Shape-based Representation and Abstraction of Time Series Data along with a Dynamic Time Shape Wrapping as a Dissimilarity Measure

Fatma Ezzahra Gmati
COSMOS
National School of Computer Sciences
University of Manouba
Manouba, Tunisia
fatmaezzahra.gmati@ensi-uma.tn

Salem Chakhar
Portsmouth Business School; and
Centre for Operational Research & Logistics
University of Portsmouth
Portsmouth, UK
salem.chakhar@port.ac.uk

Wided Lejouad Chaari
COSMOS
National School of Computer Sciences
University of Manouba
Manouba, Tunisia
wided.chaari@ensi-uma.tn

Mark Xu
Portsmouth Business School
University of Portsmouth
Portsmouth, UK
mark.xu@port.ac.uk

***Abstract*—This paper proposes a Time Series Shape (TSS) based framework for time series representation and abstraction. The paper also introduces a Dynamic Time Shape Wrapping (DTSW), which is a shape extension of the well-known Dynamic Time Wrapping (DTW) dissimilarity measure. By jointly supporting representation and abstraction, TSS and its related dissimilarity measure DTSW can be applied in hybrid time series data mining tasks, especially those involving both rule induction and classification. The paper also compares the capabilities of TSS and piecewise aggregate approximation (PAA) representation in a classification task. Results show that TSS has the same dimensionality reduction power as PAA. This means that TSS is able to maintain the same classification accuracy as PAA, with an additional time series abstraction capability. The results also indicate that DTSW is able to successfully quantify the comparison between TSS abstractions.**

***keywords*—Dissimilarity Measure, Time Series Abstraction,Time Series Representation, Dynamic Time Shape Wrapping**

I. Introduction

Time series representation maps the original time series into a new representation space, and this, generally, for dimensionality reduction purposes [15]. A representation keeps the time series details, with a relative information loss due to the dimensionality reduction process, therefore similarity and distance measures can be applied on the new representation. Time series abstraction is a description of time series intervals by a set of features [18]. The objective of time series abstraction is to describe the time series trends and their variations [4] [10]. Time series abstraction is also useful to rule induction or time series summarization [25].

This paper introduces a Time Series Shape (TSS) based framework for time series abstraction and representation. TSS represents the time series as a set of overlapping intervals and describes it with a set of features. TSS is also able to model intervals' trends and shapes, which makes it suitable to time series abstraction. Two steps are required to obtain a TSS representation: (1) original time series is first segmented using Piecewise Linear Approximation (PLA) [21]; and then (2) elementary shapes are identified from subsequent lines, such that subsequent shapes share one segment.

The paper also proposes a Dynamic Time Shape-based Wrapping (DTSW), which is a dissimilarity measure applicable over TSS. DTSW follows the same idea as the well-known Dynamic Time Warping (DTW) [5] dissimilarity measure, but the distance between time feature values is replaced by the dissimilarity between shape features. DTSW makes TSS usable as a time series representation. As TSS is both a time series representation and abstraction, it is a good candidate for a plethora of time series data mining tasks such as clustering, rule induction and classification. TSS is especially appropriate to hybrid tasks where rule induction and classification are both required [17].

The TSS has been compared to piecewise aggregate approximation (PAA) [20] representation through a classification task using the k-Nearest Neighbors (KNN) [14] [13] classifier. The KNN classifier has been applied using DTSW (in case of TSS) and DTW (in case of PAA) dissimilarity measures. Results show that TSS has the same dimensionality reduction power as PAA. This means that TSS is able to maintain the same classification accuracy as PAA, with an additional time series abstraction capability. The results also indicate that DTSW is able to successfully quantify the comparison between TSS abstractions.

The rest of the paper is organized as follows. Section II discusses related work. Section III introduces TSS. Section IV details DTSW. Section V compares and evaluates TSS/DTSW. Section VI concludes the paper.

II. Related Work

A. Time Series Representation vs Abstraction

Time series representation and abstraction are two basic operations in time series mining. As stressed by [18], relevant time series analysis approaches proposed in the literature often focus on time series representation and ignore or marginally address time series abstraction issues. In this paper, as in [18], we distinguish between time series abstraction and time series representation. Following [18], we define time series abstraction as the description of the consequent time series intervals through a set of features. In turn, time series representation looks to produce a new representation of the original data in order to be able to support quantitative comparison (often using a (dis)similarity measure) and to enable the application of different mining tasks such as classification, clustering and rule discovery.

B. Time Series Abstraction

The authors in [4] employed time series abstraction in multivariate time series classification, and this through rule discovery. They used the trend abstraction and the value abstraction. The authors in [32] used temporal abstraction in order to identify temporal patterns in the time series. They applied the proposed algorithm in clinical variables and DNA gene expression analysis. The authors in [25] used temporal abstraction in order to generate linguistic description of time series data.

It is important to state that trend variation interval description characterizes the use of the term abstraction in literature. For instance, the author in [16] reviewed feature based time series representations which corresponds to [18]'s definition of abstraction yet he didn't define it as abstraction.

