LEVERAGING TEMPORAL SUBSEQUENCES FOR TIME-SERIES CLASSIFICATION

A Dissertation
Submitted to
the Temple University Graduate Board

In Partial Fulfillment
of the Requirements for the Degree of
DOCTOR OF PHILOSOPHY

---

by
Shoumik Roychoudhury
May 2020

Examining committee members:

Dr. Zoran Obradovic, Dissertation Advisory Chair, Department of Computer and Information Sciences
Dr. Slobodan Vucetic, Department of Computer and Information Sciences
Dr. Kai Zhang, Department of Computer and Information Sciences
Dr. Iyad Obeid, External Member, Department of Electrical and Computer Engineering

ABSTRACT

Research on time-series classification has garnered importance among practitioners in the data mining community. A major reason behind the ever-increasing interest among data-miners is the plethora of time-series data available from a wide range of real-life domains. Temporal-ordered data from a variety of sensor-based domains such as wearable devices, smart homes, industrial monitoring, medical diagnosis, etc. provide classification challenges more akin to real-world scenarios. Thus, building more robust time-series classification models is imperative.

One group of popular models focuses on identifying short discriminative temporal patterns (subsequences) from the time-series for classification. These temporal subsequences, known as shapelets, are local patterns that can be used to uniquely identify the target class of a time-series instance. In this dissertation, I explore two real-world challenges pertaining to shapelet based time-series classification models and provide solutions to mitigate those challenges.

In the first challenge, the problem of cost-sensitive learning in time-series classification is explored. First, the problem of highly imbalanced time-series classification using shapelets is investigated. The current state-of-the-art approach learns generalized shapelets along with weights of the classification hyperplane via a classical cost-insensitive loss function. Cost-insensitive loss functions tend to treat different misclassification errors equally, and thus, models are usually biased towards examples of the majority class. In this research, the generalized shapelets learning framework is extended and a cost-

sensitive learning model is proposed. Instead of incorporating the misclassification cost as prior knowledge, as was done by other published methods, a constrained optimization problem was formulated to learn the unknown misclassification costs along with the shapelets and their weights. Secondly, I focus on the problem of cost-sensitive early classification in time-series datasets. High false alarm rates in intensive care units (ICUs) cause desensitization among care providers, thus risking patients' lives. Providing early detection of true and false cardiac arrhythmia alarms can alert hospital personnel and avoid alarm fatigue. This will ensure hospital personnel can act only on true life-threatening alarms, hence improving efficiency in ICUs. Furthermore, suppressing false alarms cannot be an excuse to suppress true alarm detection rates. In this study, a cost-sensitive approach for false alarm suppression while keeping near perfect true alarm detection rates was investigated using a confidence estimate for shapelets matching.

In the second challenge, the temporal dependencies among shapelets are explored. 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, a scheme is explored to extract informative orders among shapelets by considering the time gap between pairs of shapelets. In this realm, two models are proposed, Pairwise Shapelet-Orders Discovery (PSOD) and Learning pairwise Orders and Shapelets (LOS), which extracts both informative shapelets and shapelet-orders and incorporates the shapelet-transformed space with shapelet-order space for time-series classification. The two proposed models are contrasting approaches in the time-series classification paradigm. The PSOD is a search-based greedy procedure to extract unique shapelets and identify orders among the selected shapelets. On the other hand, LOS is an optimization-based approach to extract shapelet-orders among learned generalized shapelets. However, in both the hypotheses, the extracted pairwise shapelet-orders could increase the confidence of the prediction and further improve the classification

performance. The experimental results provide evidence that when considering shapelet-orders, classification accuracy is significantly improved on average over baseline methods. To the best of my knowledge, these are the first work that proposes formal methodologies to extract shapelet-orders and present augmented space of shapelets and shapelet-orders.

To my wife Shinjini.

To my parents Susanta and Arunima.

To my sister Shrabasti and her husband Aniruddha.

To my sister-in-law Shaeri, her husband Ashish and my nieces Aaliyah and Arshi.

To my father-in-law Sankar and my late mother-in-law Madhumita.

To all of my friends.

Thank you for your constant support and love
throughout the writing of this dissertation
and for always being there for me
and
A special thanks to all the medical personnel
fighting at the hospitals throughout the world
You are the true heroes of our generation.

ACKNOWLEDGEMENTS

I would like to express my sincerest gratitude to my advisor Prof. Zoran Obradovic for his professional guidance and endless support for my research and graduate studies at Temple University. I deeply appreciate his comprehensive and thorough knowledge of research topics and his collaborative initiative for high-quality works. I have tremendously benefited from his creative spirit, thought-provoking scholarship, valuable suggestion and constant encouragement, which have helped develop my research skills and motivated me to reach where I am today. Without his persistent assistance and great patience, the completion of this dissertation could not have been possible. Special thanks to Dr. Slobodan Vucetic, Dr. Kai Zhang and Dr. Iyad Obeid, great professors and researchers from whom I learned a lot, for serving on my dissertation committee and for providing me with useful comments and valuable suggestions. I would especially like to thank my collaborators Dr. Fang Zhou and Dr. Mohamed Ghalwash for being coauthors on most of my papers. I would like to thank them for their enthusiasm, pertinacity, and impeccable working habits. The discussions I had with them led to our significant achievements and papers. I thank professors and staff at the Department of Computer Information Sciences for making such a friendly and pleasant working environment. I would especially like to thank Charles Rorke, Christopher Bryant, Julie Skrocki, Hailey King, Prof. Justin Y. Shi and Prof. Eduard Dragut for always assisting graduate students. Special thanks to professors Richard Beigel, Slobodan Vucetic, Daniel Yee and Xiuqi Li, for whom I had worked as a teaching assistant. Their experience and enthusiasm were of great inspiration throughout my years of teaching. I am also grateful to all my students at the Department of Computer and Information Sciences. I thank my fellow lab-mates, postdocs and visiting scholars at the Center for Data Analytics and Biomedical Informatics: Alexey, Ana K., Ana S., Branimir, Branko, Chao, Dusan, Djole, Ivan, Jelena G., Jelena S., Jesse, Jumanah, Kosta, Marija, Martin, Milan B., Milan V., Milos J., Milos J., Milos R., Mladen, Mohamed G., Mohammad A., Nancy, Nima, Noor, Nouf, Tamiris, Tom, Tijana, Vladan, Vladisav, Xi for sharing their knowledge and perspectives during our lab meetings. I would like to acknowledge the support of IQVIA and the Office of Naval Research's SBIR program during my studies. I would also like to thank my mentors at MERL for giving the opportunity to work on diverse and exciting projects during my summer internship. Those were invaluable and unforgettable experiences. Finally, I would like to express my deep appreciation to my family for their love, kindness, inspiration and contribution. I would like to thank all my friends back home in India, here in Philadelphia and across the United States of America for being with me throughout these years. I am sincerely indebted to my wife, Shinjini Datta, for her incredible patience, support and tolerance during my Ph.D. study. Words fail to sufficiently convey my immense gratitude to her.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The dissertation is motivated by the increasing popularity of research in time-series classification in machine learning. Time-series classification can be defined as learning how to assign labels to time-series. A time-series is a collection of temporal ordered observations collected at equal intervals. The task of time-series classification consists of learning a classifier from a time-series dataset in order to map from the space of temporal inputs to a space of class variable categories. Research on time-series classification has garnered importance among practitioners in the data mining community. A major reason behind the ever-increasing interest among data-miners is the plethora of time-series data available from a wide range of real-life domains. Temporal ordered data from areas such as financial forecasting, medical diagnosis, weather prediction, etc. provide classification challenges more akin to real-world scenarios. Thus, building more robust time-series classification models is imperative.

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 (Ye and Keogh, 2009a), are local patterns that can be used to characterize a time-series instance for determining the time-series example's class membership. In this dissertation, two real-world challenges pertaining to shapelet based time-series classification models are explored. In the first challenge (Chapter 2 and Chapter 3), the problem of cost-sensitive learning in time-series classification is explored. In the second challenge (Chapter 4 and Chapter 5), the problem of extracting temporal dependencies among shapelets is studied and two contrasting frameworks time-series classification are proposed that leverages temporal dependency information among pairwise shapelets to improve the time-series classification performance.

In Chapter 2 the variable cost of sensitivity and specificity measurements in medical applications is investigated. More specifically, the problem of high false alarm rates in intensive care units (ICUs) which cause desensitization among care providers, thus risking patients' lives is explored. A shapelet based time-series classification framework is proposed that leverages the temporal uncertainty measurements of shapelet matches to estimate the confidence of early classification predictions of time-series, therefore providing a cost-sensitive prediction model. Providing early detection of true and false cardiac arrhythmia alarms can alert hospital personnel and avoid alarm fatigue, so that they can act only on true life-threatening alarms, hence improving efficiency in ICUs. However, suppressing false alarms cannot be a justification to suppress true alarm detection rates. In this study, a cost-sensitive approach is proposed for false alarm suppression while keeping near perfect true alarm detection rates.

In Chapter 3 the cost-sensitive learning with respect to algorithm-level modifications for highly imbalanced time-series classification using shapelets is explored. Cost-insensitive loss functions tend to treat different misclassification errors equally and thus, models are usually biased towards examples of majority class. The rare class (which will be referred to as positive class) is usually the important class and a false negative error is always costlier than a false positive error. Traditional 0-1 loss functions fail to differentiate between these two types of misclassification errors. Instead of using standard 0-1 loss functions a variable misclassification costs is introduces to minimize the conditional risk. By strongly penalizing false negative mistakes the decision boundaries is pushed away from the majority classes, thus leading to an improvement in generalization error in minority classes. The generalized shapelets learning framework is extended and a cost-sensitive learning model is proposed. Instead of incorporating the misclassification cost as a prior knowledge, as was done by other published methods, a constrained optimization problem is proposed to *learn* the unknown misclassification costs along with the shapelets and their weights.

In Chapter 4 the temporal dependency among shapelets in time-series datasets is leveraged to improve time-series classification accuracy. 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. In this new framework, a scheme to extract informative orders among shapelets is explored by considering the time gap between two shapelets. In addition, a model, Pairwise Shapelet-Orders Discovery, is proposed 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 improve the classification performance.

In Chapter 5 the caveats of the Pairwise Shapelet-Orders Discovery model are highlighted. The previous model which is based on a random selection of candidate shapelets does not guarantee the selection of optimal shapelets. This, in turn, may lead to poor quality shapelet-orders. It is proven that learning shapelets, instead of searching, guarantees near-optimal shapelets thus decreasing generalization error. However, the costly initialization approach for learning generalized shapelets significantly limits its scalability in large time-series datasets. In this chapter the problem of leveraging temporal dependencies among generalized shapelets from randomly initialized subsequences by jointly learning from the shapelet-transform space and the shapelet-order space is studied. The underlying hypothesis is that leveraging the temporal dependency information among generalized shapelets improves the classification performance. Furthermore, introducing a randomized subsequence initialization for learning generalized shapelets allows for a more scalable shapelet learning approach. The problem of leveraging temporal dependencies among generalized shapelets from randomly initialized subsequences is addressed by jointly learning from the shapelet-transform space and the shapelet-order space. The underlying hypothesis is that leveraging the temporal dependency information of generalized shapelets improves the classification performance. Furthermore, introducing a randomized subsequence initialization for learning generalized shapelets allows a more scalable shapelet learning approach.

