	286	2	1
Herring	64	64	512	2	1
GunPointAgeSpan	135	316	150	2	1
ArticularyWordRecognition	275	300	144	25	9
Beef	30	30	470	5	1

Appendix H. CDPS Clustering Results on the Full UCR Archive

Table H.5
Transductive NMI results for CDPS on the UCR archive.

	0%		5%		15%		25%	
Dataset	LDPS	DCC	CDPS	DCC	CDPS	DCC	CDPS	DCC
ACSF1	0.52	0.39	0.53	0.37	0.54	0.36	0.56	0.36
Adiac	0.69	0.57	0.69	0.58	0.7	0.58	0.7	0.58
ArrowHead	0.27	0.29	0.29	0.33	0.35	0.36	0.4	0.3
ArticularyWordRecognition	0.91	0.89	0.9	0.9	0.91	0.89	0.91	0.89
AtrialFibrillation	0.12	0.03	0.11	0.17	0.09	0.13	0.11	0.18
BME	0.44	0.2	0.45	0.28	0.68	0.35	0.8	0.31
BasicMotions	0.77	0.28	0.79	0.25	0.78	0.28	0.77	0.25
Beef	0.26	0.3	0.3	0.32	0.31	0.3	0.32	0.3
BeetleFly	0.16	0.05	0.21	0.09	0.29	0.06	0.41	0.11
BirdChicken	0.01	0.07	0.05	0.11	0.23	0.08	0.23	0.12
CBF	0.75	0.4	0.8	0.44	0.89	0.77	0.9	0.95
Car	0.18	0.23	0.28	0.27	0.41	0.28	0.46	0.26
Chinatown	0.4	0.02	0.73	0.47	0.83	0.47	0.84	0.39
ChlorineConcentration	0.0	0.0	0.01	0.0	0.01	0.0	0.01	0.0
CinCECGTorso	0.04	0.28	0.16	0.37	0.47	0.45	0.58	0.26
Coffee	0.63	0	0.71	0.55	0.93	0.53	0.96	0.55
Computers	0.05	0.01	0.04	0.01	0.03	0.02	0.02	0.02
Cricket	0.9	0.21	0.92	0.36	0.92	0.43	0.91	0.54
CricketX	0.38	0.23	0.36	0.22	0.36	0.23	0.39	0.23
CricketY	0.42	0.29	0.42	0.27	0.42	0.27	0.43	0.27
CricketZ	0.38	0.23	0.36	0.22	0.37	0.23	0.37	0.22
Crop	0.5	0.34	0.5	0.56	0.53	0.55	0.54	0.55
DistalPhalanxTW	0.55	0.47	0.52	0.58	0.52	0.59	0.53	0.63
ECG200	0.02	0.18	0.06	0.26	0.27	0.43	0.29	0.44
ECG5000	0.43	0.55	0.62	0.73	0.64	0.77	0.65	0.76
ECGFiveDays	0.02	0.01	0.84	0.76	0.99	1.0	1.0	0.99
EOGHorizontalSignal	0.38	0.36	0.4	0.37	0.45	0.37	0.46	0.37
EOGVerticalSignal	0.32	0.34	0.35	0.33	0.39	0.33	0.41	0.33
Earthquakes	0.04	0.02	0.04	0.01	0.04	0.01	0.04	0.01
ElectricDevices	0.36	0.13	0.38	0.07	0.42	0.07	0.44	0.07
Epilepsy	0.63	0.04	0.64	0.07	0.68	0.08	0.72	0.08
EthanolConcentration	0.0	0.0	0.0	0.0	0.0	0.0	0.01	0.0
EthanolLevel	0.01	0.01	0.01	0.01	0.03	0.01	0.06	0.01
FaceAll	0.67	0.35	0.66	0.4	0.67	0.37	0.7	0.4
FaceFour	0.58	0	0.61	0.49	0.6	0.51	0.68	0.58
FacesUCR	0.68	0.51	0.66	0.46	0.65	0.48	0.69	0.49
FiftyWords	0.67	0.63	0.67	0.62	0.68	0.61	0.67	0.61
FingerMovements	0.0	0.01	0.0	0.01	0.0	0.01	0.0	0.01
Fish	0.31	0.36	0.44	0.3	0.54	0.29	0.64	0.29
FordA	0.0	0.0	0.39	0.01	0.53	0.01	0.56	0.01
FordB	0.05	0.0	0.36	0.0	0.52	0.0	0.56	0.0
FreezerRegularTrain	0.1	0.22	0.89	0.23	0.95	0.22	0.92	0.22
FreezerSmallTrain	0.1	0.23	0.87	0.23	0.92	0.24	0.92	0.22
Fungi	0.89	0.89	0.88	0.0	0.89	0.0	0.9	0.0
GunPoint	0.0	0.0	0.08	0.09	0.68	0.16	0.81	0.37
GunPointAgeSpan	0.0	0.5	0.0	0.36	0.0	0.55	0.0	0.42
GunPointMaleVersusFemale	0.58	0.08	0.8	0.32	0.82	0.84	0.86	0.84
GunPointOldVersusYoung	0.02	0.07	0.1	0.19	0.55	0.57	0.46	0.62
Ham	0.03	0.1	0.04	0.14	0.09	0.23	0.14	0.22
HandMovementDirection	0.02	0.04	0.02	0.04	0.02	0.03	0.02	0.03
Handwriting	0.44	0.27	0.41	0.28	0.37	0.27	0.37	0.27
Haptics	0.1	0.09	0.09	0.1	0.11	0.11	0.12	0.1
Heartbeat	0.01	0.0	0.01	0.0	0.01	0.0	0.01	0.0
Herring	0.03	0.0	0.02	0.01	0.01	0.01	0.03	0.01

