Leveraging Subsequence-orders for Univariate and Multivariate Time-series Classification*

Shoumik Roychoudhury †‡ Fang Zhou†§‡ Zoran Obradovic‡

Abstract

Highly discriminative short time-series subsequences, known as shapelets, are used to classify a time-series. The existing shapelet-based methods for time-series classification assume that shapelets are independent of each other. However, they neglect temporal dependencies among pairs of shapelets, which are informative features that exist in many applications. Within this new framework, we explore a scheme to extract informative orders among shapelets by considering the time gap between two shapelets. In addition, we propose a novel model, Pairwise Shapelet-Orders Discovery, which extracts both informative shapelets and shapelet-orders and incorporates the shapelet-transformed space with shapelet-order space for time-series classification. The hypothesis of the study is that the extracted orders could increase the confidence of the prediction and further improves the classification performance. The results of extensive experiments conducted on 75 univariate and 6 multivariate real-world datasets provide evidence that the proposed model could significantly improve accuracy on average over baseline methods.

1 Introduction

Time-series classification has garnered importance [1] in the data-mining community due to the abundance of temporal-ordered data available from a wide range of domains. One group of popular models focuses on identifying short discriminative patterns (subsequences) from the time-series for classification. These short time-series subsequences, known as shapelets [27], are local patterns that can be used to identify the target class for determining the time-series class membership. The main advantage of using shapelets is that they provide local variation information within the time-series as well as high interpretability of predictions due to easy visualizations. Various approaches (such as, [9, 11, 17, 26, 27]) have been proposed to discover the discriminative subsequences from time-series data for classification and have resulted in multiple real-world applications [16, 19, 21, 28].

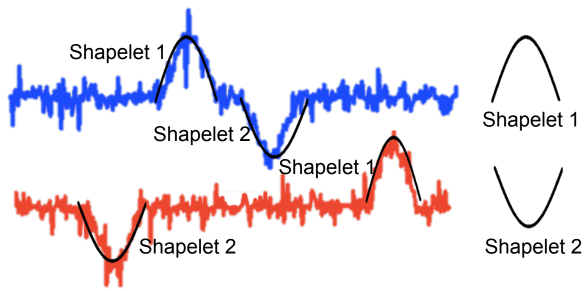


Figure 1: The blue univariate time-series is from class 1, and the red univariate time-series is from class 2. Shapelet 1 and Shapelet 2 could misclassify either classes, as they are present in both classes. However, considering pairwise shapelet-orders allows to differentiate the blue from the red time-series.

Almost all of these shapelet discovery methods are focused on univariate time-series data. Only a handful of methods (e.g., [3, 7, 10]) consider extracting shapelets from multivariate time-series. Both the univariate and multivariate shapelet discovery methods assume that the extracted shapelets are independent of each other, neglecting the role of temporal dependency among pairs of shapelets. For example, in Fig. 1, two instances which are colored differently are from different classes. *Shapelet 1* and *Shapelet 2* are two potential shapelets extracted from the dataset. These two shapelets could not distinguish instances from different classes, as they are present in both instances. However, taking the orders of shapelets into account could classify these instances correctly. A real-life example is in *Intensive Care Units (ICU)* where a patient is connected to multiple health monitoring devices that monitor the patient's health by checking heart rate, blood pressure, etc. Temporal patterns from multiple sensors are often good indicators of the patient's health status. Therefore, the order among shapelets is informative in classification.

In this paper we explore a novel scheme, named *TimeGap-based-orders*, to extract informative orders among pairwise shapelets by considering the time gap between any pairs of shapelets. Based on this scheme, we propose a novel model, Pairwise Shapelet-Orders Discovery (PSOD), which extracts both informative shapelets and shapelet-orders and

*This research was supported in part by the NSF grants IIS-8142183, IIS-1636772, SES-1659998, CNS-1625061 and Pennsylvania Department of Health CURE Health Data Science Research Project.

†Both authors contributed equally to this work.

‡Center for Data Analytics and Biomedical Informatics, Temple University, Philadelphia, USA. Email:{firstname.lastname}@temple.edu

§Corresponding author. School of Data Science and Engineering, East China Normal University, Shanghai, China. Email:fzhou@dase.ecnu.edu.cn

incorporates the shapelet-transformed space with shapelet-order space for time-series classification. Our experiments show that the extracted pairwise shapelet-orders could refine the class membership confidence, which measures the probability of belonging to a particular class of a time-series instance, and improve the classification accuracy.

The proposed model first randomly extracts a subsequence from time-series. If it is significantly different from the already accepted shapelets and rejected shapelets, it is considered as a candidate shapelet. Then the order between the candidate shapelet and any shapelet in the accepted list is evaluated. If the overall classification accuracy is improved, then the candidate shapelet and the order will be saved into the accepted shapelet list and order list respectively. Otherwise, if the candidate shapelet alone improves the classification performance, then the candidate shapelet will be accepted and the order candidate will be discarded. The classification performance of PSOD has been evaluated on both synthetic and real-life datasets.

The *main contributions* of this work are the following:

- This is the *first study that considers temporal dependency information among pairs of shapelets* and generates pairwise shapelet-orders for use in time-series classification.
- *One order-generation scheme* is explored, which emphasizes the time gap between shapelets.
- *A novel model*, PSOD, is proposed to extract informative shapelets and pairwise shapelet-orders together from data. The experimental results provide evidence that when considering shapelet-orders, classification accuracy is improved.

2 Related work

In the field of time-series classification, extracting shapelets to perform classification has recently received extensive attention [12, 22, 26, 29]. The minimum distance between a shapelet and a time-series, namely shapelet transformation [11, 15], is a very popular feature, and can be used as predictors in the traditional classifier framework. Therefore, discovering the most discriminative subsequences is crucial for the success of time-series classification using shapelets.

Search-based techniques [8, 27] conduct an exhaustive search of all possible subsequences, which is often intractable for large datasets. Numerous methods [10, 13, 17, 20, 27] have been proposed to speed up the search process for identifying discriminative shapelets from potential candidates. Alternatively, instead of searching all possible subsequences, generalized shapelets [9, 12, 29] are learned from the data. The above approaches are mainly designed for univariate time-series datasets. A few studies (e.g. [2, 3, 7, 10, 13]) have investigated the shapelet procedure for multivariate time-series datasets.