C. Time Series Representation

The authors in [24] provided a categorization of time series representations. The idea in this paper is quantify time series abstractions so that the application of hybrid tasks can be applied. The authors in [17] proposed a conceptual framework that employs rough sets on time series abstractions, in order to deduce temporal patterns. However, in order to appropriately deduce rules, the indiscernibility relation has to compare sequences of time series abstractions.

Several time series representations that reduce the data dimensionality have been proposed in the literature, The authors of [24] distinguish between data adaptive representations like Piecewise Linear Approximation PLA [21] and non data adaptive representation, like PAA [20] and Discrete Fourrier Transform [1]. A special interest will be given in this paper to PAA and PLA since they treat data in a piecewise manner, more details will be given in section V.

D. Similarity Measures

The main well-established state of the art similarity and distance measures for time series are reviewed by [12]. We may distinguish three classical types of similarities in literature [12]: lock-step distances, elastic similarities and pattern based similarities. Lock step methods compare series of equal length like the Euclidian distance and its variants the Lp norm distances [35]. Elastic similarities support phase shift like DTW [5], Move Split and Merge (MSM) [33], Edit Distance with Real penalty (EDR) [7] and Longest Common Sub Sequence (LCSS). Pattern based similarities, which take into account the series shape while computing similarities, like SpAde [8] and AMSS [27]. These similarity measures are applicable to the raw time series data.

The authors in [19] proposed a weighted version of DTW. The WDTW penalizes points with higher phase difference between a reference point and a testing point in order to prevent minimum distance distortion caused by outliers. In [34], the authors proposed a shape based similarity measure by introducing a shape coefficient into the WDTW algorithm.

E. Shape based Representation/Abstraction of Time Series

The authors in [2] defined a Shape Definition Language (SDL) to describe patterns or shapes occurring in historical data. The underlying algorithm compares every two consecutive values in a time series and decides the movement direction in the interval between the values. One limitation of SDL is that linear patterns are most intuitively described visually, typically with textual descriptions involving the use of informal language [31]. The SDL can be considered as both an abstraction and representation, yet it is used for a very specific task.

In the work by [23], the time series is represented by a Piecewise Linear Approximation formalism, and is then described through set of local and global shape features. A probabilistic distance is then defined on the basis of the degree of deformation of features pairs. Thus, two shapes are considered similar if they concur with, or can be easily deformed to, an ideal prototype. The critical component is deformation rules that allow some elasticity in the time or amplitude [31].

In [30], shapes in time series are similarly captured by an arbitrary gradient alphabet for the description of movement directions. However, instead of assessing the direction among consecutive values, the algorithm discovers subseries conforming to the desired trend, which is expressed as a sequence of symbols from the alphabet. To compare subseries, a series' length unit is used, namely, a user supplied value that must be applicable to all series under consideration. The work of [30] focuses only on the search of movement patterns in the time series and does not define a representation formalism or a similarity measure.

We also note the existence of some shape-based approaches to time series clustering, see e.g. [26] [28] [29]

III. Time Series Shape (TSS) Framework

A. Principles of TSS

Two steps are performed to obtain a TSS representation: (i) transform the original time series onto piecewise linear segments; and (i) identify primitive shapes and describe each shape by a set of features. In this paper, the first phase relies on

the bottom-up version of PLA algorithm [21]. PLA has been selected for its (ii) implementation simplicity, (ii) capability to abstract trend and smooth the data automatically, and (iii) ability to exhibit the general shape. Other segmentation techniques, such as [9], can also be applied. The PLA is a time series representation formalism that approximates a time series T of length n by N subsequent segments where $N \ll n$. Uniform segmentation within PLA produces segments of equal length n/N while non-uniform segmentation partitions the time series into segments of unequal length to best fit the shape of the time series [6]. Fig. 1 provides an illustrative example. This figure shows the original time series (top) and the corresponding Piecewise Linear Approximation (bottom). Since N is generally much smaller than n, PLA makes the storage, transmission and computation of the data more efficient [21] [22].

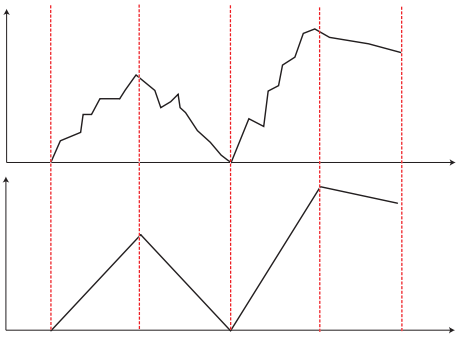

Figure 1. Illustration of Piecewise Linear Approximation.

The obtained segments are the basis for constructing a collection of primitive shapes, which will serve for abstracting and representing the original time series. Each primitive shape is composed of two subsequent segments. Contrary to existing shape based approaches, successive shapes in TSS share one segment. Fig. 2 illustrates the shape construction in TSS. Fig. 2(a) presents a PLA segmentation of the original time series into four segments s_1, s_2, s_3 and s_4. Fig. 2(b) provides the TSS representation where three primitive shapes are constructed using the previous segment. These primitive shapes are constructed by concatenating (i) segments s_1 and s_2, (ii) segments s_2 and s_3, and segments s_3 and s_4, respectively.