CHAPTER 2

COST-SENSITIVE EARLY TIME-SERIES CLASSIFICATION

2.1 Introduction

Suppressing high false alarm rates from bedside monitors in intensive care units (ICUs) has been a topic of special interest in the last decade (Aboukhalil et al., 2008; Li and Clifford, 2012; Behar et al., 2013; Salas-Boni et al., 2014; Scalzo et al., 2013). Alarm fatigue among care providers inside ICUs due to the high percentages of bedside monitor false alarms, has been identified as one of the top 10 medical hazards (Jr., 2012). Alarm fatigue results in desensitization among care providers, which ultimately can lead to lower standards of care to patients and also result in fatal consequences (Drew et al., 2014). Artifacts, noise and missing values are some primary factors that corrupt the physiological data from bedside monitors, causing high false alarm rates.

Different approaches have been applied to reduce false alarm rates. One direction is to determine the quality of the ECG signal, based on the fact that noisy signals are more prone to trigger false alarms. For example, Behar et al. (Behar et al., 2013) proposed several novel ways of measuring ECG signal quality. Another direction consists of data fusion methods where extra non-ECG waveform data, such as invasive arterial

Table 2.1: Weighted accuracy comparison between the proposed approach to the state-of-the-art on asystole cardiac alarms (ASYS) and ventricular tachycardia alarms (VTACH).

Dataset	Behar et al.(Behar et al., 2013)	Proposed approach
ASYS	45.96±14.33	**65.05±7.55**
VTACH	32.80±8.32	**48.56±7.41**

blood pressure (ABP) and photoplethysmogram (PPG) (Li and Clifford, 2012; Sayadi and Shamsollahi, 2011) are incorporated. These non-ECG waveforms are assumed to be highly correlated to ECG, and consequently could be used to identify the alarm types. Recently, several methods were developed to suppress the false ventricular tachycardia alarms without the need for additional non-ECG waveforms, which resulted in reduction of true alarm detection (Behar et al., 2013; Salas-Boni et al., 2014). Their approach is based on features extracted from the ECG signal 20 seconds prior to a triggered alarm. All aforementioned methods extract statistical features from the ECG signals and feed them into a classifier, which often results in a black-box approach. However, in medical applications, it is important not only to provide accurate prediction but also to provide interpretable results, such that medical experts get insights about the prediction.

In this chapter a *cost-sensitive* classification model for *early* and *interpretable* prediction of life threatening arrhythmia alarms is characterized. The objective of the prediction model is to suppress false alarms while keeping true alarm detection rates high. In addition, by identifying alarms early, the response time of the medical personnel can be improved in the event of life-threatening arrhythmia alarms, and the alarm fatigue problem can be reduced among care providers. In Table 2.1, the effectiveness (weighted accuracy in Eq. 2.11) of the proposed approach to suppress a large percentage of false alarms for two datasets as compared with the current state-of-the-art method is shown.

Table 2.2: Properties (interpretability, earliness, uncertainty, and false alarm suppression (FAS) used to categorize the methods.

	(Li and Clifford, 2012)	(Behar et al., 2013)	EDSC (Xing et al., 2009)	Proposed approach
Intrepretabilty	×	×	✓	✓
Earliness	×	×	✓	✓
Uncertainty	×	×	×	✓
FAS	✓	✓	×	✓

2.2 Contribution

The contributions of this study, summarized in Table 2.2, are the following:

- A time-series classification model is characterized to provide more *accurate* prediction (high true alarm detection and false alarm suppression) than the state-of-the-art methods on arrhythmia alarms.

- *Interpretable* results are provided in order to explain the rationale of the prediction, whereas all other published methods are black-box.

- *Early* prediction before the alarm happens is achieved, which helps the practitioners to respond early to the alarm, whereas all other methods provide prediction at the time when alarms happen.

- A *cost-sensitive* model to achieve the desired level of false alarm suppression rates is proposed.

2.3 Related work

Early Classification of time-series

In the field of time-series classification, early classification of time-series has gained popularity (Xing et al., 2011), especially in application areas where critical time sensitive decision making is required, such as early warning of diseases (Ghalwash et al., 2013a). The principal objective of early classification models for time-series is to predict the label of the alarm as the ECG signal is progressively recorded and before the alarm happens. If the *observed* signal is insufficient to make an accurate prediction, more ECG signal data are used and the process is repeated until the time when the alarm happens. Early prediction of life-threatening cardiac alarms would allow care providers inside ICUs to be alert at the time of (or even before) true arrhythmia alarm events, and at the same time would automatically suppress false arrhythmia alarms.

Interpretable Early Classification Model

Medical experts tend to favor interpretable methods which provide visual clarification of prediction results rather than black-box methods. A method called Early Distinctive Shapelet Classification (EDSC) was proposed to provide interpretable early classification results (Xing et al., 2011). The method extracts local discriminative patterns from the time-series in order to characterize the target class locally. These discriminative local patterns, known as shapelets (Ye and Keogh, 2009b), are effective for early classification. An example of such shapelets is shown in Fig. 2.1. The patterns extracted from the two classes of the time-series are discriminative, hence, a new signal can be classified *as soon as* a match between the signal and any of these extracted shapelets is found. In this way, the method is able to justify the prediction of the new signal as red (blue), because the new signal has a pattern that is similar to a pattern observed previously in the red (blue) class. For more details about EDSC, the reader is referred to (Xing et al., 2011).

In cases where signals from different classes are similar to each other, especially in the early phases of the signals, the shapelets extracted from these classes might exhibit similar patterns, which can mislead the prediction. For example, a true alarm signal might match a false alarm shapelet; in this case, the EDSC method would predict the signal as false alarm as soon as the match is found regardless of how *reliable* the match is. In other words, the EDSC method does not provide uncertainty estimates on the match between the signal and shapelet and depends only on the distance measurement (match) for the prediction of label

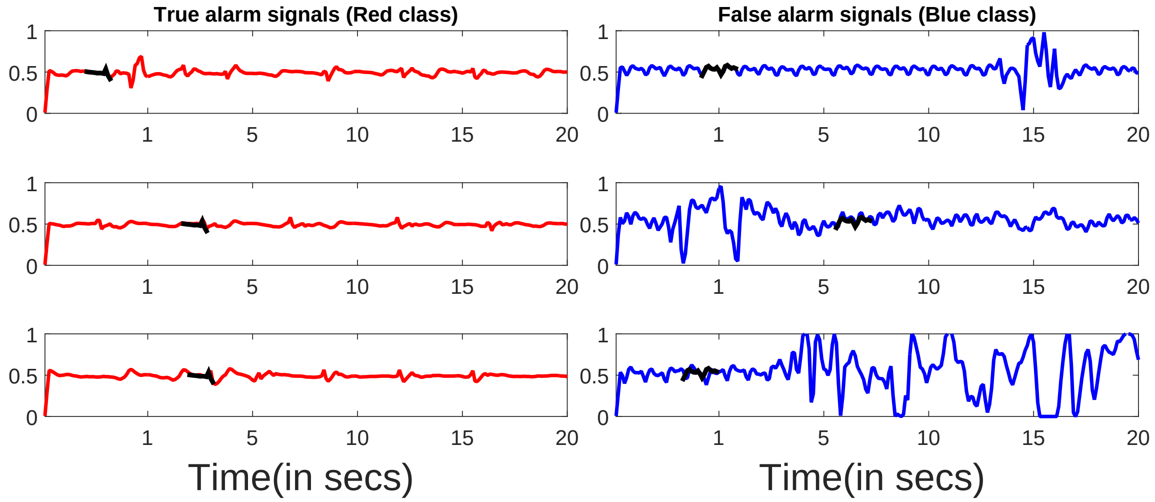


FIGURE 2.1: Shapelets (in black) from true alarm (red) and false alarm (blue) classes.

of the time-series. This drawback was addressed by Ghalwash et al. (Ghalwash et al., 2014) where the EDSC method was extended to produce interpretable early classification of time-series with uncertainty estimates, known as MEDSC-U. The uncertainty for the predicted label was used to decide the class membership of the time-series signal. In this study, the MEDSC-U method is investigated for ECG signal classification and use the uncertainty estimates to decide the alarm class membership for ECG signals. The estimated uncertainties are used to develop a cost-sensitive decision algorithm for early alarm prediction using ECG signals.

2.4 Model Description

The modified early distinctive shapelet classification method for uncertainty estimation (MEDSC-U) (Ghalwash et al., 2014) is described briefly. Given a time-series dataset D, where each time-series example is an ECG signal of 20 seconds prior to the alarm event, each signal is associated with a label (true or false alarm). The task is to correctly classify the ECG signal as early as possible. The MEDSC-U extracts all shapelets of different lengths for early classification. For each shapelet a distance threshold is learned such that the shapelet discriminates between classes. Then, MEDSC-U ranks the shapelets using a utility function that incorporates earliness and accuracy of the shapelet. The shapelets are then pruned by selecting the top performing shapelets that cover the entire dataset and finally the method classifies unknown time-series based on the most *confident* matching shapelets. In Section 2.4.3, how to characterize this method in order to suppress a large fraction of false alarms while keeping near-perfect true alarm detection rates is shown.

2.4.1 Learning Phase

The MEDSC-U method has three steps to extract all discriminative shapelets for early classification.

1. Shapelet Extraction

The shapelet is defined as $S = (s, l, \delta, c)$ where *s* is a time-series subsequence of length *l*, *c* is the alarm label of the shapelet (true or false alarm), and δ is a distance threshold which needs to be learned. The distance threshold is estimated by computing the distances between the subsequence *s* and all time-series in the dataset. To compute the distance between a subsequence *s* of length *l* and a time-series *T* of length *L* (where $l \leqslant L$), a window of length *l* is slide over the time-series *T* to extract all subsequences $\{h_1, h_2, ..., h_{L-l+1}\}$ of length *l*. Then, the distance can be computed as

$$dist(s,T) = \min_{\forall i \in \{1,2,...,L-l+1\}} dist(s, h_i) \tag{2.1}$$

where $dist(s, h_i)$ is the Z-normalized Euclidean distance, which is computed as

$$dist(s,h_i) = \sqrt{\frac{1}{l}\sum_{j=1}^{l}\left(\frac{s_j - \mu_s}{\sigma_s} - \frac{h_{ij} - \mu_{h_i}}{\sigma_{h_i}}\right)^2} \tag{2.2}$$

where μ and σ are the mean and standard deviation of the subsequence. In both (Xing et al., 2011) and (Ghalwash et al., 2014) the distance was computed using Euclidean distance without Z-normalization, however, the Z-normalized Euclidean distance is used due to different variances of the ECG signal examples. The distance threshold δ is computed such that the shapelet discriminates between the two alarm categories. Then, the MEDSC-U method iterates over all time-series in D to extract all subsequences of length *l*, where *l* is the length of the potential shapelet. The method varies *l* between *minL* and *maxL* which are user-defined parameters.

2. Ranking shapelets

MEDSC-U assigns a score to each shapelet that incorporates both the earliness and the accuracy. The earliness defines how early, on average, the shapelet matches the target time-series (the shapelet *S* matches the time-series *T* if $dist(s, T) \leqslant \delta$). Then, the shapelets are sorted in descending order based on their utility scores.