All the existing shapelet-based approaches only focus on how to select (or generalize) discriminative shapelets, but ignore the orders among shapelets, which is also an important ingredient in prediction. Mueen et al. [17] had proposed *Logical shapelets*, which are logical combinations of multiple shapelets. Using conjunctive and disjunctive logical operations, they increased the expressiveness of the shapelets by discovering logical rules. However, the rules discovered failed to capture the temporal dependency among shapelets. Moreover, *Logical shapelets* was proposed for univariate time-series datasets and the technique of combining multiple shapelets through logical rules from different dimensions was not discussed. Recently, Patri et al. [18] briefly discussed that the temporal dependency among shapelets on multivariate time-series can improve classification performance. The idea is to inter-leave time-series segments from multiple dimensions to form a final concatenated one dimensional time-series. However, this is only applicable to multivariate time-series. In contrast, we propose a formal generalized method to extract the most informative pairwise shapelet-orders that enhance the confidence of prediction on both univariate and multivariate time-series.

Another direction of analyzing time-series has focused on extracting association rules among frequent patterns [4] from time-series. A common approach is to first discretize [5, 14, 25] the time-series data into segments and convert each segment into a symbol. The rules are then discovered in the transformed symbolic domain. The discovery of high quality rules from time-series was also proposed in [23]. Tatavarty et al. [24] considered the problem of discovering temporal dependencies between frequently appearing patterns in multivariate time-series. Their work focused on discovering temporal associations among frequently occurring subsequences from different dimensions by transforming the time-series to a symbolic representation, whereas, we focus on discovering **discriminative** shapelets and the temporal gap among them in both univariate and multivariate time-series data for classification. To the best of our knowledge, this paper is the first work which proposes a formal methodology to extract shapelet-orders and present an augmented space of shapelets and shapelet-orders. In addition, our approach is applicable to both univariate and multivariate time-series datasets.

3 Method Preliminaries

A time-series dataset composed of I training instances is denoted as $\mathbf{T} \in \mathbb{R}^{I \times D \times L}$. We consider instances having d $(1 \leq d \leq D)$ dimensions where each $\mathbf{T}_i$ $(1 \leq i \leq I)$ is of length L (for notation convenience we assume $\mathbf{T}_i$ have equal frequency in all dimensions and L is fixed, however, the length of time-series can vary among training instances) and the corresponding label is a nominal variable $Y_i \in \{1, ..., C\}^I$. When $d = 1$, the data represents a univariate time-series, while $d > 1$ it corresponds to a multivariate (multidimensional) time-series.

Candidate shapelets **S** are short subsequences extracted from time-series, which are discriminative patterns and characterizes the target class. Let $s_d^k \in \mathbf{S}$ represent the k^{th} candidate shapelet of length l extracted from dimension d (l is not mentioned in the notation for simplification). Next, we introduce definitions of some terminologies used.

DEFINITION 1. *Distance between two candidate shapelets* $Dis(s^{k_1}, s^{k_2})$: The distance between two candidate shapelets s^{k_1} and s^{k_2} of same length l is calculated as $Dis(s^{k_1}, s^{k_2}) = \sqrt{\frac{1}{l}\sum_{p=1}^{l}(s_p^{k_1} - s_p^{k_2})^2}$, where s_p^k represents the p^{th} value in the candidate shapelet s_d^k of length l.

DEFINITION 2. *Minimum distance* ($m_{i,k}$): The minimum distance $m_{i,k}$ between the time-series $\mathbf{T}_i$ and a candidate shapelet s_d^k is the minimum distance between the candidate shapelet and any segment of length l extracted from $\mathbf{T}_i$, that is, $m_{i,k} = \min_{q=1,\dots L-l+1} \sqrt{\frac{1}{l}\sum_{p=1}^{l}(\mathbf{T}_{i,d,p+q-1} - s_{d,p}^k)^2}$, where $\mathbf{T}_{i,d,p+q-1}$ represents $(p+q-1)^{th}$ value in the dimension d in the instance $\mathbf{T}_i$ and s_p^k represents the p^{th} value in the candidate shapelet s_d^k of length l. Note that $m_{i,k}$ is normalized by dividing shapelet length, so that $m_{i,k}$ is independent of length l.

DEFINITION 3. *Shapelet transformation* (**M**): The minimum distance between candidate shapelets s_d^k and the time-series $\mathbf{T}_i$ indicates the degree of similarity between a candidate shapelet s_d^k and the time-series $\mathbf{T}_i$ examples. This representation is known as shapelet-transformed data [11]. The representation $\mathbf{M} \in \mathbb{R}^{I \times K}$ reduces the dimensionality of the original time-series since number of candidate shapelets K is less than the length of the time-series L.

DEFINITION 4. *Start-time* (B_i^k): The start-time of a candidate shapelet s_d^k in $\mathbf{T}_i$ is the point q from which the candidate shapelet s_d^k has minimum distance to $\mathbf{T}_i$, that is,

$$B_i^k = \underset{q}{\operatorname{argmin}} \sqrt{\frac{1}{l}\sum_{p=1}^{l}(\mathbf{T}_{i,d,p+q-1} - s_{d,p}^k)^2} \tag{3.1}$$

The benefit of using shapelet-orders to correctly classify time-series has been shown in Fig. 1. One straightforward option is to consider the relative position of two candidate shapelets in the time-series, that is, whether a candidate shapelet s^{k_1} occurs earlier (or later) than (or overlaps with) a candidate shapelet s^{k_2}. However, in most cases, the time gap between two candidate shapelets is much more informative. For example, if s^{k_1} occurs more than 10 time-points earlier than s^{k_2}, then the instance belongs to one class. Otherwise, the instance belongs to another class. Simply considering the relative position between the candidate shapelets fail to handle the time gap between the candidate shapelets. Therefore,

Algorithm 1 Selection of a random candidate shapelet

procedure SEARCH
Input: $T \in \mathbb{R}^{I \times D \times L}$, Accepted shapelet list $\mathcal{A}$, Rejected shapelet list $\mathcal{R}$, Distance threshold ε_d
  Draw random series: $i \sim \mathcal{U}\{1, \cdots, I\}$;
  Draw random dimension: $d \sim \mathcal{U}\{1, \cdots, D\}$;
  Draw random shapelet length: $l \sim \mathcal{U}\{1, \cdots, L\}$;
  Draw random start point: $p \sim \mathcal{U}\{1, \cdots, L-l+1\}$;
  Randomly selected candidate: $s^k \leftarrow \mathbf{T}_{i,d,p:p+l-1}$;
  if s^k is not similar to any previously accepted shapelets in $\mathcal{A}$ as well as any rejected shapelets in $\mathcal{R}$ **then**;
    Return s^k
  else
    $\mathcal{R} = \mathcal{R} \cup s^k$

we propose a scheme, named *TimeGap-based-orders*, which considers the time gap between a pair of candidate shapelets. Please note that the proposed order scheme incorporates and generalizes the scheme of considering the relative position of two shapelets.