As shown in Fig. 2(b), TSS shapes are overlapping since they share one segment. In TSS, any shared segment is represented two times. The use of overlapping segments permits to avoid the bias in the dissimilarity computation and guaranties that all shapes and patterns are identified.

B. TSS Primitive Shapes

A shape in TSS is defined through the angles (θ_1, θ_2) of its composing segments s_1 and s_2 with respect to the horizontal axis (see Fig. 3). Let sh be a TSS shape composed of segments s_1 and s_2. Let also v_1^i and v_2^i be the values of segment s_i $(i = 1, 2)$ endpoints; and t_1^i and t_2^i be the time indexes of segment s_i $(i = 1, 2)$ endpoints. According to this definition

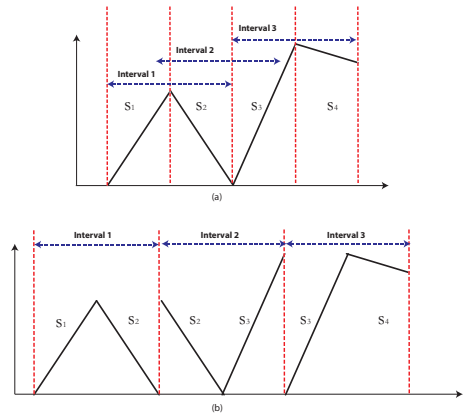

Figure 2. Mapping from PLA segments to TSS shapes.

it is obvious that $t_1^1 < t_2^1 = t_1^2 < t_2^2$. The angle θ_i of segment s_i $(i = 1, 2)$ is defined as follows:

$$\theta_i = \arctan\left(\frac{v_2^i - v_1^i}{t_2^i - t_1^i}\right) \quad (1)$$

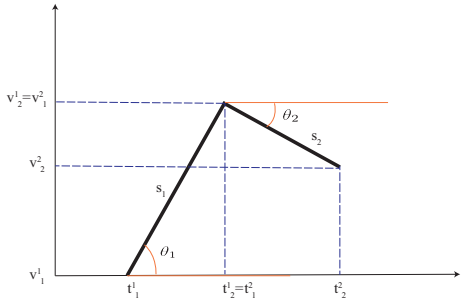

Figure 3. TSS shape definition.

In what follows, we remark that $\theta_i \in]-\frac{\pi}{2}, \frac{\pi}{2}[$, for $i = 1, 2$ and this because there is a unique measure recorded by a timestamp. In TSS, a segment s_i, $i = 1, 2$, can take 5 possible positions, depending on the value of its angle θ_i:

- $p_1 = \{\theta_i : \frac{\pi}{2} > \theta_i \geq \frac{\pi}{4}\}$
- $p_2 = \{\theta_i : \frac{\pi}{4} > \theta_i > 0\}$
- $p_3 = \{\theta_i : \theta_i = 0\}$
- $p_4 = \{\theta_i : 0 > \theta_i > -\frac{\pi}{4}\}$
- $p_5 = \{\theta_i : -\frac{\pi}{4} \geq \theta_i > -\frac{\pi}{2}\}$

Fig. 4 illustrates graphically these possible positions. Each segment of a given TSS shape can take any of these positions. Accordingly, there are 25 TSS primitive shapes, which are given in Fig. 5. The primitive shapes in this figure are grouped into five shape categories, namely Increase (I), Decrease (D), Pick Up (PU), Pick Down (PD) and Stable (S). These categories, except for Stable, are further subdivided into a collection of subcategories, as shown in Fig. 5. We note that for the category Stable with two horizontal segments and with angles $\theta_1 = \theta_2 = 0$, the shape is excluded since PLA algorithm will identify it as a unique segment, not two. In

the rest of this paper, $cat(sh)$ and $sca(sh)$ will denote the category and subcategory of the shape sh, respectively.

C. Characterization of TSS Shapes

A shape in TSS has five parameters. The two first parameters correspond to the symmetry between the amplitude and the duration of the two segments composing the shape. The three last parameters, which apply to the shape as a whole, correspond to shape amplitude, duration and mean.

1) Amplitude Symmetry of Composing Segments: The amplitude symmetry is especially designed to measure the symmetry of the amplitudes of the two segments composing the shape. This parameter is only relevant when the shape category is a PU or a PD. Let s_1 and s_2 be the two segments composing shape sh. The amplitude a_i of segment s_i $(i = 1, 2)$ is computed as follows:

$$a_i = |v_2^i - v_1^i| \tag{2}$$

Then, the amplitude symmetry between the segments composing shape sh is defined as follows:

$$symA(sh) = \begin{cases} \dfrac{a_1 - a_2}{\max(a_1, a_2)}, & \text{if } cat(sh) = \text{PU} \vee \\ & \quad cat(sh) = \text{PD} \quad (3a) \\ \delta, & \text{otherwise} \quad (3b) \end{cases}$$

In this equation, δ is a very small number. We note that, by the definition of PU and PD, $\max(a_1, a_2) \neq 0$. Shapes other than PU and PD are naturally symmetric. The use of δ will ensure that the matching of shape categories other than PU or PD would not be penalized. This definition also ensures that $-1 \leq symA(sh) \leq 1$. A value of $symA(sh) = 0$ means that the segments composing sh are in perfect symmetry with respect to their amplitude. The symmetry between the amplitude of the segments composing sh will decrease with the value of $|symA(sh)|$.