(continued on next page)

Table H.5 *(continued)*

	0%		5%		15%		25%	
Dataset	LDPS	DCC	CDPS	DCC	CDPS	DCC	CDPS	DCC
HouseTwenty	0.53	0.13	0.57	0.13	0.66	0.17	0.67	0.22
InlineSkate	0.11	0.05	0.08	0.06	0.1	0.05	0.13	0.05
InsectEPGRegularTrain	0.36	0.13	0.43	0.24	0.53	0.3	0.62	0.3
InsectEPGSmallTrain	0.36	0.18	0.44	0.25	0.53	0.27	0.62	0.34
InsectWingbeatSound	0.22	0.55	0.38	0.53	0.45	0.54	0.49	0.54
ItalyPowerDemand	0.01	0.01	0.64	0.81	0.77	0.67	0.79	0.08
LargeKitchenAppliances	0.13	0.02	0.13	0.03	0.17	0.03	0.24	0.03
Libras	0.63	0.54	0.62	0.53	0.63	0.54	0.63	0.54
Lightning2	0.11	0.04	0.11	0.04	0.09	0.07	0.07	0.05
Lightning7	0.54	0.42	0.56	0.44	0.56	0.44	0.57	0.43
Mallat	0.87	0.87	0.88	0.85	0.88	0.83	0.89	0.85
Meat	0.76	0.49	0.67	0.54	0.59	0.51	0.64	0.55
MedicalImages	0.33	0.26	0.27	0.23	0.31	0.25	0.33	0.24
MiddlePhalanxOutlineAgeGroup	0.4	0.39	0.39	0.42	0.38	0.41	0.39	0.43
MiddlePhalanxOutlineCorrect	0.0	0.02	0.0	0.03	0.0	0.02	0.0	0.02
MiddlePhalanxTW	0.41	0.45	0.41	0.46	0.42	0.5	0.42	0.49
MixedShapesRegularTrain	0.56	0.34	0.63	0.49	0.68	0.57	0.7	0.5
MixedShapesSmallTrain	0.55	0.47	0.62	0.52	0.68	0.58	0.69	0.52
MoteStrain	0.09	0.49	0.47	0.58	0.63	0.68	0.68	0.75
NATOPS	0.65	0.44	0.64	0.45	0.64	0.47	0.63	0.51
NonInvasiveFetalECGThorax1	0.7	0.7	0.74	0.71	0.81	0.71	0.82	0.71
OSULeaf	0.29	0.2	0.36	0.18	0.41	0.19	0.43	0.19
OliveOil	0.53	0.49	0.53	0.37	0.58	0.5	0.57	0.49
PenDigits	0.68	0	0.71	0.68	0.73	0.82	0.74	0
PhalangesOutlinesCorrect	0.0	0.02	0.01	0.03	0.01	0.01	0.0	0.01
Phoneme	0.32	0.16	0.29	0.16	0.29	0.16	0.29	0.15
PigAirwayPressure	0.63	0.55	0.62	0.54	0.63	0.54	0.63	0.54
PigArtPressure	0.84	0.63	0.84	0.63	0.83	0.62	0.84	0.62
PigCVP	0.65	0.55	0.67	0.56	0.68	0.56	0.69	0.56
Plane	0.89	0.84	0.89	0.86	0.9	0.85	0.89	0.88
PowerCons	0.43	0.43	0.28	0.32	0.43	0.42	0.59	0.91
ProximalPhalanxOutlineAgeGroup	0.53	0.44	0.49	0.51	0.49	0.52	0.51	0.5
ProximalPhalanxOutlineCorrect	0.08	0.07	0.08	0.1	0.08	0.12	0.08	0.11
ProximalPhalanxTW	0.58	0.57	0.55	0.6	0.55	0.6	0.56	0.6