DEFINITION 5. *TimeGap* $g_i(s^{k_1}, s^{k_2})$: Given two candidate shapelets s^{k_1} and s^{k_2} and a time-series $\mathbf{T}_i$, the time gap between two candidate shapelets is the difference of start-time of two shapelets in the $\mathbf{T}_i$, that is,

$$g_i(s^{k_1}, s^{k_2}) = B_i^{k_1} - B_i^{k_2}. \tag{3.2}$$

Note that the candidate shapelets s^{k_1} and s^{k_2} could be either from the same dimension, or from different dimensions (in case of multivariate time-series datasets), thus we omit d in their notations.

4 Model Description:

In this section, we introduce **P**airwise **S**hapelet-**O**rders **D**iscovery (PSOD) model for extracting informative shapelets and pairwise shapelet-orders for time-series classification. The proposed model computes the confidence of classifying a time-series instance to a particular class category. The confidence is calculated from two different spaces, shapelet-transformed space and shapelet-order space. We first discuss the process of identifying candidate shapelets, followed by the identification of candidate orders. In each case, a confidence measure is introduced to evaluate the quality of a candidate shapelet as well as a candidate order respectively.

Randomized shapelet candidate extraction: Inspired from the huge speed up by Grabocka et al. [10], we take a similar shapelets extraction approach to randomly select subsequences from time-series, and then evaluate it by computing classification accuracy. The steps to randomly select a candidate shapelet is summarized in Algorithm 1.

The primary idea of this method is to select a candidate shapelet from randomly chosen subsequences (lines 3-7) and prune similar candidate subsequences of same length (lines 8-11). The motivation behind randomly choosing subsequences lies in the fact that the majority of subsequences from time-series instances are similar, therefore it is computationally efficient to only consider a small set of non-redundant candidate segments which are helpful in classification. The distance threshold ε_d, obtained from the P percentile of distances between any pairs of random segments from time-series examples [10], prunes the search space of similar shapelets. The distance between a randomly selected subsequence s^k and any shapelet of same length in the accepted set $\mathcal{A}$ as well as rejected shapelets set $\mathcal{R}$ is calculated based on Definition 1. If the distance is larger than the threshold ε_d, then s^k will be considered as a candidate shapelet. Otherwise, it will be pruned and added in $\mathcal{R}$.

Class membership confidence in shapelet-transformed space: The shapelet-transformed space is a matrix $\mathbf{M}_{I\times K}$ of minimum distances between K accepted shapelets and I time-series instances where each element of the matrix is $m_{i,k}$. For a time-series instance $\mathbf{T}_i$, the shapelet-transformed space is a vector of size $1\times K$ denoted as $\mathbf{m}_i$. The probability p_{ij} of a time-series instance $\mathbf{T}_i$ selecting another instance $\mathbf{T}_j$ as its neighbor is calculated using the softmax over Euclidean distances in the shapelet-transformed space, that is,

$$p_{ij} = \frac{e^{\alpha||\mathbf{m}_i-\mathbf{m}_j||^2}}{\sum_{z=1\cdots I, z\neq i} e^{\alpha||\mathbf{m}_i-\mathbf{m}_z||^2}}\ , \qquad p_{ii}=0 \tag{4.3}$$

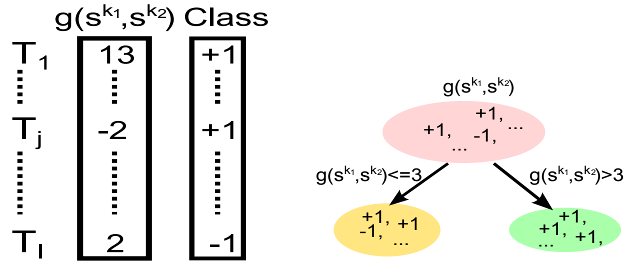
where α ($\alpha < 0$) is a parameter to control the precision of the function. When α is very small, e.g. $\alpha = -100$, the instance will have a high probability of choosing the instance with the smallest distance as its closest neighbor, which may make the model biased to the nearest neighbor.

The class membership confidence $p^{\mathbf{S}}_{i,c}$ of time-series instance $\mathbf{T}_i$ for class c in shapelet-transformed space is the sum of the probability of $\mathbf{T}_i$ selecting other instances $\mathbf{T}_j$ whose labels are c, that is,

$$p^{\mathbf{S}}_{i,c} = \sum_{Y_j=c} p_{ij} \tag{4.4}$$

where $Y_j = c$ represents that the label of time-series instance $\mathbf{T}_j$ is class c. Each time-series instance $\mathbf{T}_i$ shall have $|C|$ confidence values and the class with the highest probability shall be assumed to be the estimated class of the instance $\mathbf{T}_i$.

Pairwise shapelet-order extraction: Assume s^{k_1} is an accepted shapelet and s^{k_2} is a candidate shapelet. Before accepting s^{k_2}, we extract the potential orders between s^{k_1} and s^{k_2} using TimeGap-based-order scheme introduced in Sec. 3.



(a) Computing the time difference of s^{k_1} and s^{k_2} in I instances.

(b) Selecting the cut point $h = 3$, and separate the instances into two subsets.

Figure 2: An example of finding a TimeGap-based-order candidate.