Technically, the earliness between the shapelet $S = (s, l, \delta, c)$ and the time-series *T* of length *L* is defined as

$$EML(S,T) = \min_{\forall i \in \{1,2,\ldots,L-l+1\}} dist(s, h_i) \leqslant \delta \tag{2.3}$$

where h_i are all subsequences of the ECG signal *T* of length *l*. Using earliness the weighted recall of the shapelet is calculated as

$$WeightedRecall(S) = \frac{1}{\|T_{\bar{c}}\|} \sum_{T \in D} \frac{1}{\sqrt[\alpha]{EML(S,T))}} \tag{2.4}$$

where α is a user defined parameter that determines the importance of the earliness, and $\|T_{\bar{c}}\|$ is the number of false alarm ECG signal. Finally, the utility score of the shapelet is defined as

$$Utility(S) = \frac{2 \times Precision(S) \times Weighted\ Recall(S)}{Precision(S) + Weighted\ Recall(S)} \tag{2.5}$$

where *Precision* is the fraction of the matched time-series that are relevant (true alarm time-series) and is computed as

$$Precision(S) = \frac{\|\{d_i \leqslant \delta \wedge class(T_i) = c\}\|}{\|\{d_i \leqslant \delta\}\|} \tag{2.6}$$

where $d_i = dist(s, T_i)$ and $class(T_i)$ is the class of the i^{th} time-series T_i

3. Pruning Phase

The process begins from the highest ranked shapelet S. The MEDSC-U method removes all time-series from the dataset that are covered by the shapelet S. This shapelet is stored along with all other shapelets that have the same score as S (equal-performance shapelets as S). Then, the next ranked shapelet is considered. If the shapelet covers any of the remaining time-series, the shapelet and all other equal-performance shapelets are added to the extracted list and all covered time-series are removed. The method iteratively does so until all time-series in the dataset are covered.

2.4.2 Testing Phase

When an ECG signal T with unknown label (true or false alarm) is encountered, the distance between the *observed* signal and all extracted discriminative shapelets is computed. When the shapelet $S = (s, l, \delta, c)$ matches T (i.e. the distance $dist(s, T)$ between T and S is less than or equal to δ) then T is classified as class c. Since ECG signals from bedside monitors are often contaminated with artifacts and noise which cause false alarms. The distance between T and S contains uncertainty in itself. To account for that uncertainty, the distance is defined as a random variable d

$$d = dist(s, T) + \varepsilon \tag{2.7}$$

where ε is some random variable with 0 mean and standard deviation equal to σ.

If the shapelet S matches T, the confidence C_S^c of classifying T as class c can be estimated by computing two components: 1) confidence in the fact that d is less than a threshold δ and 2) confidence in the ability of shapelet S to accurately classify time-series T. Therefore, C_S^c is defined as

$$C_S^c = C_S(d < \delta | S\ matches\ T) \times C_S(class(T) = c | S\ matches\ T)$$

The first component is defined as

$$C_S(d < \delta | S\ matches\ T) = \frac{(\delta - dist(s,T))^2}{\sigma^2 + (\delta - dist(s,T))^2} \tag{2.8}$$

The closer $dist(s,T)$ is to δ, the lower the confidence is. Also, larger σ means lower confidence. More details about the derivation of Eq. 2.8 can be found in (Ghalwash et al., 2014). The second component is computed as

$$C_S(class(T) = c | S\ matches\ T) = Precision(S) \tag{2.9}$$

where *Precision* is the fraction of the matched time-series that are from class c (Ghalwash et al., 2014). Thus the lower bound of the class confidence estimate of the prediction C_S^c is calculated as

$$C_S^c \geqslant \frac{(\delta - dist(s,T))^2}{\sigma^2 + (\delta - dist(s,T))^2} \times Precision(S) \tag{2.10}$$

Since both terms in this product take value between 0 and 1, the highest value of the C_S^c is 1.

Eq. 2.10 computes the confidence of predicting the time-series T as class c using the shapelet S. So, for any time-series T, the distance $dist(s,T)$ between the time-series and the shapelet is computed. If the distance is less than or equal to the threshold, then the confidence C_S^c is computed using Eq. 2.10. If the distance is greater than the threshold, the confidence is not computed, Hence, the confidence is computed only when the shapelet matches the time-series.

When multiple shapelets match T over time, the overall confidence of the prediction increases as more evidences are gathered for the particular time-series. For more details regarding computing the class confidence when multiple shapelets match, the readers are encouraged to read (Ghalwash et al., 2014).

2.4.3 One-Sided MEDSC-U (1-MEDSCU)

Next the method of adapting MEDSC-U for the task of suppressing false alarm while keeping high true alarm detection rates is described. Since missing true alarm could lead to fatal consequences and risking patients' lives, the naive method is to predict every alarm as a true alarm. In this case, the true alarm detection (sensitivity) is 100% but false alarm suppression (specificity) is 0.

To ensure that no true alarms is missed, a cost-sensitive alarm detection is provided by comparing the computed C_S^c to a predefined confidence threshold value. In particular, a confidence threshold is set for predicting true alarm as very low and for false alarm as very high (99%). Therefore, when a true alarm shapelet (shapelet extracted from true alarm signals) matches the time-series, the signal is classified immediately as a true alarm. On the other hand, when a false alarm shapelet (shapelet extracted from false alarm signals) matches the time-series, the estimated confidence at that particular time point is checked. If the estimated confidence is less than the predefined confidence threshold (no *strong* evidence yet that the signal is a false alarm), the prediction task is delayed and larger signal is looked is taken into consideration in the hope that the confidence estimate will increase with access to more data. If at the end of the time-series the conditions failed to satisfy (no confident true or false alarm shapelets match so far), the ECG signal is classified as a true alarm.

Setting high confidence threshold for false alarm prediction ensures that a signal can be predicted as false alarm *only* if the confidence in the proposed model's prediction is more than 99%, thus ensuring high true alarm detection rates. This approach in decision making ensures no true alarm is missed. On the other hand, the signal is predicted as a true alarm as soon as a match is found so that an early alert for every true alarm is provided.

This approach could be viewed as a hybrid approach between MEDSC-U and EDSC methods, where it utilizes high confidence level for predicting false alarm and predicts a true alarm as soon as a match is found. this approach is termed as One Sided MEDSC-U (1-MEDSCU).

2.5 Data description and Pre-processing

Two different critical alarm datasets were extracted from PhysioNet's MIMIC II version 3 repository (Saeed et al., 2011) (Goldberger et al., e 13). The database is a multiparameter ICU repository containing patient records of up to eight signals from bedside monitors in ICUs. The signals are sampled at 125 Hz. The extracted datasets contains the time stamps and human-annotated true and false asystole and ventricular tachycardia alarms. A subset of patient's records was extracted which contained only signal from lead ECG II, because it was identified as the sensor which contained the least number of missing values across the patients. For each alarm a 20-second window prior to the alarm event was extracted similar to (Salas-Boni et al., 2014). Few alarm events contained missing values, that was ignored in this study. Finally, 261 asystol (ASYS) alarms and 629 ventricular tachycardia (VTACH) alarms was selected. Details about distribution of true and false alarms in the individual datasets are explained in Table 2.3.

The raw signals extracted from MIMIC II was very noisy with high frequency signal components. In order to obtain a smooth signal, the ECG signal was passed through a low pass filter to remove the white noise. A 20-second analysis window prior to the alarm event was considered in the proposed algorithm. However, each 20-second ECG signal contained 2500 data points in the time-series, which increased the computational cost in

Table 2.3: Dataset description.

Dataset	Total alarms	True Alarms (%)	False Alarms (%)
ASYS	261	40 (15.3%)	221 (84.7%)
VTACH	629	227 (36.09%)	402 (63.91%)

the proposed pattern extraction algorithm. Thus, each ECG signal was down-sampled from 125 Hz to 12.5 Hz, resulting in 250 temporal points in each signal.

2.6 Experimental setup

Assuming that the number of true alarms is N, the true alarm dataset was partitioned into four distinct partitions, hence, each partition had $N/4$ true alarms. For each partition, $N/4$ false alarms were randomly selected from the false alarm dataset to ensure balanced training data. To train the proposed method (and the baseline methods) using the training data (of size $N/2$) and test them on the remaining examples. In addition, the entire process was repeated 20 times (each repetition had 4 distinct partitions on true alarm) which resulted in 80 different combinations of training data.

4 evaluation measures was used: True alarm detection (TAD) rate, which is sensitivity; False alarm suppression (FAS) rate, which is specificity; and Earliness, which is the fraction of the time points used for classification. However, since missing true alarm (positive class) is more severe than missing false alarm (negative class), different errors incur different weights. The balanced accuracy (the average between sensitivity and specificity) considers similar weights for different errors. To account for this, a weighted accuracy metric is proposed where the false negative is penalized more than the false positive by $1+\beta^2$. Higher β penalizes false negatives more than false positives. Therefore, the weighted balanced accuracy ($WAcc$) is computed as:

$$WAcc = (WSens + Spec)/2 \tag{2.11}$$

where

$$WSens = TP/(TP + (1+\beta^2)FN)$$

$$Spec = TN/(TN + FP)$$

where TP, TN, FP, FN is the number of true positives, true negatives, false positives, and false negatives, respectively.

2.7 Baseline methods

The proposed method is compared to three baseline models.

1. BeharRaw: A state-of-the-art false alarm suppression method (Behar et al., 2013) was applied on the raw ECG signals.

2. BeharFiltered: The same state-of-the-art method(Behar et al., 2013) was applied on the filtered ECG signals.

3. EDSC (Xing et al., 2011) with the Z-normalized version of distance measuring (Eq. 2.2).

The original EDSC method resulted in 0 sensitivity, thus it was not include as a baseline method.

2.8 Results

2.8.1 Accuracy performance

The evaluation of each method is shown in Table 2.4. Clearly, 1-MEDSCU method has near optimal TAD rate, while all other methods have much less TAD rate. For example, on ASYS dataset, BeharFiltered has comparable TAD rate (92.37%) to 1-MEDSCU method (99.12%), however, it has lower FAS rate (18.97%) compared to 1-MEDSCU (34.29%). EDSC has better FAS rate (74.16%) than the proposed model but on the cost of TAD rate (83.62%). This shows that the proposed method has moderate FAS rates while keeping high TAD rate, which is an extremely challenging task. The same conclusion applies on VTACH. However, the desired level of FAS can be obtained by adjusting the confidence threshold of the method, which will reduce TAD rate but will still be comparable to other methods. This is explained in the next section.

In addition to TAD and FAS, it is clear that 1-MEDSCU method has better weighted accuracy $WAcc$ than all other methods. There is a statistically significant difference

Table 2.4: Evaluation of the models in terms of true alrm detection rate (TAD), false alarm suppression rate (FAS), earliness (100 - earliness) and weighted accuracy (WAcc). Larger value has better performance. Pvlaue is computed between 1-MEDSCU and the best baseline method on the corresponding evaluation measure.