RacketSports	0.61	0.23	0.61	0.21	0.62	0.23	0.61	0.29
RefrigerationDevices	0.05	0.02	0.05	0.02	0.06	0.02	0.07	0.02
Rock	0.26	0.43	0.25	0.0	0.26	0.0	0.25	0.0
ScreenType	0.02	0.02	0.02	0.03	0.02	0.03	0.01	0.03
SemgHandGenderCh2	0.08	0.13	0.08	0.12	0.1	0.13	0.13	0.14
SemgHandMovementCh2	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.16
ShapeletSim	0.02	0.01	0.04	0.03	0.05	0.02	0.06	0.03
ShapesAll	0.75	0.69	0.75	0.69	0.75	0.68	0.76	0.69
SmallKitchenAppliances	0.24	0.01	0.24	0.01	0.25	0.01	0.26	0.01
SmoothSubspace	0.59	0.47	0.49	0.43	0.67	0.43	0.74	0.5
SonyAIBORobotSurface1	0.69	0.73	0.77	0.83	0.9	0.96	0.91	0.95
SonyAIBORobotSurface2	0.31	0.24	0.63	0.57	0.81	0.79	0.88	0.9
StandWalkJump	0.16	0.05	0.12	0.12	0.13	0.16	0.15	0.14
Strawberry	0.12	0.14	0.49	0.16	0.64	0.17	0.63	0.15
SwedishLeaf	0.66	0.6	0.68	0.55	0.71	0.54	0.73	0.54
Symbols	0.79	0.81	0.85	0.82	0.85	0.85	0.85	0.83
SyntheticControl	0.86	0.58	0.88	0.52	0.86	0.53	0.88	0.57
ToeSegmentation1	0.03	0.01	0.18	0.01	0.63	0.02	0.73	0.04
ToeSegmentation2	0.14	0.02	0.33	0.04	0.47	0.05	0.61	0.06
Trace	0.7	0.54	0.79	0.57	0.85	0.61	0.88	0.6
TwoLeadECG	0.06	0.0	0.92	0.67	1.0	0.54	1.0	0.12
TwoPatterns	0.91	0.02	0.69	0.02	0.76	0.03	0.78	0.04
UMD	0.37	0.37	0.32	0.21	0.41	0.21	0.46	0.23

(continued on next page)

Table H.5 *(continued)*

	0%		5%		15%		25%	
Dataset	LDPS	DCC	CDPS	DCC	CDPS	DCC	CDPS	DCC
UWaveGestureLibrary	0.66	0.6	0.64	0.65	0.65	0.64	0.66	0.63
UWaveGestureLibraryAll	0.5	0.72	0.5	0.72	0.66	0.75	0.71	0.71
UWaveGestureLibraryX	0.45	0.48	0.45	0.46	0.49	0.45	0.5	0.44
UWaveGestureLibraryY	0.42	0.44	0.42	0.4	0.44	0.4	0.46	0.38
Wafer	0.0	0.01	0.28	0.84	0.28	0.21	0.32	0.16
Wine	0.01	0.0	0.01	0.01	0.02	0.01	0.02	0.02
WordSynonyms	0.49	0.42	0.45	0.41	0.47	0.41	0.49	0.41
Worms	0.16	0.09	0.1	0.07	0.11	0.07	0.13	0.08
WormsTwoClass	0.03	0.01	0.02	0.02	0.03	0.02	0.04	0.03
Yoga	0.01	0.01	0.0	0.03	0.0	0.0	0.0	0.0

Table H.6
Inductive NMI results for CDPS on the UCR archive.