For a pair of shapelets, s^{k_1} and s^{k_2}, the time gap $g_i(s^{k_1}, s^{k_2})$ between s^{k_1} and s^{k_2} related to an individual time-series instance $\mathbf{T}_i$ is calculated based on Eq. 3.2. For I training instances, a vector $\langle g_1(s^{k_1}, s^{k_2}), \cdots, g_I(s^{k_1}, s^{k_2})\rangle$ of length I is obtained. Then, the cut-point $h \in \{g_1(s^{k_1}, s^{k_2}), \cdots, g_I(s^{k_1}, s^{k_2})\}$ that separates the dataset into two subsets and maximizes the information gain is chosen. The left subset contains the instances which satisfy $g(s^{k_1}, s^{k_2}) \leq h$, and the right subset contains the instances which satisfy $g(s^{k_1}, s^{k_2}) > h$. An illustrated procedure is shown in Fig. 2. The entropy of both subsets are calculated. The one that has the smaller entropy will be selected as a candidate order. For example, if the entropy of the left subset is smaller, then $g(s^{k_1}, s^{k_2}) \leq h$ will be selected as a candidate order, otherwise, $g(s^{k_1}, s^{k_2}) > h$ will be considered as a candidate order.

Let o represent a candidate order. The class of a candidate order is determined by the class that has the highest number of instances in the subset with the smaller entropy. For example, in Fig. 2b, assume that the left subset has the smaller entropy and the number of instances with label $+1$ is more than the instances with other label, then the candidate order $g(s^{k_1}, s^{k_2}) \leq h$ shall be assigned to the class of $+1$. In this example, $o : g(s^{k_1}, s^{k_2}) \leq h$, and $Y_o = +1$.

The precision of the candidate order o of $Class = c$ is defined as $P(Class = c|o \text{ exists}) = \frac{P(o \text{ exists}|Class=c)\ P(Class=c)}{P(o \text{ exists})}$.

The confidence of the candidate order o of class = c is defined as a product of the precision of the candidate order and the probability of the intersection that the candidate order exists and belongs to class c, that is,

$$\mathbb{C}(o) = P(Class = c|o \text{ exists}) \times P(Class = c \cap o \text{ exists}) \tag{4.5}$$

Both terms in Eq. 4.5 are probabilities, thus the confidence measure for order o is a value between 0 and 1.

Updating class membership confidence using orders: Let $\mathbf{O}$ denote order space. The class membership confidence $p^{\mathbf{O}}_{i,c}$

Algorithm 2 Pairwise shapelet-orders discovery - training

1: **procedure** PSOD-TRAIN
2: **Input**: $T \in \mathbb{R}^{I \times D \times L}$, Labels $Y \in \mathbb{C}^I$
3: **Initialize**: Accepted Shapelets list $\mathcal{A} \leftarrow \emptyset$, Accepted order list $\mathcal{O} \leftarrow \emptyset$, Rejected shapelet list $\mathcal{R} \leftarrow \emptyset$;
4: **for** iteration = 1: $\mathbb{N}_1^{ILQD}$ **do**
5: $acc \leftarrow$ ACCURACY($\mathcal{A}, \mathcal{O}$);
6: $s^k \leftarrow$ SEARCH();
7: $\{\mathcal{A}, \mathcal{O}\} \leftarrow$ EVALUATE(s^k, acc);
Return $\mathcal{A}, \mathcal{O}$;

of time-series $\mathbf{T}_i$ for class c in the order space is calculated using the confidences of orders of class c that exist in $\mathbf{T}_i$,

$$p^{\mathbf{O}}_{i,c} = \mathbb{C}\left(\bigcup_{Y_{o_n}=c \,\cap\, o_n \text{ occurs in } \mathbf{T}_i} o_n\right) \tag{4.6}$$

For example, suppose two orders of class c exist in instance $\mathbf{T}_i$. The class membership confidence of instance $\mathbf{T}_i$ for class c from the order space is computed as $p^{\mathbf{O}}_{i,c} = \mathbb{C}(o_1 \cup o_2) = \mathbb{C}(o_1) + \mathbb{C}(o_2) - \mathbb{C}(o_1 \bigcap o_2) = \mathbb{C}(o_1) + \mathbb{C}(o_2) - \mathbb{C}(o_1) * \mathbb{C}(o_2)$. In a general case when there are multiple orders, Eq. 4.6 can be calculated according to the inclusion-exclusion principle of probability.

The initial class membership confidence for each time-series instance is computed in the shapelet-transformed space using Eq. 4.4. The confidence of the orders provides further evidence for or against the class membership for each time-series $\mathbf{T}_i$ instance to each class categories. Therefore, the updated class membership confidence of $\mathbf{T}_i$ when orders of class c occur can be computed as following,

$$P(Y_i = c|\mathbf{M}, \mathbf{O}) = p^{\mathbf{S}}_{i,c} \times p^{\mathbf{O}}_{i,c} \tag{4.7}$$

If no order of class c occurs in $\mathbf{T}_i$, then the class membership probability is penalized by $\frac{1}{C}$, that is,

$$P(Y_i = c|\mathbf{M}, \mathbf{O}) = p^{\mathbf{S}}_{i,c} \times \frac{1}{C} \tag{4.8}$$

This update rule is valid since we can assume that prior probability of an example being from class c is equal to $\frac{1}{C}$.

Pairwise shapelet-orders discovery Finally, we are in position to introduce the training phase of PSOD. The pseudo code is outlined in Algorithm 2. The model begins by searching a candidate shapelet using SEARCH() function (outlined in Algorithm 1), and then evaluating the candidate shapelet s^k as well as the potential pairwise shapelet-orders using EVALUATE() (outlined in Algorithm 3). This process (lines 5-7) is repeated within a limited number of iterations or stops when the accuracy of the model in the training set converges. The maximum number of iterations is upper bounded by the maximum number of candidate subsequences which is the product of the $I \times L \times Q \times D$ for a particular dataset. Q is the number of shapelet lengths to be evaluated.