		Behar (Raw)	Behar (Filtered)	EDSC	1-MEDSCU	pvalue
2 ASYS	TAD	84.62±11.46	92.37±11.27	83.62±15.19	99.12± 3.25	
	FAS	35.03±7.64	18.97±7.24	74.16±9.41	34.29±12.36	
	100-Earliness	0	0	62.8±6.27	38.39±9.05	
	$WAcc$ ($\beta = 2$)	47.11±11.13	49.16±10.84	**66.12±11.55**	65.68±6.32	0.74
	$WAcc$ ($\beta = 3$)	41.9±13.7	45.96±14.33	59.75±13.39	**65.05±7.55**	1.20e-03
2 VTACH	TAD	86.22±8.69	52.60±25.27	64.78±23.16	**95.67±8.81**	
	FAS	31.18±5.7	51.07±24.75	65.07±14.77	20.32±13.43	
	100-Earliness	0	0	59.9±11.71	39.96±9.34	
	$WAcc$ ($\beta = 2$)	44.49±3.11	37.20±5.99	48.65±2.47	**52.85±5.52**	4.72e-09
	$WAcc$ ($\beta = 3$)	36.34±2.91	32.80±8.32	42.58±3.39	**48.56±7.41**	3.13e-09

(pvalue is shown in the last column of Table 2.4) between the proposed method and all other methods using significance level 0.05, except for EDSC on ASYS at $\beta = 2$.

2.8.2 Controlling False Alarm Suppression Rate

1-MEDSCU has advantage over other methods in controlling the balance between TAD and FAS. In other words, the false alarm confidence threshold used in 1-MEDSCU controls the sensitivity of the model to predict true and false alarm. When a test ECG signal matches a false alarm shapelet (blue shapelet as in Fig. 2.1), the method computes the confidence of the match. If the estimated confidence is greater than the false alarm confidence threshold, then 1-MEDSCU predicts the signal as a false alarm. Increasing the false alarm confidence threshold *guarantees* that no true alarm is incorrectly predicted as a false alarm but at the same time decreases the false alarm suppression rate. In the previous results, a 99% false alarm confidence threshold was used to ensure near-optimal TAD rates. By varying the confidence level FAS rate comparable to other method can be obtained but still with higher (but not near-optimal) TAD rates. The results of varying the false alarm confidence threshold is shown in Fig. 2.2.

The blue dotted (red dashed) line represents the varying FAS (TAD) rates for different false alarm confidence thresholds (x-axis), respectively. The blue marks (star, circle, and diamond) indicate the FAS rates, while the red marks show the TAD rates achieved by the three baseline models. It is clear that, the proposed model can achieve similar FAS rates as the baseline methods with comparable or even higher TAD rate. For example, in order to achieve FAS rate similar to EDSC (blue star) a TAD rate of 83 (the vertical line that touches the blue star, touches the red dashed line at 83%) can be achieved by setting false alarm confidence threshold to 4.9%. So, comparable TAD rates to EDSC (red star) is achieved. Comparing to BeharRaw, it has 35% FAS (blue diamond) and 85% TAD (red circle), while 1-MEDSCU can achieve 99% TAD at 35% FAS, significantly outperforming BeharRaw.

Therefore, by varying the confidence threshold, one can achieve the desired level of FAS with comparable or even better TAD rates.

2.8.3 Earliness

The results of earliness of the methods are shown in Table 2.4. 1-MEDSCU not only provide accurate results (as shown in the previous sections) but also provide these results early. So, the prediction takes place even before the actual alarm alerts, whereas all other methods, except EDSC, provide results at the time when the alarm happens. The prediction of the proposed method is provided, on average, using around 60% of the time points (8 seconds before the actual alarm) at a false alarm confidence threshold of 99%. Although, it is evident that EDSC has better earliness performance than 1-MEDSCU, the proposed method outperforms EDSC in terms of TAD as in Table 2.4. By varying the false alarm confidence threshold, the earliness of 1-MEDSCU improves as shown in Fig. 2.3. At 4.9% confidence threshold, the predictions of the proposed method were provided using only 40% of time-series' length, comparable to EDSC.

From Fig. 2.2 and 2.3, it can be concluded that by lowering the false alarm confidence threshold one can obtain earlier predictions and higher FAS rates but at the cost of reducing

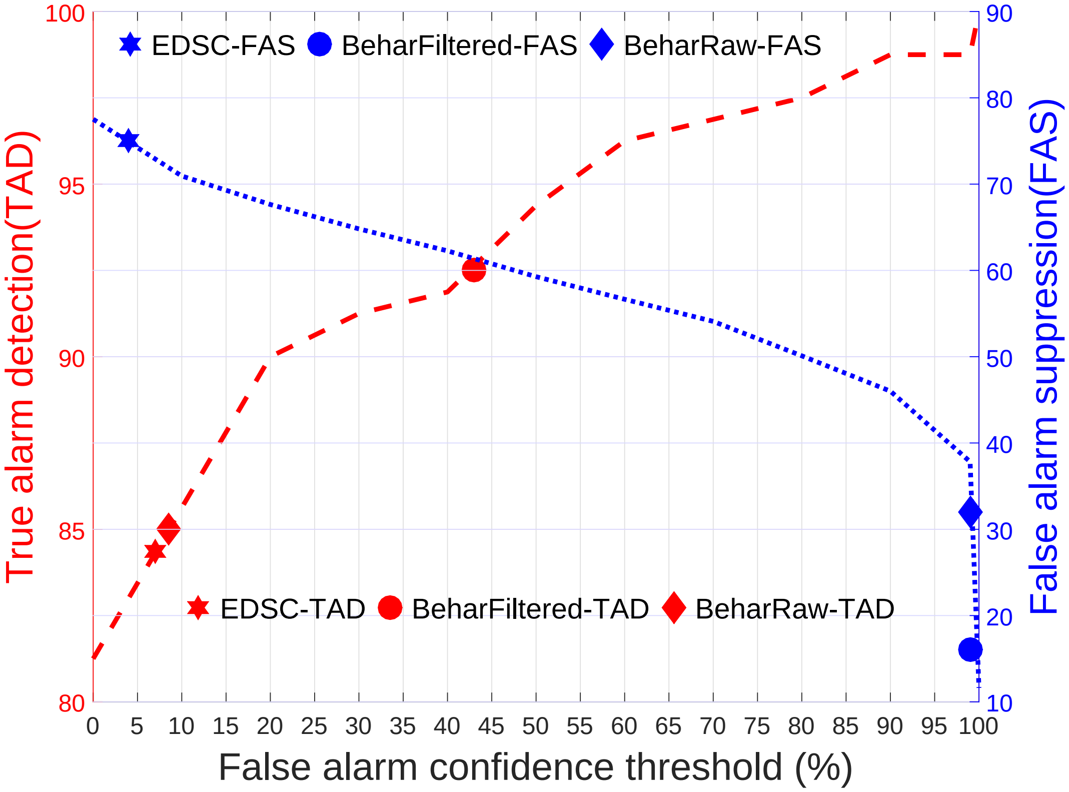


FIGURE 2.2: Varying false alarm confidence threshold for ASYS. The red line shows increasing true alarm detection with increasing false alarm confidence threshold. The blue line show decreasing false alarm suppression with increasing false alarm confidence threshold. The red and blue marks indicate TAD and FAS respectively achieved by the baseline methods.

TAD rates. Therefore, a proper trade-off has to decided by hospital administrators between earliness, FAS and TAD.

2.8.4 Interpretability: Case Study

An example is presented to show the effectiveness of the proposed interpretable method that utilizes the confidence levels to produce more accurate results. In Fig. 2.4, a true alarm signal matches a false alarm shapelet (solid blue segment) with confidence 1% at time point 4 (so 16 seconds before the alert). EDSC would classify that example at that time as false alarm. However, 1-MEDSCU does not classify the signal at that time because the confidence is less than the false alarm confidence threshold (99%), therefore, delays

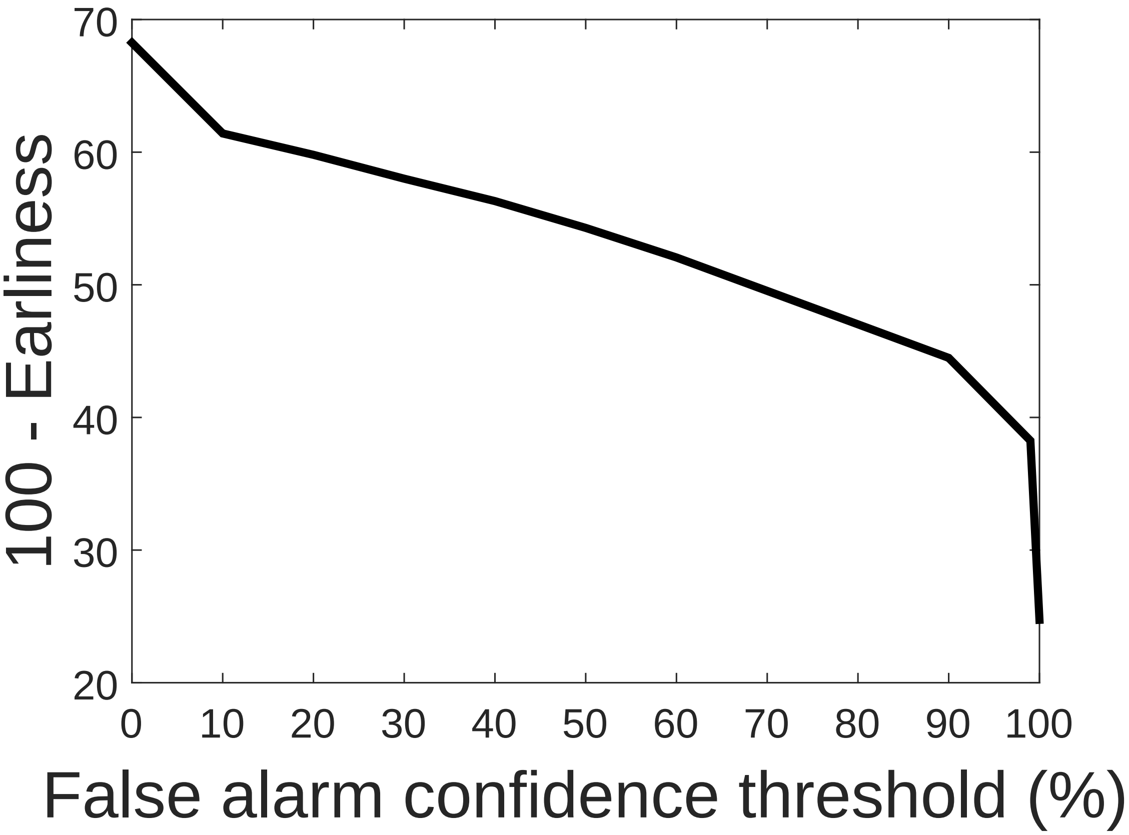


FIGURE 2.3: Trend of Earliness with varying false alarm confidence threshold (100 - Earliness) on ASYS. Larger the value, the earlier the prediction.

the decision. At time 4.9 second, another false alarm shapelet (dotted blue) matches the signal resulting in confidence 8%. 1-MEDSCU continues until time 16 where the signal matches a true alarm shapelet (red shapelet). The method immediately classifies the signal correctly as a true alarm, because the method does not require confidence to predict the signal as a true alarm. It is clear that the signal can be mistakenly classified as a false alarm because two evidences (two shapelets) were found in the early phases of the signal. However, since the evidences are not strong enough the method continues until either a very strong evidence to classify as a false alarm is found or any evidence to classify it as a true alarm is found, thus ensuring high TAD rates.