2) Duration Symmetry of Composing Segments: The duration symmetry permits to measure the symmetry of the duration of the two segments composing the shape. This parameter is only relevant when the shape category is a PU or a PD. Let s_1 and s_2 be the two segments composing shape sh. The duration d_i of segment s_i $(i = 1, 2)$ is computed as follows:

$$d_i = t_2^i - t_1^i \tag{4}$$

The duration symmetry between the segments composing shape sh is then defined as follows:

$$symD(sh) = \begin{cases} \dfrac{d_1 - d_2}{\max(d_1, d_2)}, & \text{if } cat(sh) = \text{PU} \vee \\ & \quad cat(sh) = \text{PD} \quad (5a) \\ \delta, & \text{otherwise} \quad (5b) \end{cases}$$

We remark that, by definition (see section III-A), we have $d_i > 0$ $(i = 1, 2)$. This ensures that $\max(d_1, d_2) > 0$. Furthermore, the definition in (5) ensures that $-1 \leq symD(sh) \leq 1$. A value of $symD(sh) = 0$ means that the segments composing sh are in perfect symmetry with respect to their duration. The symmetry between the duration of the segments composing sh will decrease with the value of $|symD(sh)|$.

3) Shape Amplitude: The amplitude of shape sh as a whole is computed as follows:

$$shpA(sh) = \begin{cases} \max(a_1, a_2), & \text{if } cat(sh) = \text{PU} \vee \\ & \quad cat(sh) = \text{PD} \quad (6a) \\ a_1 + a_2, & \text{otherwise} \quad (6b) \end{cases}$$

As shown in this equation, the shape amplitude takes the maximum value of its composing segments' amplitudes when the category of the shape is either PU or PD. In all other cases, the shape amplitude is defined as the sum of its composing segments' amplitudes.

4) Shape Duration: The duration of shape sh is defined as follows:

$$shpD(sh) = t_2^2 - t_1^1 \tag{7}$$

As stated in section III-A, we have $t_2^2 > t_1^1$. This means that $shpD(sh) > 0$.

5) Shape Mean: The mean of shape sh is defined as follows:

$$shpM(sh) = \frac{1}{3}(v_1^1 + v_{mid} + v_2^2) \tag{8}$$

where $v_{mid} = v_2^1 = v_1^2$ is the shared point value of the composing segments of shape sh.

IV. Dynamic Time Shape Wrapping

A. Principles of DTSW

The DTSW can be seen as an extension of DTW to TSS representation of time series. The DTW is considered as a benchmark elastic similarity measure [11]. An elastic similarity relatively supports phase shift in the time series and considers the whole series during its computation [3]. Several elastic similarity measures have been proposed in literature (see e.g. [3]), however none of them outperforms significantly DTW.

DTSW differs from DTW with respect to points: (i) DTSW assumes that the elements of the time series are a sequence of primitive shapes, as defined in section III-B while DTW uses the series raw values as input, and (ii) DTSW uses a similarly measures while DTW relies on a distance function.

Let $T = t_1, \ldots, t_i, \ldots, t_n$ and $S = s_1, \ldots, s_j, \ldots, s_m$ be two time series representated with TSS. Each element of time series T and S is a TSS primitive shape. These time series can be arranged to form a n-by-m matrix M where each cell (i, j) corresponds to an alignement between elements t_i and s_j. A shape warping path W maps the elements of T and S such

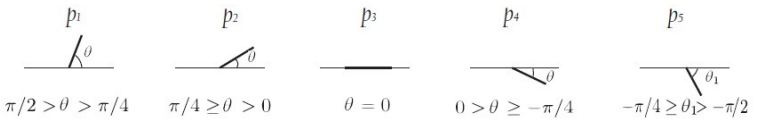

Figure 4. Possible positions of a shape segment.

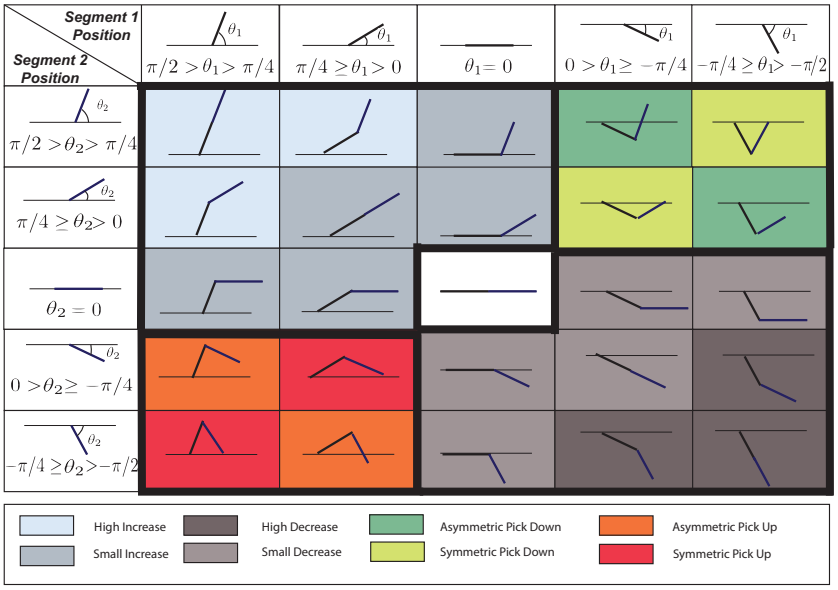

Figure 5. TSS primitive shapes.