Algorithm 3 Evaluate candidate shapelets

1: **procedure** EVALUATE
2: **Input**: Accepted shapelet list $\mathcal{A}$, Accepted order list $\mathcal{O}$, Rejected shapelet list $\mathcal{R}$, current accuracy acc, a candidate shapelet s^k;
3: tempAcc1 $\leftarrow$ ACCURACY($\mathcal{A}$, $\mathcal{O}$, s^k)
4: tempAcc2 = 0, tempOrder = 0;
5: **for** m = 1, $\cdots$, $|\mathcal{A}|$ **do**
6: Extract a order candidate o_m between s^k and s^m.
7: tempAcc2 $\leftarrow$ ACCURACY($\mathcal{A}$, $\mathcal{O}$, s^k, o_m);
8: **if** tempAcc2 >tempAcc1 **then**
9: tempAcc1 = tempAcc2
10: tempOrder = o_m
11: **if** tempAcc1 $> acc$ and tempOrder ! = 0 **then**
12: $\mathcal{O} \leftarrow \mathcal{O} \bigcup$ tempOrder;
13: $\mathcal{A} \leftarrow \mathcal{A} \bigcup \{s^k\}$;
14: **else**
15: **if** tempAcc1 $> acc$ and tempOrder == 0 **then**
16: $\mathcal{A} \leftarrow \mathcal{A} \bigcup \{s^k\}$;
Return $\mathcal{A}, \mathcal{O}$

In Algorithm 2 line 7, EVALUATE() returns an updated list of accepted shapelets and an updated list of orders. In EVALUATE() (Algorithm 3), a candidate shapelet s_k is first evaluated (line 3) by calculating the classification accuracy (outlined in Algorithm 4). Then, the potential orders between s^k and any already accepted shapelet in $\mathcal{A}$ are evaluated (lines 5-10). Only the candidate order, which yields the highest accuracy compared to other orders and s^k alone, is considered. If the overall classification accuracy is improved, then the candidate order and the candidate shapelet (lines 11-13) are selected. Otherwise, the candidate order is discarded. The candidate shapelet is accepted if it alone improves the accuracy (lines 15-16). At the beginning, for the first candidate shapelet, the accuracy is computed only in shapelet-transformed space, as no order exists.

While computing accuracy (outlined in Algorithm 4), if the accepted shapelet-orders list $\mathcal{O}$ is empty, then the class membership confidence is calculated only in the shapelet-transformed space (line 6). If $\mathcal{O}$ is not empty and multiple orders of class c exist in the instance, then the class membership confidence is calculated based on Eq. 4.7, otherwise, it is computed according to Eq. 4.8. Note that the class membership confidence is calculated for each class (Algorithm 4, line 5), and the predicated class of instance $\mathbf{T}_i$ is the class with the highest probability (Algorithm 4, line 12).

The pseudocode of the testing phase of PSOD is outlined in Algorithm 5. For a test instance $\mathbf{T}'_i$, the minimum distances between the K accepted shapelets and $\mathbf{T}'_i$ are computed according to Definition 1. The probability of $\mathbf{T}'_i$ selecting an instance $\mathbf{T}_j$ in the training dataset as its neighbor

Algorithm 4 Classification accuracy

procedure ACCURACY
Input: T, Labels Y, Accepted shapelet list $\mathcal{A}$, Accepted order list $\mathcal{O}$, a candidate shapelet s^k and a candidate order o
  $acc = 0, \mathcal{A} \leftarrow \mathcal{A} \bigcup \{s^k\}, \mathcal{O} \leftarrow \mathcal{O} \bigcup \{o\}$
  for $i = 1, \cdots, I$ **do**
    for $c = 1, \cdots, C$ **do**
      $P(Y_i = c|\mathbf{M}, \mathbf{O}) = p^{\mathbf{S}}_{i,c}$;
      if $|\mathcal{O}| > 0$ **then**
        if Order of class = c exist **then**
          $P(Y_i = c|\mathbf{M}, \mathbf{O}) = p^{\mathbf{S}}_{i,c} \times p^{\mathbf{O}}_{i,c}$;
        else
          $P(Y_i = c|\mathbf{M}, \mathbf{O}) = p^{\mathbf{S}}_{i,c} \times \frac{1}{C}$;
    $\hat{Y}_i = \text{argmax}_c\, P(Y_i = c|\mathbf{M}, \mathbf{O})$;
    if $\hat{Y}_i == Y_i$ **then**
      $acc = acc + 1$;
  $acc = acc/I$;
  $\mathcal{A} \leftarrow \mathcal{A} \setminus \{s^k\}, \mathcal{O} \leftarrow \mathcal{O} \setminus \{o\}$
  Return acc

Algorithm 5 Pairwise shapelet-orders discovery - testing

procedure PSOD-TEST
Input: $T' \in \mathcal{R}^{I_{Test} \times D \times L}$, $T \in \mathcal{R}^{I_{Train} \times D \times L}$, Accepted Shapelets list $\mathcal{A}$, Accepted order list $\mathcal{O}$;
  for i = 1: $\mathcal{I}_{Test}$ **do**
    for j = 1: $\mathcal{I}_{Train}$ **do**
      $p_{ij} = \frac{e^{\alpha||\mathbf{m}_i - \mathbf{m}_j||^2}}{\sum_{z=1 \cdots I, z \neq i} e^{\alpha||\mathbf{m}_i - \mathbf{m}_z||^2}}$;
      for $c = 1, \cdots, C$ **do**
        $P(Y_i = c|\mathbf{M}, \mathbf{O}) = p^{\mathbf{S}}_{i,c} = \sum_{Y_j = c} p_{ij}$;
        if $|\mathcal{O}| > 0$ **then**
          if Order of class = c exist **then**
            $P(Y_i = c|\mathbf{M}, \mathbf{O}) = p^{\mathbf{S}}_{i,c} \times p^{\mathbf{O}}_{i,c}$;
          else
            $P(Y_i = c|\mathbf{M}, \mathbf{O}) = p^{\mathbf{S}}_{i,c} \times \frac{1}{C}$;
    $\hat{Y}_i = \text{argmax}_c\, P(Y_i = c|\mathbf{M}, \mathbf{O})$;
  Return $\hat{Y}$

is computed based on Eq. 4.3, and the class membership confidence for $\mathbf{T}'_i$ in shapelet-transformed space is computed using Eq. 4.4. Next, the selected orders in the order list $\mathcal{O}$ will be checked whether they occur in $\mathbf{T}'_i$. For class = c, if some orders belonging to class c occur, the class membership will be updated according to Eq. 4.7, otherwise, it will be updated based on Eq. 4.8. The class with the highest membership confidence will be selected as the final predicted class.