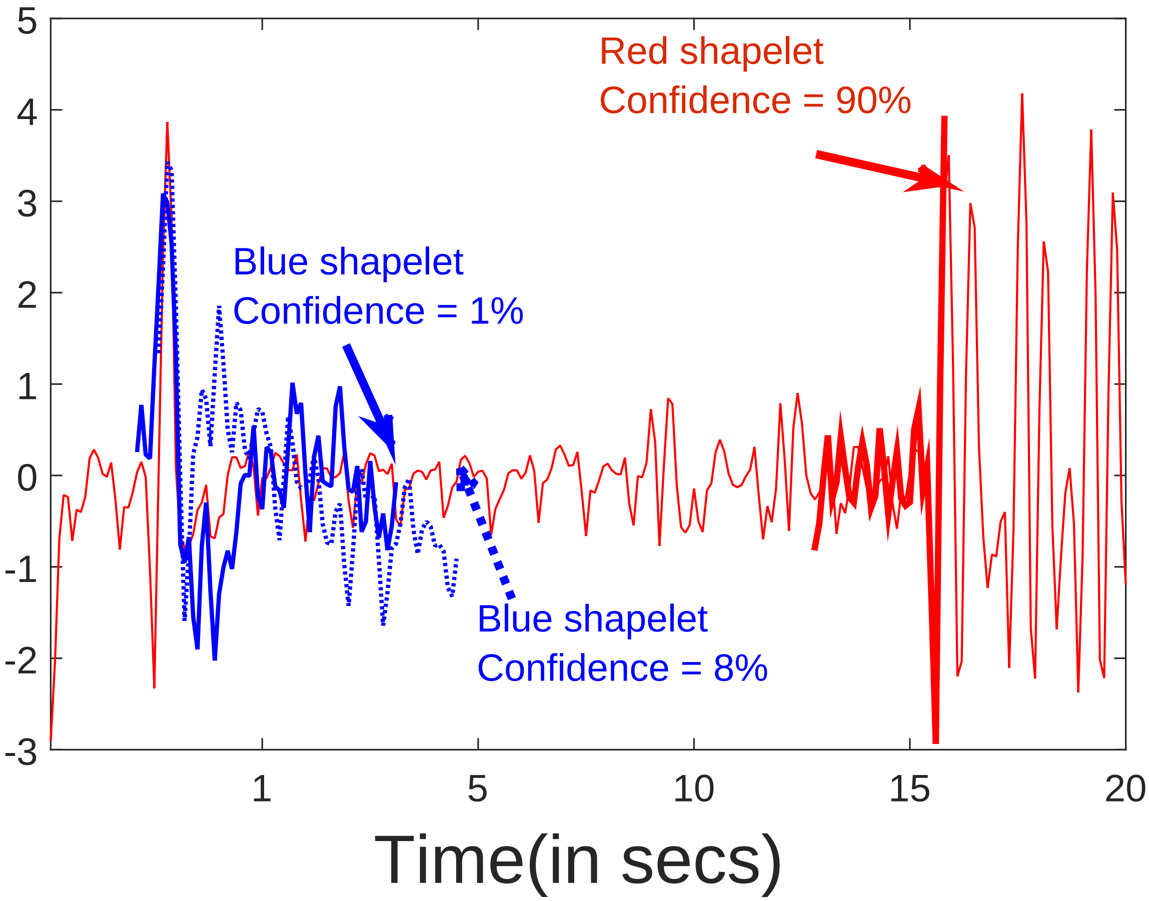


FIGURE 2.4: True alarm example wrongly classified as false alarm by EDSC at time 4, however, correctly classified as true alarm by 1-MEDSCU at time 16.

2.9 Conclusion

In this chapter, the problem of suppressing high cardiac false alarms using univariate ECG signals is addressed. The objective of this study was to reduce false alarms as much as possible without compromising TAD performance. This objective was achieved in the proposed (1-MEDSCU) model by keeping high confidence threshold for false alarm predictions to ensure high TAD. A moderate percentage of FAS was achieved while keeping high rate of early TAD predictions. The proposed early alarm detection approach had outperformed the state-of-the-art methods on both datasets in terms of weighted accuracy. In addition, it was shown how one can control the FAS rate at the cost of

TAD rate, nevertheless, the method achieved higher suppression rate than other methods keeping comparable TAD rate. In addition, the method provides not only accurate results but also provides interpretable results early.

CHAPTER 3

HIGHLY IMBALANCED TIME-SERIES CLASSIFICATION

3.1 Introduction

One of the key sources of performance degradation in the field of time-series classification is the class imbalance problem (López et al., 2013) the minority class (positive class) is outnumbered by abundant negative class instances. Models built using standard classification algorithms on such imbalanced datasets, which generally have minimum classification error as a criterion for classifier design often, are biased towards the majority class; and therefore, have higher misclassification error for the minority class examples. Moreover, in real-world scenarios such as object detection, medical diagnosis etc., the positive class is usually the more important class and false negatives are always costlier than false positives. Traditional 0 - 1 loss function classifiers fail to differentiate between these two types of errors and final outcomes are naturally biased towards the abundant negative class. Thus, a cost-sensitive classifier is preferred when dealing with datasets where examples from different classes carry different misclassification costs.

Recently, in the realm of time-series classification, a novel framework was proposed known as Learning Time-series Shapelets (LTS) (Grabocka et al., 2014) to directly learn generalized short time-series subsequences known as shapelets ((Ye and Keogh, 2009a)) along with weights of a classifier hyperplane to differentiate temporal instances in a binary classification framework. Shapelets are local discriminative patterns (or subsequences) that can be used to characterize the target class, for determining the time-series class membership. Shapelets have been proven to have high predictive powers as they provide local variation information within the time-series as well as high interpretability of predictions due to easier visualizations. LTS formulates an optimization problem where a cost-insensitive 0-1 logistic loss function is minimized in order to learn generalized shapelets. The minimum Euclidean distances of the learned shapelets to the time-series can be used to linearly separate the time-series examples from different classes.

However, LTS uses cost-insensitive loss function that treats false positive and false negative errors equally, which limits its applicability on balanced datasets. In this chapter, a cost-sensitive time-series classification framework (henceforth known as CS-LTS) is proposed by extending the LTS model. A cost-sensitive logistic loss function is minimized to enhance the modeling capability of LTS. The cost-sensitive logistic loss function uses variable misclassification costs for false positive and false negative errors. Generally, these misclassification cost values are available from the cost matrix provided by domain experts which is often a cumbersome procedure. Instead of using fixed cost parameters, the proposed method *learns* the variable misclassification costs from the training data via a constrained optimization problem.

3.2 Contribution

The main contributions of this chapter are summarized as the following.

- The proposed method learns the misclassification costs from the training data thus nullifying the need for predetermination of cost values for misclassification errors. To the best of my knowledge, the proposed model is the first algorithmic approach to solve

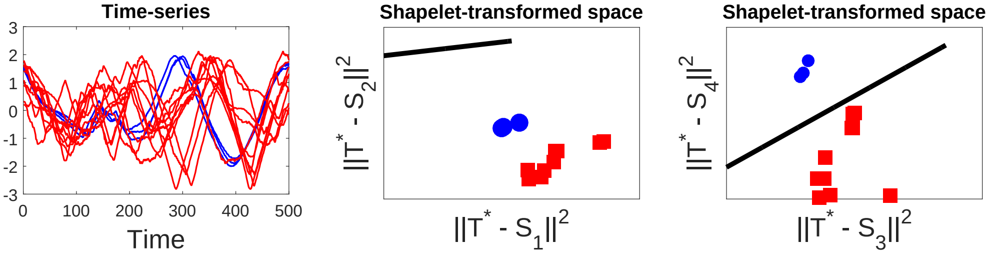


FIGURE 3.1: An illustration of the proposed CS-LTS model (right) compared to LTS (middle) using 2 shapelets learned on an imbalanced version of BirdChicken dataset (left)

highly imbalanced time-series classification problem.

- A constrained optimization problem is proposed which jointly learns shapelets (highly interpretable patterns), their weights, and most importantly misclassification costs, while other cost-sensitive approaches mainly consider misclassification costs are given a priori.
- The effectiveness of the method is demonstrated on life-threatening cardiac arrhythmia dataset from Physionets MIMIC II repository showing improved true alarm detection rates over the current state-of-the-art method for false alarm suppression.
- Finally, the method is evaluated extensively on 34 real-world time-series datasets with varied degree of imbalances and compared to a large set of baseline methods previously proposed in the realm of imbalance time-series classification problems.

In Fig. 3.1, all time-series examples are shown for the blue and red classes. The blue class has only 3 time-series, while the red class has 10 time-series. Since LTS does not handle imbalance dataset, the learned hyperplane is very biased. This is clear from the middle image in Fig. 3.1 that shows the distance between the two learned shapelets using LTS and the training time-series. CS-LTS learns a hyperplane that is aware about the imbalance in the data, as shown in rightmost image of Fig. 3.1.

3.3 Related Work

Time-series classification via shapelets.

In the field of time-series classification, the concept of shapelets have received a lot of attention (Grabocka et al., 2014; Hou et al., 2016; Zhang et al., 2016; Ghalwash et al., 2014; Ye and Keogh, 2009a). Shapelets are local discriminative patterns (or subsequences) that characterize the target class and maximally discriminate instances of time-series from various classes. Discovering the most discriminative subsequences is crucial for the success of time-series classification using shapelets. The primary approach, based on search-based techniques, proposed by Ye et al. ((Ye and Keogh, 2009a)), exhaustively search for all possible subsequences and a decision tree was constructed based on information gain criterion. The information gain accuracy was ranked based on the minimum distance of the candidate subsequences to the entire time-series training set. Hills et al. ((Hills et al., 2014)) perceived this minimum distance of the set of shapelets to a time-series dataset as a data transformation to a shapelet-transformed space where standard classifiers could be used to achieve high classification accuracy using the shapelet-transformed data as predictors. Recently, Grabocka et al. ((Grabocka et al., 2014)) proposed a novel framework known as Learning Time-series Shapelets (LTS) to jointly learn generalized shapelets along with weights of a logistic regression model using the minimum Euclidean distances of shapelets to time-series dataset as predictors. The method discovered optimal shapelets and reported statistically significant improvements in accuracy compared to other shapelet-based time-series classification models. However, a major drawback is low true positive rate in case of highly imbalanced time-series datasets. The logistic loss used in the LTS framework is a cost-insensitive loss function which treats false positive and false negative misclassifications errors equally. Classification models built using such loss functions suffer from the class imbalance problem.

Cost-sensitive classification.

Classification techniques for handling imbalanced data-sets can broadly be divided into two kinds of approaches, data-level approaches (Cao et al., 2011; He et al., 2008; Cao et al., 2013; Chawla et al., 2002; Han et al., 2005; Liu et al., 2009; Cao et al., 2014) and algorithmic-level (Sun et al., 2007) approaches. Data-level methods are sampling techniques that act as a pre-processing steps prior to the learning algorithm to balance the imbalanced datasets either through oversampling of the minority class or under sampling of the majority class or combination of both. Algorithmic-level approaches directly manipulate the learning algorithm by incorporating a predefined misclassification cost for each class to the loss function. These methods have reported excellent performance with good theoretical guarantees (He and Garcia, 2009); however, predetermination of optimal class misclassification cost or data-space weighting is required which can vary on a case-by-case basis among different datasets and also require domain expertise.