	0%		5%		15%		25%	
Dataset	LDPS	DCC	CDPS	DCC	CDPS	DCC	CDPS	DCC
ACSF1	0.57	0.31	0.54	0.35	0.55	0.39	0.56	0.35
Adiac	0.73	0.6	0.73	0.59	0.73	0.6	0.73	0.6
ArrowHead	0.25	0.46	0.36	0.48	0.37	0.47	0.42	0.45
ArticularyWordRecognition	0.89	0	0.89	0.89	0.88	0.89	0.88	0.88
AtrialFibrillation	0.16	0	0.17	0.28	0.17	0.25	0.18	0.3
BME	0.52	0.3	0.46	0.35	0.42	0.43	0.47	0.42
BasicMotions	0.88	0	0.87	0.34	0.88	0.29	0.89	0.35
Beef	0.33	0.35	0.33	0.37	0.36	0.39	0.35	0.41
BeetleFly	0.14	0.19	0.17	0.12	0.21	0.1	0.3	0.11
BirdChicken	0.08	0.3	0.09	0.22	0.18	0.33	0.16	0.25
CBF	0.8	0.48	0.78	0.46	0.79	0.43	0.8	0.53
Car	0.27	0.26	0.3	0.28	0.32	0.27	0.36	0.28
Chinatown	0.2	0.01	0.24	0.05	0.34	0.02	0.3	0.07
ChlorineConcentration	0.0	0.01	0.01	0.01	0.01	0.02	0.01	0.02
CinCECGTorso	0.06	0.24	0.15	0.3	0.19	0.33	0.21	0.35
Coffee	0.28	0	0.37	0.64	0.59	0.66	0.77	0.64
Computers	0.05	0.02	0.03	0.02	0.03	0.02	0.02	0.03
Cricket	0.88	0	0.89	0.62	0.89	0.55	0.88	0.6
CricketX	0.37	0.26	0.37	0.27	0.36	0.27	0.39	0.27
CricketY	0.46	0.29	0.44	0.28	0.44	0.29	0.43	0.28
CricketZ	0.37	0.26	0.37	0.26	0.38	0.26	0.39	0.26
Crop	0.49	0.46	0.47	0.56	0.5	0.56	0.52	0.56
DistalPhalanxTW	0.52	0.53	0.52	0.63	0.52	0.65	0.53	0.67
ECG200	0.02	0.2	0.06	0.27	0.16	0.34	0.17	0.43
ECG5000	0.44	0.62	0.55	0.67	0.61	0.76	0.61	0.79
ECGFiveDays	0.04	0.06	0.12	0.19	0.18	0.14	0.39	0.17
EOGHorizontalSignal	0.37	0.43	0.47	0.44	0.48	0.45	0.5	0.44
EOGVerticalSignal	0.39	0.35	0.38	0.39	0.43	0.39	0.44	0.38
Earthquakes	0.06	0.0	0.03	0.01	0.03	0.01	0.03	0.03
ElectricDevices	0.4	0.04	0.4	0.14	0.44	0.14	0.45	0.13
Epilepsy	0.62	0	0.59	0.07	0.61	0.08	0.61	0.07
EthanolConcentration	0.01	0	0.01	0.01	0.01	0.01	0.01	0.01
EthanolLevel	0.01	0.01	0.02	0.02	0.02	0.02	0.03	0.01
FaceAll	0.66	0.44	0.65	0.42	0.65	0.43	0.66	0.42
FaceFour	0.6	0	0.59	0.51	0.6	0.54	0.58	0.51
FacesUCR	0.62	0.54	0.58	0.53	0.61	0.54	0.61	0.54
FiftyWords	0.7	0.67	0.7	0.67	0.7	0.66	0.7	0.66
Fish	0.42	0.36	0.46	0.35	0.51	0.35	0.56	0.34
FordA	0.01	0.01	0.32	0.01	0.5	0.01	0.55	0.01

(continued on next page)