that the dissimilarity between them is minimized. Thus, W is defined as sequence $w_1, \ldots, w_k, \ldots, w_p$ of TSS primitive shapes and where each w_k corresponds to a cell $(i, j)_k$ in M.

Similarly to DTW, DTSW uses a dynamic programming approach to identify the best shape warping path. The dynamic programming formulation requires the definition of a dissimilarity measure between two time series shapes. Once a dissimilarity measure is defined, the dynamic time shape warping problem can be defined as a minimization over shape warping paths based on a cumulative dissimilarity measure for each path.

B. Definition of DTSW Dissimilarity Measure

As we assumed a non-uniform segmentation, two shapes with the same angles may have different amplitudes and/or durations for segment and/or shape levels. Alternatively, two shapes with the same amplitude and/or duration with respect to segment and/or shape-levels may have different angles. To ensure that the dissimilarity measure is correctly computed, DTSW compares TSS shapes with respect to three aspects: (i) shape trend; (ii) shape area; and (iii) shape mean. A partial dissimilarity measure is defined for each of these aspects. The obtained partial disssimilarity measures are then aggregated in order to obtain an overall disssimilarity measure.

1) Trends Dissimilarity: The dissimilarity between the trends of two TSS shapes requires the calculation of three matching scores.

a) Matching Score with respect to Angles: Let sh_1 and sh_2 be two TSS shapes defined by the following parameters (θ_1^1, θ_2^1) and (θ_1^2, θ_2^2), respectively. Then, angles matching score $\mu_{ang}(sh_1, sh_2)$ between sh_1 and sh_2 is defined by (9). The four cases in (9) should be mapped to the extremities of three equal subintervals in $[\epsilon, 1]$, i.e. $(\epsilon, \frac{1}{3}, \frac{2}{3}, 1)$. This strategy guarantees that the dissimilarity increases proportionally to the level of mismatch between the considered shapes. The first case in (9) corresponds to a perfect match since the intervals angles fall in the same angle defined position. In this case, a small positive value ϵ is assigned to the matching score $\mu_{ang}(sh_1, sh_2)$ between the shape features of sh_1 and sh_2. The second case holds when the shape features of sh_1 and sh_2 belong to the same subcategory. A value of $\frac{1}{3} > \epsilon$ is then assigned to $\mu_{ang}(sh_1, sh_2)$. The third case holds when the shape features of sh_1 and sh_2 belong to the same category. A value of $\frac{2}{3}$ is then assigned to $\mu_{ang}(sh_1, sh_2)$. If none of the three first cases holds, then a default value of 1 is assigned to $\mu_{ang}(sh_1, sh_2)$.

$$\mu_{ang}(sh_1, sh_2) = \begin{cases} \epsilon, & \text{if } (\theta_1^1 = \theta_1^2) \wedge (\theta_2^1 = \theta_2^2) \quad (9a) \\ \frac{1}{3}, & \text{if } \neg[(\theta_1^1 = \theta_1^2) \wedge (\theta_2^1 = \theta_2^2)] \wedge sca(sh_1) = sca(sh_2) \quad (9b) \\ \frac{2}{3}, & \text{if } \neg[(\theta_1^1 = \theta_1^2) \wedge (\theta_2^1 = \theta_2^2)] \wedge cat(sh_1) = cat(sh_2) \quad (9c) \\ 1, & \text{otherwise} \quad (9d) \end{cases}$$

b) Matching Score with Respect to Segments Amplitude Symmetry: The matching score with respect to the amplitude symmetry of shape segments is calculated as follows:

$$\mu_{symA}(sh_1, sh_2) = [symA(sh_1) - symA(sh_2)]^2 \quad (10)$$

The value of μ_{symA} falls within the range [0,4]. The previous equation captures the distance between segments amplitude symmetry. For instance, for a shape intensely skewed to right with $symA = -1$, and a second shape intensely skewed to the left with $symA = 1$, the difference between the features would be equal to -2. The power of two of this difference is assigned to the amplitude symmetry score.

c) Matching Score with Respect to Segments Duration Symmetry: The matching score with respect to the duration of shape segments is calculated as follows:

$$\mu_{symD}(sh_1, sh_2) = [symD(sh_1) - symD(sh_2)]^2 \quad (11)$$

d) Dissimilarity with Respect to Trends: The shape trend related matching scores can then be aggregated to obtain the dissimilarity between sh_1 and sh_2 in respect to their shapes as in (12).

$$diss_{Tr}(sh_1, sh_2) = (\sqrt{\mu_{symA}(sh_1, sh_2)} + \sqrt{\mu_{symD}(sh_1, sh_2)} \times \mu_{ang}(sh_1, sh_2)) \quad (12)$$