In this study, an algorithmic approach is followed to directly manipulate the learning procedure by minimizing a cost-sensitive logistic loss function. An additive asymmetric learning function is fitted to the training data. In addition to learning the shapelets and weight parameters of the classification hyperplane, the cost parameters are also estimated from the training data. A constrained optimization problem is formulated that is optimized to jointly learn shapelets, weights of the classification hyperplane and misclassification cost parameters nullifying the need for predetermination of cost values for misclassification errors.

3.4 Method Preliminaries

A binary class time-series dataset composed of I training examples denoted as $\mathbf{T} \in \mathbb{R}^{I \times Q}$ is considered where, each T_i $(1 \leqslant i \leqslant I)$ is of length Q and the label for each time-series instance is a nominal variable $Y \in \{0, 1\}^I$. Candidate shapelets are segments of length L from a time-series starting from $j-$th time point inside the i^{th} time-series. The objective is to learn k shapelets **S**, each of length L, that are most discriminative in order to characterize the target class. The shapelets are denoted as $S \in \mathbb{R}^{K \times L}$.

The minimum distance $M_{i,k}$ between the i^{th} series T_i and the k^{th} shapelet S_k is the distance between the segment and time-series. This is defined as

$$M_{i,k} = \min_{j=1,\ldots,J} \frac{1}{L} \sum_{l=1}^{L} (T_{i,j+l-1} - S_{k,l})^2 \tag{3.1}$$

Given a set of I time-series training examples and K shapelets, a shapelet-transformed matrix (Hills et al., 2014) **M** $\in \mathbb{R}^{I \times K}$ can be constructed which is composed of minimum distances $M_{i,k}$ between the i^{th} series T_i and the k^{th} shapelet S_k. The minimum distance M matrix is a representation in the shapelet transformed space and acts as predictors for each target time-series. However, the function in Eq. 1 is not continuous and thus non-differentiable. Grabocka et al. (Grabocka et al., 2014) defined a soft-minimum function (shown in Eq. 3.2), which is an approximation for $M_{i,k}$.

$$M_{i,k} \approx \hat{M}_{i,k} = \frac{\sum_{j=1}^{J} D_{i,k,j} \exp(\alpha D_{i,k,j})}{\sum_{\bar{j}=1}^{J} \exp(\alpha D_{i,k,\bar{j}})} \tag{3.2}$$

where $D_{i,k,j}$ is defined as the distance between the j^{th} segment of series i and the k^{th} shapelet given by the formula

$$D_{i,k,j} = \frac{1}{L} \sum_{l=1}^{L} (T_{i,j+l-1} - S_{k,l})^2 \tag{3.3}$$

3.5 Model Description

A linear learning model (shown in Eq. 3.4) was proposed by (Grabocka et al., 2014) using the minimum distances M as predictors in the transformed shapelet space.

$$\hat{Y}_i = W_0 + \sum_{k=1}^{K} M_{i,k} W_k \quad \forall i \in \{1, ..., I\} \tag{3.4}$$

The learning function (Eq. 3.4) is extended by incorporating C_{FN} and C_{FP} for false negative and false positive misclassifications cost respectively. The new asymmetric learning model is defined as Eq. 3.5.

$$Z_i = \frac{1}{C_{FN} + C_{FP}} ln \frac{\sigma(\hat{Y}) C_{FN}}{1 - \sigma(\hat{Y}) C_{FP}} = \frac{1}{C_{FN} + C_{FP}} (\hat{Y} + ln \frac{C_{FN}}{C_{FP}}) \tag{3.5}$$

$\sigma()$ is the logistic function and $\sigma(\hat{Y})$ represents the posterior probability of $P(Y = 1|X)$.

Additionally, a cost-sensitive loss function (Eq. 3.6) is proposed which is a differential cost-weighted logistic loss between the actual targets Y and the estimated targets Z.

$$\mathcal{L}(Y, Z) = -Y ln \sigma(C_{FN} Z) - (1 - Y) ln(1 - \sigma(C_{FP} Z)) \tag{3.6}$$

A regularized cost-sensitive logistic loss function defined by Eq. 5.6 is the regularized objective function denoted by $\mathcal{F}$.

$$\underset{S,W,C}{\operatorname{argmin}} \mathcal{F}(S, W, C) = \underset{S,W,C}{\operatorname{argmin}} \sum_{i=1}^{I} \mathcal{L}(Y_i, Z_i) + \lambda_W \|W\|^2 \tag{3.7}$$

where $C \in \{C_{FN}, C_{FP}\}$. The problem is formulated as a constrained optimization problem since the misclassification costs should always be positive. The misclassification cost denotes the loss incurred when a wrong prediction occurs. The constraints ensure both costs are positive and also the fact that cost of false negative is at least θ times greater than cost of false positive. These conditions ensure the loss function to be penalized more in the event of an error in the positive class than an error in the negative class.

$$\begin{aligned} &\operatorname*{argmin}_{S,W,C} \mathcal{F}(S,W,C) \\ \text{subject to } &C_{FN} > 0,\ C_{FP} > 0 \\ &C_{FN} > \theta C_{FP} \end{aligned} \tag{3.8}$$

Similar to (Grabocka et al., 2014), a Stochastic gradient descent (henceforth SGD) approach is adopted to solve the optimization problem. The SGD algorithm optimizes the parameters to minimize the loss function by updating through per instance of the training data. Thus, the per-instance decomposed objective function $\mathcal{F}_i$ (denoted by Eq. 3.9) shows the division of Eq. 5.6 into per-instance losses for each time-series.

$$\mathcal{F}_i = \mathcal{L}(Y_i, Z_i) + \frac{\lambda_W}{I} \sum_{k=1}^{K} W_k^2 \tag{3.9}$$

The objective of the learning algorithm is to learn the optimal shapelet S_k, the weights W for the hyperplane and the misclassification costs C which minimizes the loss function (Eq. 5.6).

The SGD algorithm requires definitions of gradients of the objective function with respect to shapelets, hyperplane weights and misclassification costs. Eq. 5.9 shows the point gradient of objective function for the i^{th} time-series with respect to shapelet S_k.

$$\frac{\partial \mathcal{F}_i}{\partial S_{k,l}} = \frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial Z_i} \frac{\partial Z_i}{\partial \hat{Y}_i} \frac{\partial \hat{Y}_i}{\partial \hat{M}_{i,k}} \sum_{j=1}^{J} \frac{\partial \hat{M}_{i,k}}{\partial D_{i,k,j}} \frac{\partial D_{i,k,j}}{\partial S_{k,l}} \tag{3.10}$$

Furthermore, the gradient of the cost-sensitive loss function with respect to the learning function Z_i is defined in Eq. 3.11. Also the gradient of the cost-sensitive learning function with respect to the estimated target variable $\hat{Y}_i$ is shown in Eq. 3.12

$$\frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial Z_i} = (1 - Y_i)\sigma(C_{FP} Z_i) C_{FP} - Y_i(1 - \sigma(C_{FN} Z_i)) C_{FN} \tag{3.11}$$

$$\frac{\partial Z_i}{\partial \hat{Y}_i} = \frac{1}{C_{FN} + C_{FP}} \tag{3.12}$$

Eq. 3.13 shows the gradient of the estimated target variable with respect to the minimum distance. The gradient of the over all minimum distance with respect to the segment distance and the gradient of the segment distance with respect to a shapelet point is defined by Eq. 3.14 and Eq. 3.15 respectively.

$$\frac{\partial \hat{Y}_i}{\partial \hat{M}_{i,k}} = W_k \tag{3.13}$$

$$\frac{\partial \hat{M}_{i,k}}{\partial D_{i,k,j}} = \frac{\exp(\alpha D_{i,k,j}(1 + \alpha(D_{i,k,j} - \hat{M}_{i,k}))}{\sum_{\bar{j}=1}^{J} \exp\left(\alpha D_{i,k,\bar{j}}\right)} \tag{3.14}$$

$$\frac{\partial D_{i,k,j}}{\partial S_{k,l}} = \frac{2}{L}(S_{k,l} - T_{i,j+l-1}) \tag{3.15}$$

The hyperplane weights W are learned by minimizing the objective function 5.6 via SGD. The gradients for updating the weights W_k is shown in Eq. 3.16 and Eq. 3.17 shows the gradient for update of the bias term W_0.

$$\frac{\partial \mathcal{F}_i}{\partial W_k} = \frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial Z_i} \frac{\partial Z_i}{\partial \hat{Y}_i} \hat{M}_{i,k} + \frac{2\lambda_W}{I} W_k \tag{3.16}$$

$$\frac{\partial \mathcal{F}_i}{\partial W_0} = \frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial Z_i} \frac{\partial Z_i}{\partial \hat{Y}_i} \tag{3.17}$$

The learning procedure for estimating the misclassification cost values in the proposed framework is a constrained optimization problem since $C_{FN} > 0$, $C_{FP} > 0$ and $C_{FN} > \theta C_{FP}$, where $\theta \in \mathbb{Z}$. However, Stochastic Gradient Descent algorithm can only be applied to solve unconstrained optimization problems. Thus, the constrained optimization is converted into an unconstrained optimization similar to (Radosavljevic et al., 2010) SGD algorithm is applied to solve the optimization problem for learning the optimal misclassification costs.

$$C_{FN} = \theta C_{FP} + \mathcal{D} \tag{3.18}$$

The false negative misclassification cost (C_{FN}) is first written in terms of false positive misclassification cost as shown in Eq. 3.18 and replaced in Eq. 3.6 changing the optimization problem to Eq. 5.12.

$$\begin{aligned} &\underset{S,W,C_{FP},\mathcal{D}}{\text{argmin}}\ \mathcal{F}(S, W, C_{FP}, \mathcal{D}) \\ &\text{subject to } C_{FP} > 0 \end{aligned} \tag{3.19}$$

$\mathcal{D}$ is a regularization term for the misclassification cost. The objective function is then minimized with respect to $\log C_{FP}$ instead of C_{FP}. As a result, the new optimization problem becomes unconstrained. Derivatives of objective function with respect to $\log C_{FP}$ and $\mathcal{D}$ in gradient descent are computed as:

$$\frac{\partial \mathcal{F}_i}{\partial \log c_{FP}} = c_{FP} \frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial c_{FP}} \tag{3.20}$$

$$\frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial c_{FP}} = \frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial Z_i} \frac{\partial Z_i}{\partial c_{FP}} \tag{3.21}$$

$$\frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial \mathcal{D}} = \frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial Z_i} \frac{\partial Z_i}{\partial \mathcal{D}} \tag{3.22}$$

The steps of the proposed cost-sensitive time-series classification method (CS-LTS, henceforth) are shown in Algorithm 1. The pseudocode shows that the procedure updates all K shapelets and the weights W, W_0, false positive cost C_{FP} and parameter $\mathcal{D}$ by a learning rate η.