Table H.6 *(continued)*

	0%		5%		15%		25%	
Dataset	LDPS	DCC	CDPS	DCC	CDPS	DCC	CDPS	DCC
FordB	0.02	0.02	0.26	0.0	0.41	0.01	0.44	0.0
FreezerRegularTrain	0.1	0.27	0.24	0.26	0.63	0.25	0.78	0.25
FreezerSmallTrain	0.1	0.33	0.22	0.35	0.35	0.37	0.38	0.4
Fungi	0.85	1.0	0.85	0.0	0.85	0.0	0.85	0.0
GunPoint	0.0	0.03	0.02	0.13	0.06	0.19	0.17	0.12
GunPointAgeSpan	0.0	0.52	0.0	0.43	0.0	0.43	0.0	0.46
GunPointMaleVersusFemale	0.61	0.15	0.65	0.24	0.82	0.3	0.85	0.73
GunPointOldVersusYoung	0.02	0.07	0.05	0.18	0.12	0.21	0.11	0.34
Ham	0.11	0.05	0.03	0.12	0.07	0.12	0.12	0.21
HandMovementDirection	0.04	0	0.05	0.05	0.05	0.05	0.05	0.06
Handwriting	0.38	0	0.37	0.55	0.37	0.55	0.37	0.54
Haptics	0.09	0.16	0.12	0.14	0.14	0.13	0.14	0.13
Heartbeat	0.01	0	0.01	0.0	0.01	0.0	0.01	0.0
Herring	0.06	0.02	0.04	0.02	0.03	0.02	0.03	0.01
HouseTwenty	0.33	0.26	0.56	0.19	0.66	0.19	0.63	0.25
InlineSkate	0.11	0.23	0.2	0.23	0.19	0.23	0.19	0.21
InsectEPGRegularTrain	0.32	0.28	0.35	0.34	0.42	0.33	0.42	0.37
InsectEPGSmallTrain	0.36	0.18	0.49	0.21	0.49	0.3	0.52	0.24
InsectWingbeatSound	0.24	0.57	0.32	0.58	0.39	0.57	0.42	0.57
ItalyPowerDemand	0.01	0.78	0.06	0.2	0.34	0.21	0.46	0.62
LargeKitchenAppliances	0.14	0.04	0.15	0.06	0.16	0.05	0.26	0.06
Libras	0.65	0	0.65	0.55	0.64	0.54	0.64	0.55
Lightning2	0.19	0.16	0.17	0.04	0.13	0.03	0.09	0.09
Lightning7	0.6	0.44	0.58	0.46	0.58	0.48	0.58	0.47
Mallat	0.89	0.97	0.87	0.93	0.86	0.93	0.87	0.93
Meat	0.74	0.52	0.73	0.63	0.64	0.52	0.62	0.66
MedicalImages	0.34	0.24	0.28	0.26	0.29	0.26	0.31	0.29
MiddlePhalanxOutlineAgeGroup	0.1	0.45	0.43	0.44	0.43	0.46	0.43	0.45
MiddlePhalanxOutlineCorrect	0.07	0.02	0.0	0.03	0.0	0.05	0.0	0.04
MiddlePhalanxTW	0.41	0.47	0.43	0.49	0.43	0.51	0.43	0.53
MixedShapesRegularTrain	0.54	0.42	0.52	0.45	0.61	0.43	0.64	0.47
MixedShapesSmallTrain	0.53	0.5	0.54	0.54	0.58	0.56	0.59	0.55
MoteStrain	0.07	0.28	0.04	0.24	0.14	0.26	0.26	0.32
NATOPS	0.64	0	0.64	0.44	0.64	0.46	0.63	0.51
OSULeaf	0.27	0.24	0.32	0.23	0.34	0.24	0.37	0.25
OliveOil	0.65	0.37	0.57	0.55	0.59	0.56	0.59	0.55
PenDigits	0	0	0	0.7	0	0.82	0	0
PhalangesOutlinesCorrect	0.03	0.05	0.0	0.04	0.0	0.01	0.0	0.01
Phoneme	0.24	0.52	0.23	0.51	0.23	0.51	0.23	0.51
PigAirwayPressure	0.69	0.82	0.83	0.82	0.81	0.82	0.82	0.82
PigArtPressure	0.86	0.83	0.88	0.84	0.87	0.84	0.87	0.84
PigCVP	0.74	0.81	0.83	0.8	0.83	0.8	0.84	0.8
Plane	0.87	0.82	0.88	0.84	0.89	0.85	0.89	0.86
PowerCons	0.32	0.48	0.28	0.31	0.34	0.29	0.43	0.36
ProximalPhalanxOutlineAgeGroup	0.51	0.46	0.51	0.45	0.49	0.47	0.51	0.46
ProximalPhalanxOutlineCorrect	0.14	0.05	0.05	0.08	0.06	0.1	0.07	0.09
ProximalPhalanxTW	0.57	0.5	0.55	0.62	0.57	0.64	0.56	0.66
RacketSports	0.48	0	0.51	0.29	0.49	0.27	0.53	0.31
RefrigerationDevices	0.06	0.04	0.04	0.03	0.06	0.02	0.07	0.03
Rock	0.31	0.24	0.33	0.0	0.31	0.0	0.32	0.0
ScreenType	0.02	0.04	0.02	0.05	0.02	0.04	0.01	0.05
SemgHandGenderCh2	0.07	0.14	0.09	0.14	0.11	0.2	0.14	0.26
SemgHandMovementCh2	0.17	0.2	0.19	0.23	0.19	0.22	0.2	0.22
ShapeletSim	0.65	0.07	0.04	0.06	0.06	0.07	0.08	0.06
ShapesAll	0.76	0.72	0.76	0.73	0.77	0.73	0.77	0.73
SmallKitchenAppliances	0.3	0.01	0.2	0.01	0.2	0.02	0.2	0.01
SmoothSubspace	0.47	0.43	0.5	0.42	0.68	0.45	0.71	0.42