2) Shape Area Dissimilarity: The dissimilarity between areas of two TSS shapes requires the calculation of two matching scores: one for shape amplitude and the second for shape duration. Let sh_1 and sh_2 be two TSS shapes. Then, matching scores with respect to shape amplitude and duration are respectively defined as follows:

$$\mu_{shpA}(sh_1, sh_2) = [shpA(sh_1) - shpA(sh_2)]^2 \quad (13)$$

$$\mu_{shpD}(sh_1, sh_2) = [shpD(sh_1) - shpD(sh_2)]^2 \quad (14)$$

The shape area dissimilarity is then computed as follows:

$$diss_{shpAr} = \sqrt{\mu_{shpA}(sh_1, sh_2)} \times \sqrt{\mu_{shpD}(sh_1, sh_2)} \quad (15)$$

3) Shape Mean Dissimilarity: The shape mean dissimilarity is defined as follows

$$diss_{shpM}(sh_1, sh_2) = \sqrt{[shpM(sh1) - shpM(sh2)]^2} \quad (16)$$

4) Overall Dissimilarity: The overall dissimilarity measure is then given in (17). The computation of the overall dissimilarity takes into account the fact that for different interval shapes, shape mismatch quantification may become irrelevant. Hence, the definition in (17) penalizes the dissimilarity by replacing the shape mismatch score $diss_{Tr}$ with the maximal value it can take. This approach guarantees balance in shape matching.

$$diss(sh_1, sh_2) = \begin{cases} diss_{Tr}(sh_1, sh_2) + diss_{shpAr} + diss_{shpM}(sh_1, sh_2), & \text{if } \mu_{ang}(sh_1, sh_2) < 1 \quad (17a) \\ \max\limits_{sh_1, sh_2} \mu_{ang}(sh_1, sh_2) + diss_{shpAr} + diss_{shpM}(sh_1, sh_2), & \text{otherwise} \quad (17b) \end{cases}$$

C. Identification of Optimal Time Shape Warping Path

The dynamic programming formulation formulation relies on a cumulative dissimilarity $\gamma(i, j)$ for each cell (i, j) in M. The cumulative dissimilarity is defined using the recurrence relation given in (18). Accordingly, $\gamma(i, j)$ is the sum of the dissimilarity between the current elements (i, j) and the minimum of the cumulative dissimilarities of the neighboring points.

$$\gamma(i, j) = diss(i, j) + \min[\gamma(i-1, j), \gamma(i-1, j-1), \gamma(i, j-1)] \quad (18)$$

The dynamic time shape warping problem is then defined as a minimization over shape warping paths which is based on the cumulative dissimilarity measure. Formally,

$$DTSW(T, S) = \min_{W}[\sum_{k=1}^{p} \gamma((i, j)_k)] \quad (19)$$

Similarly to DTW, searching through all possible time shape warping paths is combinatorially expensive. One possible solution is to reduce the search space through considering some restrictions of permissible paths between two cell points [5]. An intuitive restriction consists in imposing that cell points must be monotonically ordered with respect to time, i.e., $i_{k-1} \leq i_k$ and $j_{k-1} \leq j_k$. Another restriction is to constraint allowable cell points to fall within a given warping window. i.e., $|i_k - j_k| \leq \omega$ where ω is a positive integer window width. Some other possible restrictions are enumerated in [5].

It is important to remark that DTSW uses the skeleton of DTW, but performs internal dissimilarities on the TSS intervals features. We used overlapping intervals for TSS definition for the following reason. DTSW considers the intervals as input and each interval is composed by two segments. If intervals are not overlapping, DTSW would dismiss intermediate shapes and this would generate bias in the dissimilarity computation. The proposed definition guarantees the identification of all shapes and patterns by DTSW.

V. Comparison and Evaluation

A. Comparison of TSS to PAA and PLA

Table I compares TSS, PAA and PLA according to various aspects. PAA and PLA, are two state of the art representations that describe the time series in a piecewise manner, such that the description is derivable into an abstraction in various contexts [18], especially when the compression ratio

is high. TSS is based on non-uniform PLA, which makes PLR representations PAA and PLA acceptable as comparison baseline.

Table I
COMPARISON BETWEEN TSS, PAA AND PLA

Comparison aspect	**TSS**	**PAA**	**PLA**
Abstraction capabilities	**High**, It characterizes the time series shapes with a set of predefined shape primitives	**Low**, Maintains the mean of segments as an abstraction, so there is a loss of the series shape	**Average**, characterization of segment endpoints by segments definitions
Compression ratio	Dependant on the series shape	Predefined	For non uniform PLA, it is dependant on the series shape, while predefined for uniform PLA
Representation capabilities: the ability to support a similarity measure	Supports DTSW dissimilarity measure	Supports any type of similarity	Non-uniform PLA does not support similarity (since duration of segments is variable), while uniform PLA supports any type of similarity
Support of rule discovery	**Yes**	**No**, Needs further preprocessing	**No**, Needs further characterization of segments

Furthermore, TSS framework is both a representation and an abstraction. By abstracting the time series, it automatically reduces drastically its dimension, and by memorising some time series shape features it maintains accuracy during the quantification of comparison between abstractions. There is no approach to our knowledge that uses the same strategy. Finally, one should mention that, at least from theoretical point of view, TSS can be applied directly on row data (that is without segmentation). However, this may lead to a high number of shapes.