Algorithm 1 Cost-sensitive learning time-series shapelets

```
procedure CS-LTS
Input: $T \in \mathcal{R}^{I \times Q}$, Number of shapelets $K$, length of a shapelet $L$, Regularization parameter $\lambda_W$, Learning rate $\eta$, maxIter
Initialize: Shapelets $S \in \mathbb{R}^{K \times L}$, classification hyperplane weights $W \in \mathbb{R}^K$, Bias $W_0 \in \mathbb{R}$, Misclassification cost $C_{FP} \in \mathbb{R}$, $\theta \in \mathbb{Z}$, $\mathcal{D} \in \mathbb{R}$
  for iterations = $\mathbb{N}_1^{maxIter}$ do
    for $i = 1, ..., I$ do
      for $k = 1, ..., K$ do
        $W_k^{new} \leftarrow W_k^{old} - \eta \frac{\partial \mathcal{F}_i}{\partial W_k}$
        for $l = 1, ..., L$ do
          $S_{k,l}^{new} \leftarrow S_{k,l}^{old} - \eta \frac{\partial \mathcal{F}_i}{\partial S_{k,l}}$
        end for
      end for
      $W_0^{new} \leftarrow W_0^{old} - \eta \frac{\partial \mathcal{F}_i}{\partial W_0}$
      $\log C_{FP}^{new} \leftarrow \log C_{FP}^{old} - \eta \frac{\partial \mathcal{F}_i}{\partial \log C_{FP}}$
      $\mathcal{D}^{new} \leftarrow \mathcal{D}^{old} - \eta \frac{\partial \mathcal{F}_i}{\partial \mathcal{D}}$
    end for
  end forReturn $S, W, W_0, C_{FP}$
end procedure=0
```

3.6 Experimental Evaluation

In this section, the effectiveness of the proposed method (CS-LTS) is evaluated on different setting represented by different datasets. The objective function in Eq. 5.6 is a non-convex function with respect to parameters and solving it via SGD requires a good initialization of the parameters. The initialization step is very important in this scenario as it influences whether the optimization reaches the region of global minimum.

3.6.1 Model parameter initialization

Shapelets were initialized using K-means centroids of all segments similar to (Grabocka et al., 2014). First the minimum length (L_{min}) of a shapelet is set to 10% of the length of the time-series examples. Then the total number of shapelets was computed as L_{min} multiplied by number of training time-series. The number of shapelets used as input for the optimization function was determined using $K = \log(total\ number\ of\ segments)$.

Three scales $\{L_{min}, 2 \times L_{min}, 3 \times L_{min}\}$ of subsequence lengths were investigated.

The weight parameters W_k and W_0 were initialized randomly around 0. C_{FP} was initially set to 1. The values for θ and initial value of $\mathcal{D}$ were determined through a grid search approach using internal cross-validations over the training data. The values for θ were searched from the set $\{1, 5, 10, 25, 50, 100\}$ and the initial values for $\mathcal{D}$ was chosen from $\{0.001, 0.01, 0.1, 10, 100, 1000\}$. The best parameter value was identified via internal cross-validation on training data. Once the best parameter value was identified, the methods were trained on the entire training set using the best chosen parameters, and the learned model was tested on the test set which was completely separate from the training procedure. The learning rate η was initialized to a small value of 0.01. The $maxIter$ for the optimization was set to 5000 iterations.

3.6.2 Evaluation measures

F_β score for $\beta \in \{1, 2, 3\}$ is reported since this is a commonly used performance metric for imbalanced learning. These are simple functions of the precision and recall. The traditional F-score or F_1 score is the harmonic mean of precision and recall that is considered a balanced measure between precision and recall. For $\beta > 1$ the evaluation metric rewards higher true positive rates. The sensitivity and specificity evaluation metrics is also considered, as the objective is to achieve lower false negative with minimum increase in false positive rates.

3.7 Results

3.7.1 Cost Sensitive Cardiac Arrhythmia Alarms Detection

In this set of experiments, the effectiveness of the proposed method is demonstrated on two cost-sensitive applications from PhysioNet's MIMIC II version 3 repository (Goldberger et al., e 13; Saeed et al., 2011). The objective is to detect true alarms while suppressing false alarms, where missing true alarms (positive class) is more severe than missing false alarms (negative class), since missing true alarm could lead to serious consequences and risk patients' lives.

The database is a multi-parameter ICU repository containing patients' records of up to eight signals from bedside monitors in Intensive Care Units (ICU). The extracted datasets contain human-annotated true and false cardiac arrhythmia alarms. A subset of patients' records was extracted that contained signal from lead ECG II, because it was identified as the sensor that contained the least number of missing values across the patients. For each alarm event, a 20-second window prior to the alarm event was extracted similar to (Roychoudhury et al., 2015).

The dataset is partitioned into four distinct cross-validation datasets, where the model is trained on 3 folds and tested on the fourth one. In addition to the cross validation experiment, the entire process of cross-validation is repeated for 10 independent trials (each trial has 4 distinct partitions on true alarm instances) which results in 40 different combination of training data. The mean and standard deviation of the evaluation metrics is then reported.

The two datasets selected are VTACH and CHALLENGE. VTACH consists of true and false Ventricular Tachycardia alarms from the ICU patients. CHALLENGE dataset is a mixture of different true and false arrhythmia alarms. The alarms categories are Asystole, Extreme Bradycardia, Extreme Tachycardia, Ventricular Tachycardia and Ventricular Flutter/Fibrillation. This dataset was presented at a competition in 2015 organized by PhysioNet to encourage the development of algorithms to reduce the incidence of false alarms in the Intensive Care Unit (ICU).

Achieving high true alarm detection rate (TAD) or high sensitivity is important when suppressing high false alarm rates from bedside monitors in ICU. High false alarm rates cause desensitization among care providers, thus risking patients' lives (Drew et al., 2014). The objective of the prediction task is to provide high false alarms suppression (FAS) rates (achieve high specificity) while keeping TAD (sensitivity) high. In the two datasets,

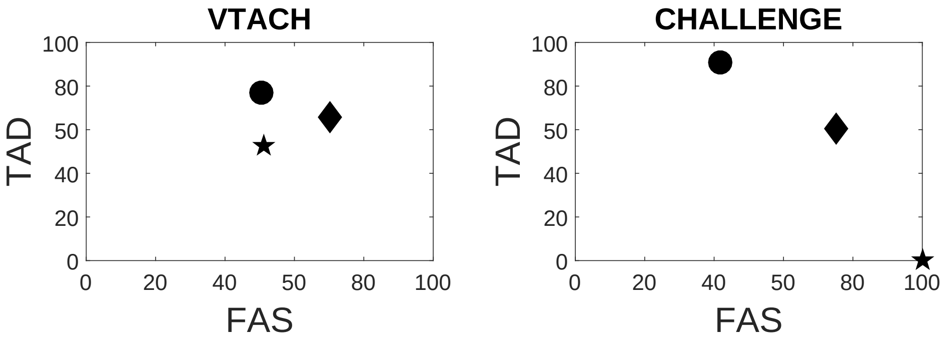


FIGURE 3.2: CS-LTS[•] vs. LTS[⧫] vs. BEHAR[★] in terms of true alarm detection (TAD) and false alarm suppression (FAS) rates over 2 critical alarm datasets. CS-LTS achieves higher TAD on both datasets compared to LTS and BEHAR.

(Fig. 3.2) CS-LTS (circle) achieves higher TAD (Y-axis) than LTS (diamond) and the current state-of-the-art baseline BEHAR (Behar et al., 2013) (star) in the field of critical alarm detection. FAS (X-axis) is better for LTS (diamond) on both datasets compared to CS-LTS (circle). However, improving TAD by decreasing FAS is acceptable as missing true alarms may result in patient fatality. CS-LTS (circle) beats BEHAR (star) in terms of true alarm detection rate on both the datasets. In terms of false alarm suppression, CS-LTS achieves comparable performance on VTACH dataset. BEHAR (star) achieves 100% FAS for CHALLENGE dataset, however, true alarm detection rate is 0. Fig. 3.3 shows the comparison of F_β scores for VTACH and CHALLENGE datasets. In both datasets CS-LTS outperforms LTS with respect to β = 2 and β = 3. This proves that CS-LTS improves the TAD score on both datasets when compared to LTS.

3.7.2 Balanced time-series Datasets

In this set of experiments, the proposed model attains comparable or better classification accuracy when compared to state-of-the-art LTS on balanced datasets. So, incorporating cost sensitive learning does not hurt the optimization algorithm because it automatically learns the cost sensitive parameters. This is very useful if the intrinsic sensitivity of the data is not known a priori.

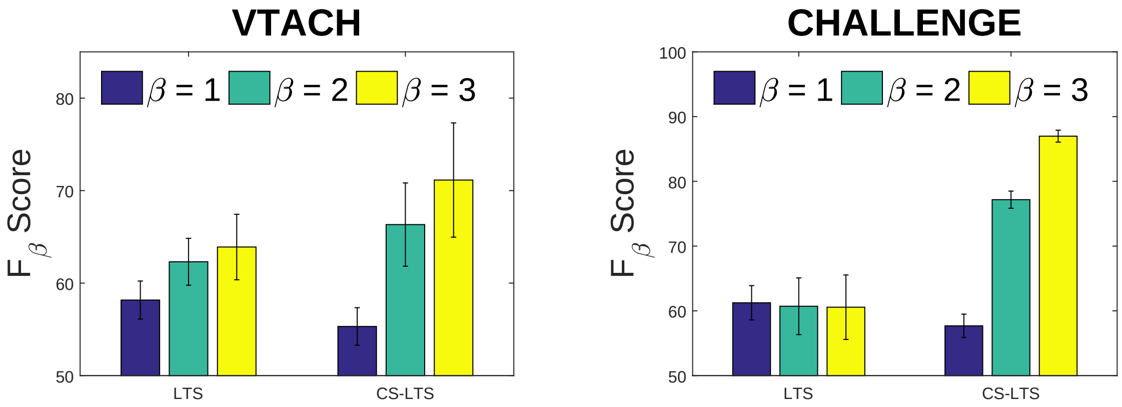


FIGURE 3.3: Comparison of CS-LTS vs. LTS in terms F_1, F_2 andF_3 scores over 2 false alarm suppression datasets.

Sixteen binary-class datasets were selected from UCR time-series repository (Chen et al., 2015). In order to ensure fair comparison with LTS, the default train and test splits were used. Ten independent runs (with different initialization for both LTS and CS-LTS) were conducted and the average and standard deviation of the evaluation metric are reported.

The results of comparing CS-LTS to LTS on the 16 datasets are shown in Fig. 3.4. It is observed that CS-LTS outperforms or comparable to LTS on all 16 datasets. This set of experiments highlights the fact that the CS-LTS model provides a good alternative to LTS as it can handle balanced datasets quite effectively. The proposed method attains higher sensitivity with little loss of specificity when compared to LTS.