Analysis of runtime: Given a dataset of I training examples of length L having C classes, the total number of shapelet candidates has an order of $\mathcal{O}(IL^2)$. We would like to recall that Eq. 4.4 (class membership confidence) is computed for each class for each time-series. Thus the worst-case time complexity to identify the best shapelet is $\mathcal{O}(I^2L^4C)$. Using the distance threshold ϵ_d reduces the number of total shapelet candidates to an order of $\mathcal{O}(fIL^2)$ where f is a fraction of the total candidate shapelets that are evaluated, denoted by $f = \frac{\#Accepted\ shapelts + \#Rejected\ shapelets}{IL^2}$. The time complexity is thus lowered to $\mathcal{O}(fI^2L^4C)$. Furthermore, the discovery of shapelet orders among the accepted shapelets increases the time complexity of the algorithm. The total number of possible shapelet orders evaluated is upper bounded by the total number of accepted shapelets. The running time to accept the best candidate shapelet order is $\mathcal{O}(\#Accepeted\ shapelets \times I \times C)$. Therefore, the overall running time can be denoted as $\mathcal{O}(fIL^2 \times (IL^2C + \#Accepted\ shapelets \times I \times C))$. In Table 1 we empirically compare PSOD's training time with state-of-art shapelet based methods on different datasets.

5 Experimental Evaluation

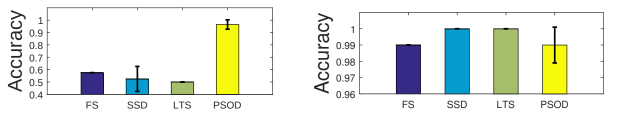
The proposed model was evaluated extensively on both univariate and multivariate real-world datasets. Additionally, two synthetic datasets were used to highlight the advantage of leveraging shapelet-orders over shapelet information. The univariate datasets were obtained from the UEA & UCR Time Series Classification Repository [1]. Details about each univariate datasets can be viewed on repository's website[1]. Moreover, we chose 6 multivariate datasets used in [10] to highlight the advantage of shapelet-orders in real-world multidimensional time-series datasets.

The focus of the proposed model is to improve upon shapelet-based classification models and since there is no existing model which considers orders between shapelets, we compared the proposed method, PSOD, with 5 state-of-art shapelet-based time-series classification models.

- Scalable Shapelet Discovery (SSD) [10]: This method is a fast procedure to extract random shapelets from the time-series dataset. Our PSOD method generalizes SSD by taking pairwise shapelet-orders into account and this is why these two methods are compared.
- Learning Time-Series Shapelets (LTS) [9]: The LTS generalizes shapelets, thus it obtains more accurate prediction on most datasets. However, LTS is not directly applicable to multivariate datasets while PSOD is applicable.
- Fast Shapelets (FS) [20]: The FS algorithm discretize and approximates the shapelets rather than a complete search at each node of the decision tree. It is also not directly applicable to multivariate time-series datasets.
- Naive Shapelets[2] (NS) [19]: This is an naive extension of the FS algorithm where a d-dimensional multivariate time-

[1] The UEA & UCR Time Series Classification Repository, www.timeseriesclassification.com

[2] We implemented the model as original source code was not available.



(a) Orders between shapelets exist in the data. (b) No order between shapelets exists in the data.

Figure 3: Average accuracies of FS, SSD, LTS and PSOD on synthetic dataset where (a) orders between shapelets and (b)no order between shapelets exists in the data.

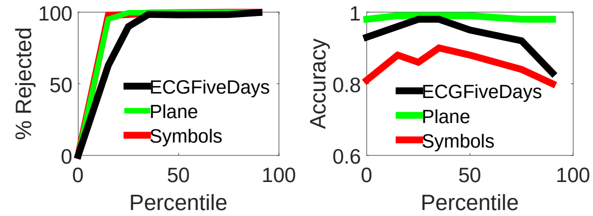
series example is converted into d univariate time-series instances and FS is applied to each equivalent univariate representation to learn a decision tree independently. The final label is determined via a majority voting scheme.

- Shapelet Forests[2] (SF) [19]: The SF algorithm combines the FS algorithm for univariate time series to build an ensemble of classifiers, one for each time series dimension in the multivariate time-series instances.

The default training and test sets were used in all experiments. The average accuracy of 5 trials was reported for each method. Three shapelet lengths were used, $l \in \{0.1L, 0.2L, 0.3L\}$. The distance threshold percentile (P) was set at $P = 0.35$ for all datasets. For parameter α, we consider two choices, -10 and -100, and chose the best one through internal cross validation in the training set. All experiments were run on a windows 10 machine with 32 GB RAM and Intel i7 quad core processor.

Results on synthetic datasets Two groups of synthetic time-series datasets were generated: (1) the orders of shapelets are different in two classes; (2) there is no order between shapelets. Two subsequences with specific patterns were considered. The first one is $y = \sin x, x \in [0, \pi]$, and the second one is $y = -\sin x, x \in [0, \pi]$. In the group of synthetic dataset where order matters, the first pattern always occurs before the second one in the data that is labeled as "Class 1", whereas the first pattern always occurs after the second one in the data that is labeled as "Class 2". A sample of time-series from both classes and two patterns are plotted in Fig. 1. In both synthetic datasets, the start-time of patterns were randomly selected, and the remaining points in the time-series follow Gaussian distribution $\mathcal{N}(0, 0.05)$. Besides, we added noise sampled from a product distribution comprised of $\mathcal{N}(0, 0.05)$ and $\mathcal{U}(0, 0.25)$ to the patterns. The length of time-series is 400, and 20 time-series were generated for each class in both training and test datasets.

Fig. 3a shows the classification accuracy obtained by all baselines and PSOD on the synthetic dataset where shapelets have different orders in two classes. PSOD has significantly outperformed all baselines in terms of classification accuracy. As expected, the accuracies obtained by all baselines are random, as there is no subsequence that could differentiate the class. Moreover, all baselines and PSOD perform comparable on the dataset where there is no order among



(a) Percentile Vs. % Shapelet rejected (b) Percentile Vs. Accuracy

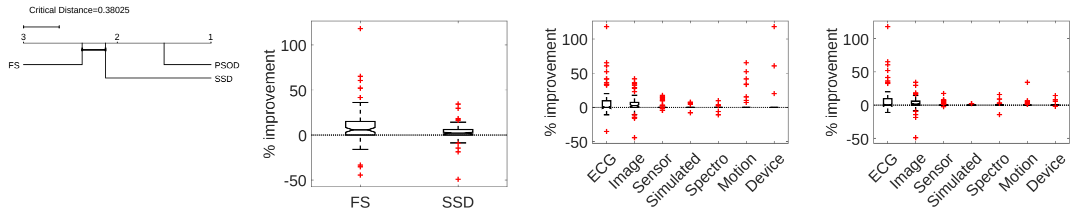
Figure 4: The effect of varying percentile parameter P on (a) number of shapelets rejected and (b) accuracy.

shapelets (Fig. 3b). Therefore, the benefit of taking shapelet-orders into account was more evident when temporal dependency among pairs of shapelets could differentiate the class.