B. Comparison of TSS to other Shape Based Frameworks

The SDL [2] provides a language that describes the time series such that queries can be matched in a way inspired from regular expressions matching. SDL requires specific parameters in order to define the lower and upper bounds allowed for shape alphabets, which can be inadequate for a task involving heterogeneous datasets, containing different types of time series. The authors in [30] convert the time series into sub-sequences then convert them to the letters of an alphabet representing time series movements, and finally match the converted time series words to a query according to a regular expression in a way that resembles SDL approach. This strategy is computationally expensive, so they adopt some enhancement techniques in order to improve the effectiveness of the process.

In TSS, the definition of shapes is based on PLA, which segments the time series adaptively to its variations. In addition, TSS adopts a different strategy, it opts for a shape to shape comparison, hence can be adapted to different tasks and not exclusively applied to query matching which is the case of [2] and [30]. In [23], the authors use prior probabilistic knowledge in order to compute the shape based similarity measure, yet TSS and DTSW is a based on a purely empirical comparison of the time series.

C. Performance Evaluation

TSS has been evaluated and compared to PAA through a classification task using the k-Nearest Neighbours (KNN) [14] classifier. The KNN classifier has been applied using DTSW (in case of TSS) and DTW (in case of PAA) (dis)similarity measures. A collection of 73 datasets from the UCR archive 2018 [11] have been used. Source code is available under demand to the authors.

We applied the Wilcoxon signed-rank statistic to the obtained results. For this statistics, the p-value must be less than 0.05 in order to reject the null hypothesis. The obtained p-value is equal to $0.34 > 0.05$, therefore the null hypothesis holds. We can deduce that TSS has the same dimensionality reduction power as PAA, yet maintains abstraction expressiveness, and that DTSW ensures an acceptable quantification of comparison between TSS shapes.

VI. CONCLUSION

Our paper proposes a TSS a time series abstraction that has the qualities of a time series representation and can be applied in combined classification and rule induction tasks. The performance of TSS and DTSW, the corresponding dissimilarity measure has been assessed in classification, the results shows that TSS is successful in the considered task.

In this paper, PLA has been selected for its implementation simplicity, but segmentation techniques can be used. In particular, currently, we are investigating the use of the segmentation technique proposed in [9], which is based on genetic algorithms. In the future, we intend to (i) conduct a more advanced comparative study by including other datasets and other techniques; and (ii) evaluate the performance of TSS and DTSW in hybrid tasks that involve rule induction and time series classification.

REFERENCES

[1] R. Agrawal, C. Faloutsos, and A. Swami. Efficient similarity search in sequence databases. In D. B. Lomet, editor, *Foundations of Data Organization and Algorithms*, pages 69–84, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg.

[2] Rakesh Agrawal, Giuseppe Psaila, Edward L. Wimmers, and Mohamed Zaït. Querying shapes of histories. In Umeshwar Dayal, Peter M. D. Gray, and Shojiro Nishio, editors, *Proceedings of 21th International Conference on Very Large Data Bases (VLDB'95), September 11-15, Zurich, Switzerland*, pages 502–514. Morgan Kaufmann, 1995.

[3] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 31(3):606–660, 2017.

[4] I. Batal, L. Sacchi, R. Bellazzi, and M. Hauskrecht. Multivariate time series classification with temporal abstractions. In *Proceedings of the 22nd International Florida Artificial Intelligence Research Society Conference, FLAIRS-22*, pages 344–349, 2009.

[5] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, AAAIWS'94, pages 359–370. AAAI Press, 1994.

[6] V. Bettaiah and H.S. Ranganath. An analysis of time series representation methods: Data mining applications perspective. In *Proceedings of the 2014 ACM Southeast Regional Conference*, ACM SE '14, pages 16:1–16:6, New York, NY, USA, 2014. ACM.

[7] L. Chen and R. Ng. On the marriage of Lp-norms and edit distance. In Mario A. Nascimento, M. Tamer Özsu, Donald Kossmann, Renée J. Miller, José A. Blakeley, and K. Bernhard Schiefer, editors, *Proceedings of the Thirtieth International Conference on Very Large Data Bases (VLDB 2004), August 31-September 3, 2004 Toronto, Canada*, pages 792–803. Morgan Kaufmann, 2004.

[8] Y. Chen, M. A. Nascimento, B. C. Ooi, and A. K. H. Tung. SpADe: On shape-based pattern detection in streaming time series. In *IEEE 23rd International Conference on Data Engineering*, pages 786–795. IEEE, 2007.

[9] Fu-Lai Chung, Tak-Chung Fu, Vincent Ng, and Robert WP Luk. An evolutionary approach to pattern-based time series segmentation. *IEEE transactions on evolutionary computation*, 8(5):471–489, 2004.