(continued on next page)

Table H.6 *(continued)*

	0%		5%		15%		25%	
Dataset	LDPS	DCC	CDPS	DCC	CDPS	DCC	CDPS	DCC
SonyAIBORobotSurface1	0.35	0.1	0.34	0.23	0.42	0.26	0.46	0.21
SonyAIBORobotSurface2	0.26	0.3	0.36	0.46	0.39	0.47	0.49	0.35
StandWalkJump	0.21	0	0.18	0.19	0.17	0.26	0.17	0.23
Strawberry	0.1	0.14	0.33	0.16	0.62	0.18	0.62	0.17
SwedishLeaf	0.67	0.59	0.69	0.54	0.71	0.52	0.72	0.52
Symbols	0.83	0.78	0.83	0.8	0.84	0.8	0.84	0.81
SyntheticControl	0.82	0.54	0.86	0.55	0.87	0.54	0.87	0.58
ToeSegmentation1	0.09	0.03	0.05	0.02	0.16	0.03	0.32	0.04
ToeSegmentation2	0.44	0.01	0.51	0.01	0.59	0.03	0.66	0.01
Trace	0.74	0.56	0.73	0.59	0.8	0.62	0.83	0.63
TwoLeadECG	0.07	0.09	0.08	0.23	0.16	0.17	0.31	0.14
TwoPatterns	0.88	0.06	0.74	0.03	0.69	0.04	0.71	0.06
UMD	0.32	0.24	0.34	0.28	0.35	0.28	0.34	0.29
UWaveGestureLibrary	0.61	0	0.64	0.71	0.63	0.71	0.64	0.69
UWaveGestureLibraryAll	0.5	0.68	0.41	0.71	0.52	0.72	0.59	0.75
UWaveGestureLibraryX	0.45	0.47	0.43	0.46	0.46	0.46	0.46	0.47
UWaveGestureLibraryY	0.42	0.45	0.4	0.45	0.42	0.44	0.44	0.4
Wafer	0.0	0.01	0.01	0.16	0.04	0.78	0.05	0.32
Wine	0.0	0.09	0.02	0.07	0.02	0.07	0.03	0.07
WordSynonyms	0.5	0.5	0.51	0.52	0.53	0.52	0.54	0.5
Worms	0.18	0.09	0.11	0.09	0.11	0.1	0.12	0.1
WormsTwoClass	0.0	0.01	0.02	0.02	0.02	0.03	0.03	0.03
Yoga	0.0	0.01	0.0	0.01	0.0	0.04	0.01	0.03