Analysis of percentile (P) parameter We evaluated the sensitivity of the distance threshold ϵ_d in Algorithm 1 on three real-world datasets. The threshold distance used for pruning similar candidates has a significant effect on the quantity of rejected candidates. The distance threshold ϵ_d is controlled by the parameter P which denotes the percentile of distances. Larger values of P means that two subsequences which have large distance will be considered as similar. As indicated in Fig. 4a larger percentile values result in more candidate subsequences being rejected with a degradation in accuracy (shown in Fig. 4b).

Results on real-world univariate datasets We first compared the performance of PSOD versus all the baseline methods on 6 datasets that have a variety of properties in terms of time-series length and number of classes. The average accuracy and their training time (in brackets) are reported in Table 1. On the first group of datasets, *BeetleFly* and *Earthquark*, which have binary classes and moderate time series length, PSOD produced much better results than FS and SSD. Although PSOD and LTS obtained comparable accuracy on *Earthquake* dataset, PSOD only took a quarter of LTS's training time to finish training the model. On the second group of datasets, *HandOutline* and *StarLingthCu.*, which have long length, PSOD produced the most accurate results. The superiority of PSOD compared to LTS with respect to training time is also more clear. On the third group of datasets, *InsetW.* and *FaceAll*, which have multiple classes, PSOD still outperform FS and SSD with respect to accuracy. Table 1 revealed that (1) although PSOD attained slightly inferior results than LTS, it is efficient. (2) PSOD obtained better (or comparable) classification accuracy than FS and SSD on all 6 datasets. Although SSD was the fastest, the training time of PSOD is better than *FS* and *LTS*.

Next, we evaluated the effectiveness of PSOD on 75 real-world univariate datasets obtained from 7 categories namely ECG, Image, Sensor, Simulated, Spectro, Motion



(a) Statistical significance (b) Accuracy improvement (c) Improvement over FS (d) Improvement over SSD

Figure 5: (a) Critical difference diagram. (b) Average percentage improvement of PSOD over FS and SSD across 75 univariate datasets. (c) Accuracy improvement over FS w.r.t 7 categories. (d) Accuracy improvement over SSD w.r.t 7 categories.

Table 1: Average accuracy (*Training time in minutes*) of 6 different real-world time-series datasets.

Dataset	C	L	FS	LTS	SSD	$PSOD$
BeetleFly	2	512	0.65 (*0.3*)	0.7 (*1.3*)	0.7 (*0.001*)	**0.75** (*0.2*)
Earthquake	2	512	0.71 (*27.7*)	**0.74** (*41*)	0.68 (*1.2*)	0.73 (*11*)
HandOutline	2	2709	0.81 (*2051*)	>2 days	0.81 (*0.8*)	**0.86** (*627*)
StarLightCu.	3	1024	0.91 (*131*)	0.85 (*920*)	**0.94** (*1.5*)	**0.94** (*781*)
InsectW.	11	256	0.47 (*2.6*)	**0.60**(*157*)	0.45 (*0.3*)	0.48 (*12*)
FaceAll	14	131	0.62 (*4.1*)	**0.74** (*303*)	0.73 (*0.1*)	0.73 (*26*)

Table 2: Average accuracy of NS, SF, SSD and PSOD on 6 multivariate datasets over 5 trials.

Dataset	D	C	L	NS	SF	SSD	$PSOD$
mhealth	23	12	51 - 3431	0.75	0.78	0.73	**0.81**
Characters	3	20	109 - 205	0.90	0.96	**0.97**	**0.97**
HMP	3	21	125 - 9318	0.70	**0.73**	0.71	**0.73**
RealDisp	117	33	318 - 5643	0.67	0.69	0.71	**0.78**
Wafer[3]	6	2	126 - 146	0.85	**0.91**	0.87	0.88
ECG[3]	3	2	68 - 104	0.73	0.75	**0.76**	**0.76**

and Device. We compared PSOD against FS and SSD only, since LTS is very costly for longer time-series datasets. The significance test, calculated based on [6], shows that PSOD is significantly better than FS and SSD at the 5% level (see Fig 5a). The percentage of improvement of PSOD over FS and SSD (plotted in Fig. 5b) shows that across 75 datasets PSOD has significantly improved the classification accuracy. On average PSOD was 8.9% more accurate than FS and 2.6% better than SSD.

The percentage improvement of PSOD over FS and SSD for datasets from different categories are shown in Fig. 5c and Fig. 5d respectively. Clearly, PSOD improved the classification accuracy for most of the datasets from different categories, especially from Motion and Device categories. For the few datasets, that PSOD failed to improve the accuracy, it is possible that the procedure of randomly selecting shapelets may have selected bad-quality shapelets which decreased the performance of PSOD (discussed in section 5). Table 1 and Fig. 5 clearly indicate the superiority of PSOD.

Results on multivariate datasets We further assessed the proposed model on 6 real-world multivariate time-series datasets. Their characteristics and the average accuracy of 5 trials are shown in Table 2. We compared PSOD with three multivariate time-series classification techniques namely NS, SF and SSD. Table 2 shows that PSOD produced higher or comparable accuracy compared to three baselines on 5 datasets, except *Wafer*. PSOD achieves higher accuracy on *Wafer* dataset compared to NS by 3% and 1% higher compared to SSD, however SF achieves 3% higher accuracy than PSOD.