[10] G. Das, K. I. Lin, H. Mannila, G. Renganathan, and P. Smyth. Rule discovery from time series. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, KDD'98, pages 16–22. AAAI Press, 1998.

[11] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The ucr time series archive. *CoRR*, 2018.

[12] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh. Querying and mining of time series data: Experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment*, 1(2):1542–1552, 2008.

[13] E. Fix and J.L. Hodges Jr. Discriminatory analysis-nonparametric discrimination: consistency properties. Technical report, University of California, Berkeley, 1951.

[14] E. Fix and J.L. Hodges Jr. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review*, 57(3):238–247, 1989.

[15] T.C. Fu. A review on time series data mining. *Engineering Applications of Artificial Intelligence*, 24(1):164–181, 2011.

[16] B. D. Fulcher. Feature-based time-series analysis. *CoRR*, 2017.

[17] F. Z. Gmati, S. Chakhar, W. Lejouad Chaari, and H. Chen. A rough set approach to events prediction in multiple time series. pages 796–807, 2018.

[18] F. Höppner. Time series abstraction methods — A survey. In S. E. Schubert, B. Reusch, and N. Jesse, editors, *Informatik bewegt: Informatik 2002 - 32. Jahrestagung der Gesellschaft für Informatik e.v. (GI), 30. September - 3.Oktober 2002 in Dortmund, Germany*, LNI, pages 777–786. GI, 2002.

[19] Young-Seon Jeong, Myong K. Jeong, and Olufemi A. Omitaomu. Weighted dynamic time warping for time series classification. *Pattern Recognition*, 44(9):2231–2240, 2011.

[20] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and Information Systems*, 3(3):263–286, 2001.

[21] E. Keogh, S. Chu, D. Hart, and M. Pazzani. An online algorithm for segmenting time series. In *Proceedings 2001 IEEE International Conference on Data Mining*, pages 289–296. IEEE, 2001.

[22] E. Keogh, S. Chu, D. Hart, and M. Pazzani. *Segmenting time series: a survey and novel approach*, pages 1–21. World Scientific, Singapore, 2004.

[23] Eamonn Keogh and Padhraic Smyth. A probabilistic approach to fast pattern matching in time series databases. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, KDD'97, page 24–30. AAAI Press, 1997.

[24] J. Lin, E. Keogh, L. Wei, and S. Lonardi. Experiencing SAX: a novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, 15(2):107–144, 2007.

[25] N. Marín and D. Sánchez. On generating linguistic descriptions of time series. *Fuzzy Sets and Systems*, 285:6–30, 2016.

[26] Warissara Meesrikamolkul, Vit Niennattrakul, and Chotirat Ann Ratanamahatana. Shape-based clustering for time series data. In Pang-Ning Tan, Sanjay Chawla, Chin Kuan Ho, and James Bailey, editors, *Advances in Knowledge Discovery and Data Mining*, pages 530–541, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[27] Tetsuya Nakamura, Keishi Taki, Hiroki Nomiya, Kazuhiro Seki, and Kuniaki Uehara. A shape-based similarity measure for time series data with ensemble learning. *Pattern Analysis and Applications*, 16(4):535–548, 2013.

[28] Vit Niennattrakul, Dararat Srisai, and Chotirat Ann Ratanamahatana. Shape-based template matching for time series data. *Knowledge-Based Systems*, 26:1–8, 2012.

[29] John Paparrizos and Luis Gravano. k-shape: Efficient and accurate clustering of time series. *SIGMOD Record*, 45(1):69–76, 2016.

[30] Yunyao Qu, Changzhou Wang, and X. Sean Wang. Supporting fast search in time series for movement patterns in multiple scales. In *Proceedings of the Seventh International Conference on Information and Knowledge Management*, CIKM'98, page 251–258, New York, NY, USA, 1998. Association for Computing Machinery.

[31] John F. Roddick and Myra Spiliopoulou. A survey of temporal knowledge discovery paradigms and methods. *IEEE Transactions on Knowledge and Data Engineering*, 14(4):750–767, 2002.

[32] L. Sacchi, C. Larizza, C. Combi, and R. Bellazzi. Data mining with temporal abstractions: learning rules from time series. *Data Mining and Knowledge Discovery*, 15(2):217–247, 2007.

[33] A. Stefan, V. Athitsos, and G. Das. The move-split-merge metric for time series. *IEEE Transactions on Knowledge and Data Engineering*, 25(6):1425–1438, 2012.

[34] Y. Ye, C. Niu, J. Jiang, B. Ge, and K. Yang. A shape based similarity measure for time series classification with weighted dynamic time warping algorithm. In *The 4th International Conference on Information Science and Control Engineering (ICISCE)*, pages 104–109, July 2017.

[35] B.-K. Yi and C. Faloutsos. Fast time sequence indexing for arbitrary lp norms. In Amr El Abbadi, Michael L. Brodie, Sharma Chakravarthy, Umeshwar Dayal, Nabil Kamel, Gunter Schlageter, and Kyu-Young Whang, editors, *Proceedings of 26th International Conference on Very Large Data Bases (VLDB 2000), September 10-14, 2000, Cairo, Egypt*, pages 385–394. Morgan Kaufmann, 2000.