Discussion From the experiments, we noticed that (1) in most datasets, PSOD is more accurate than SSD and FS. The quality of the proposed order extraction schemes is dependent on the quality of extracted shapelets. Poor quality shapelets may lead to poor quality orders and consequently result in lower classification accuracy. Since in PSOD, subsequence candidates are randomly extracted, the quality of shapelets may compromise to speedup the shapelet extrac-

[3] Balanced binary datasets were used.

tion procedure. (2) PSOD is applicable to both univariate and multivariate time-series, especially with shorter length, which is common in many domains. For longer time-series, the efficiency of PSOD may vary, because the computational complexity of PSOD increases with the number of potential candidate shapelets. One future direction is to generate shapelets of good quality by generalizing subsequences, as well as, developing more efficient methods for learning shapelet-orders with smaller time complexity.

6 Conclusion

In this paper, we propose a novel order-generation scheme, *TimeGap-based-orders*, to capture temporal dependency among shapelets, and present a novel model PSOD aimed to extract both informative shapelets and shapelet-orders. From the extensive experimental results, we found that (1) the PSOD model produces more accurate classification results compared to state-of-the-art alternatives in majority of the datasets; (2) the proposed order-generation scheme is generalized, and can identify and extract shapelet-orders from both univariate and multivariate time-series datasets.

References

[1] 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 Min. Knowl. Discov, 31 (2017), pp. 606–660.

[2] A. Bostrom and A. Bagnall, *A shapelet transform for multivariate time series classification*, CoRR, abs/1712.06428 (2017).

[3] M. S. Cetin, A. Mueen, and V. D. Calhoun, *Shapelet ensemble for multi-dimensional time series.*, in SDM, 2015, pp. 307–315.

[4] T. chung Fu, *A review on time series data mining*, Engg. Appli. of Arti. Intelli., 24 (2011), pp. 164 – 181.

[5] G. Das, K.-I. Lin, H. Mannila, G. Renganathan, and P. Smyth, *Rule discovery from time series*, in SIGKDD, 1998, pp. 16–22.

[6] J. Demšar, *Statistical comparisons of classifiers over multiple data sets*, Jour. of Machh. Learn. Res., 7 (2006), pp. 1–30.

[7] M. Ghalwash, V. Radosavljevic, and Z. Obradovic, *Extraction of interpretable multivariate patterns for early diagnostics*, in ICDM, 2013, pp. 201–210.

[8] M. F. Ghalwash, V. Radosavljevic, and Z. Obradovic, *Utilizing temporal patterns for estimating uncertainty in interpretable early decision making*, in SIGKDD, 2014, pp. 402–411.

[9] J. Grabocka, N. Schilling, M. Wistuba, and L. Schmidt-Thieme, *Learning time-series shapelets*, in SIGKDD, 2014, pp. 392–401.

[10] J. Grabocka, M. Wistuba, and L. Schmidt-Thieme, *Fast classification of univariate and multivariate time series through shapelet discovery*, Knowl. Info. Syst., 49 (2016), pp. 429–454.

[11] J. Hills, J. Lines, E. Baranauskas, J. Mapp, and A. Bagnall, *Classification of time series by shapelet transformation*, Data Min. Knowl. Discov, 28 (2014), pp. 851–881.

[12] L. Hou, J. T. Kwok, and J. M. Zurada, *Efficient learning of timeseries shapelets*, in AAAI, 2016, pp. 1209–1215.

[13] I. Karlsson, P. Papapetrou, and H. Boström, *Generalized random shapelet forests*, Data Min. Knowl. Discov, 30 (2016), pp. 1053–1085.

[14] W. Leigh, N. Modani, R. Purvis, and T. Roberts, *Stock market trading rule discovery using technical charting heuristics*, Exp. Sys. with Appli., 23 (2002), pp. 155 – 159.

[15] J. Lines, L. M. Davis, J. Hills, and A. Bagnall, *A shapelet transform for time series classification*, in SIGKDD, 2012, pp. 289–297.

[16] T. Mirowski, S. Roychoudhury, F. Zhou, and Z. Obradovic, *Predicting poll trends using twitter and multivariate time-series classification*, in SocInfo, Springer, 2016, pp. 273–289.

[17] A. Mueen, E. Keogh, and N. Young, *Logical-shapelets: An expressive primitive for time series classification*, in SIGKDD, 2011, pp. 1154–1162.

[18] O. P. Patri, R. Kannan, A. V. Panangadan, and V. Prasanna, *Multivariate time series classification using inter-leaved shapelets*, in NIPS Time Series Workshop, 2015.

[19] O. P. Patri, A. B. Sharma, H. Chen, G. Jiang, A. V. Panangadan, and V. K. Prasanna, *Extracting discriminative shapelets from heterogeneous sensor data*, in IEEE BIG DATA, 2014, pp. 1095–1104.

[20] T. Rakthanmanon and E. Keogh, *Fast shapelets: A scalable algorithm for discovering time series shapelets*, in SDM, 2013, pp. 668–676.

[21] S. Roychoudhury, M. F. Ghalwash, and Z. Obradovic, *False alarm suppression in early prediction of cardiac arrhythmia*, in BIBE, 2015, pp. 1–6.

[22] S. Roychoudhury, M. F. Ghalwash, and Z. Obradovic, *Cost sensitive time-series classification*, in ECML/PKDD, 2017, pp. 495–511.

[23] M. Shokoohi-Yekta, Y. Chen, B. Campana, B. Hu, J. Zakaria, and E. Keogh, *Discovery of meaningful rules in time series*, in SIGKDD, 2015, pp. 1085–1094.

[24] G. Tatavarty, R. Bhatnagar, and B. Young, *Discovery of temporal dependencies between frequent patterns in multivariate time series*, in 2007 IEEE Symp. on CIDM, 2007, pp. 688–696.

[25] J. Ting, T.-C. Fu, and K. F.-L. Chung, *Mining of stock data: Intra- and inter-stock pattern associative classification*, in DMIN, 2006.

[26] Z. Xing, J. Pei, P. S. Yu, and K. Wang, *Extracting interpretable features for early classification on time series.*, in SDM, 2011, pp. 247–258.

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

[28] J. Zakaria, A. Mueen, and E. Keogh, *Clustering time series using unsupervised-shapelets*, in ICDM, 2012, pp. 785–794.

[29] Q. Zhang, J. Wu, H. Yang, Y. Tian, and C. Zhang, *Unsupervised feature learning from time series*, in IJCAI, 2016, pp. 2322–2328.