3.7.3 Imbalanced time-series Datasets

In order to highlight the advantage of cost-sensitive learning over cost-insensitive learning, in this set of experiments, the proposed model was extensively evaluated on 18 highly imbalanced datasets and compare it with LTS and different over-sampling and under-sampling methods. The imbalanced time-series datasets were constructed by authors in Cao et al. (2014) from 5 multi-class datasets from the UCR time-series repository and the

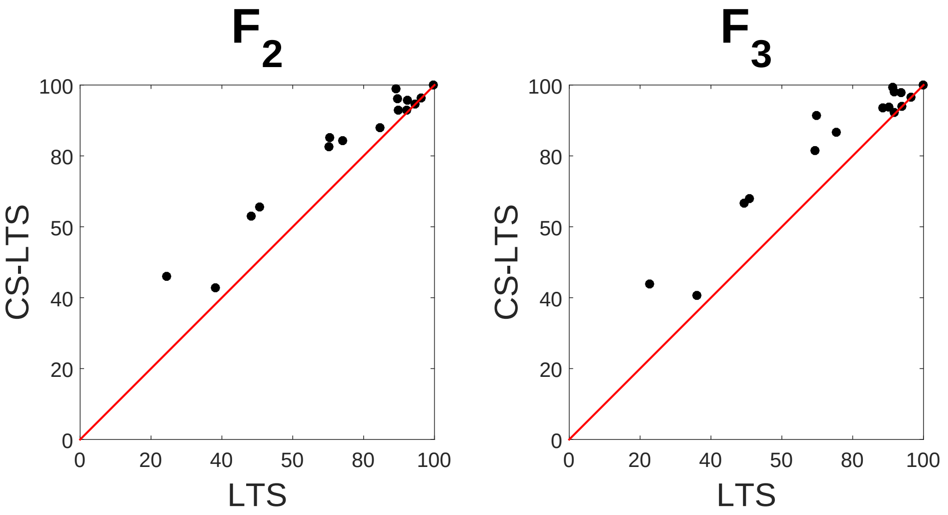


FIGURE 3.4: F_2 and F_3 scores between CS-LTS and LTS for 16 balanced time-series datasets. (Left) In terms of F_2 score CS-LTS outperforms or is comparable to LTS in all 16 datasets. (Right) In terms of F_3 score CS-LTS outperforms or is comparable to LTS in all 16 datasets.

Table 3.1: Imbalanced datasets constructed from UCR Repository (Chen et al., 2015) where $*$ is the index of the original class that is assumed as the positive class

Dataset	Training			Test		Length
	#Positive	#Negative	IM Ratio	#Positive	#Negative	
FaceAll*	80-150	1000	6.7 - 12.5	91-123	977 - 1079	131
SLeaf*	35	450	12.9	40	600	128
TwoPatterns*	200	180	9	1001 - 1106	1894 - 1999	128
Wafer*	200	380-3000	1.9-15	562 - 6220	392 - 3402	152
Yoga*	200	800-900	4 - 4.5	1300 - 1570	730 - 870	426

details are shown in Table 3.1.

The main advantage of CS-LTS over LTS is its superior performance in case of imbalanced datasets. In Fig. 3.5, it is shown that CS-LTS comfortably outperforms LTS on all 18 imbalanced datasets in terms of both F_1 and F_2 scores.

Moreover, in comparison to the state-of-the-art methods for imbalanced time-series classification, CS-LTS is very competitive. As shown in Table 3.2 in terms of F_1 score. The best method per dataset is shown in bold. The proposed CS-LTS method attains the highest number of absolute wins (5.86 wins) where a point is awarded to a method if

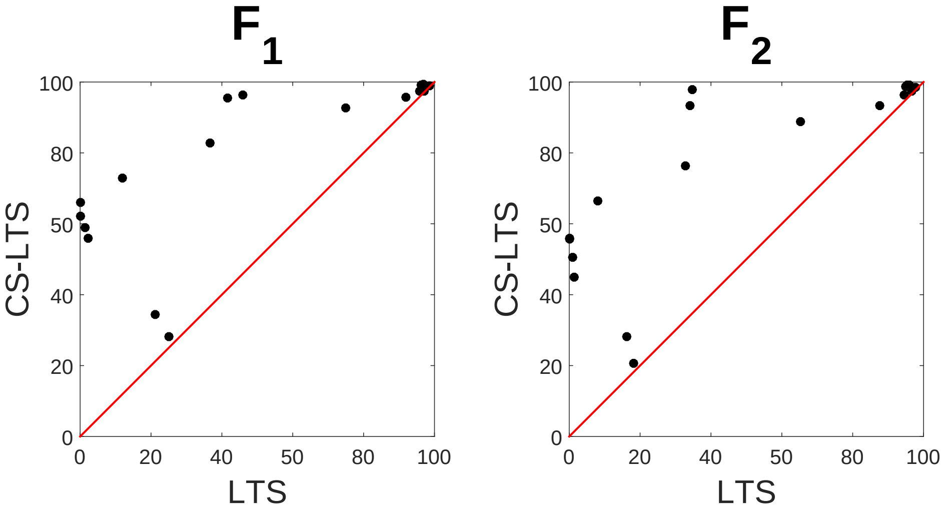


FIGURE 3.5: F_1 and F_2 score between CS-LTS and LTS for 18 imbalanced time-series datasets. (Left) In terms of F_1 score CS-LTS achieves very high accuracy compared to LTS on 15 datasets and is comparable to LTS in 3. (Right) In terms of F_2 score CS-LTS outperforms or is comparable to LTS in all 18 datasets.

it attains the highest F_1 score among the rest of the baseline methods for that particular dataset. In case of draws, the point is split into equal fractions and awarded to each method having the highest F_1 for a particular dataset.

3.8 Discussion

Amongst the baselines, SPO (Cao et al., 2011), SMOTE (Chawla et al., 2002), BORSMOTE (Han et al., 2005), ADASYN (He et al., 2008), DB (Guo and Viktor, 2004) and MoGT (Cao et al., 2014) are over-sampling techniques which mostly act as a preprocessing technique to over sample the rare class examples in order to construct balanced datasets. Easy (Liu et al., 2009) and Balanced (Liu et al., 2009) are under-sampling methods which reduces the number of examples from the majority class via under-sampling the majority class to balance the datasets.

From Table 3.2, it can be inferred that CS-LTS beats LTS and Easy across all datasets

Table 3.2: Comparison of mean F_1 scores for various baseline methods against proposed method. CS-LTS achieves highest absolute wins.

Dataset	SPO (Cao et al., 2011)	Repeat	SMOTE (Chawla et al., 2002)	BORSMOTE(Han et al., 2005)	ADASYN(He et al., 2008)	DB(Guo and Viktor, 2004)	1MoGT (Cao et al., 2014)	2MoGT (Cao et al., 2014)	1 NN	Easy (Liu et al., 2009)	Balanced (Liu et al., 2009)	LTS (Grabocka et al., 2014)	CS-LTS
FaceAll1	96(0.9)	94(0.0)	95(0.6)	95(0.5)	95(0.5)	95(0.8)	96(0.5)	97(0.5)	98(0.0)	67(5.9)	86(2.4)	98(0.4)	**99(0.2)**
FaceAll2	93(1.0)	83(0.0)	88(0.5)	88(0.7)	88(0.8)	92(0.4)	90(0.5)	86(0.8)	83(0.0)	76(3.2)	93(1.3)	93(0.4)	**95(0.4)**
FaceAll3	95(0.6)	**97(0.0)**	96(0.6)	**97(0.2)**	96(0.4)	91(0.4)	95(0.6)	94(0.6)	**97(0.0)**	60(6.6)	73(2.7)	90(2.6)	92(0.4)
FaceAll4	94(0.5)	96(0.0)	95(0.6)	96(0.5)	96(0.5)	90(1.0)	95(0.5)	95(0.5)	96(0.0)	72(3.0)	87(2.7)	94(0.3)	**98(0.1)**
FaceAll5	96(0.4)	**97(0.0)**	**97(0.1)**	**97(0.1)**	**97(0.2)**	95(0.3)	**97(0.2)**	95(0.3)	95(0.0)	85(2.5)	92(1.1)	95(0.4)	**97(0.1)**
SLeaf1	83(0.8)	81(0.0)	79(1.4)	79(1.6)	79(1.6)	81(1.6)	**87(2.1)**	83(1.7)	57(0.0)	54(5.1)	50(4.4)	4(20.2)	49(1.8)
SLeaf2	96(1.0)	94(0.0)	95(0.7)	96(0.0)	96(0.4)	96(0.0)	**98(0.7)**	95(0.3)	91(0.0)	85(6.7)	87(3.9)	96(1.5)	**98(0.5)**
SLeaf3	**88(1.6)**	83(0.0)	83(1.0)	83(1.1)	83(1.1)	82(0.5)	84(0.7)	84(1.4)	66(0.0)	66(5.6)	54(6.6)	0.0(0.0)	84(1.6)
SLeaf4	**93(1.0)**	61(0.0)	72(2.4)	71(0.7)	73(0.4)	89(1.5)	83(2.9)	88(1.9)	68(0.0)	56(7.9)	66(4.8)	66(36.7)	88(0.0)
SLeaf5	**90(1.1)**	88(0.0)	89(0.7)	89(0.7)	89(0.6)	87(0.8)	89(1.0)	89(0.8)	71(0.0)	59(8.3)	52(5.2)	36(3.9)	82(1.4)
TwoPatterns1	92(0.3)	71(0.0)	77(0.2)	77(0.2)	78(0.3)	89(0.2)	84(0.6)	84(0.6)	92(0.0)	95(4.0)	75(1.6)	96(1.4)	**99(1.4)**
TwoPattern2	78(0.7)	65(0.0)	68(0.3)	68(0.1)	68(0.2)	73(0.2)	75(0.5)	81(0.6)	**89(0.0)**	31(2.4)	68(1.2)	51(1.7)	72(3.4)
Twopattern3	86(0.3)	65(0.0)	70(0.4)	71(0.5)	71(0.7)	57(0.2)	82(0.6)	89(0.6)	**91(0.0)**	36(3.0)	69(0.9)	5(13.1)	51(11.3)
TwoPattern4	90(0.5)	68(0.0)	73(0.2)	73(0.2)	73(0.2)	73(0.2)	82(0.7)	87(0.4)	87(0.0)	35(2.5)	71(1.5)	96(1.4)	**99(1.2)**
Wafer0	**99(0.0)**	**99(0.0)**	**99(0.0)**	**99(0.0)**	**99(0.0)**	**99(0.0)**	**99(0.0)**	**99(0.0)**	**99(0.0)**	93(1.1)	**99(0.1)**	98(0.6)	**99(0.0)**
Wafer1	**99(0.1)**	**99(0.0)**	**99(0.1)**	**99(0.0)**	**99(0.1)**	**99(0.1)**	**99(0.1)**	**99(0.1)**	98(0.0)	93(0.8)	98(0.6)	97(1.6)	**99(0.1)**
Yoga1	89(0.2)	88(0.0)	**90(0.1)**	**90(0.2)**	**90(0.2)**	88(0.0)	88(0.1)	88(0.2)	83(0.0)	59(2.5)	85(0.6)	24(1.7)	70(0.9)
Yoga2	**91(0.2)**	90(0.0)	**91(0.1)**	**91(0.1)**	**91(0.1)**	**91(0.0)**	90(0.1)	**91(0.1)**	86(0.0)	61(2.5)	87(0.7)	5(0.0)	81(1.7)
Absolute Wins	3.36	0.69	0.85	1.18	0.85	0.36	1..86	0.36	2.42	0	0.09	0	5.86

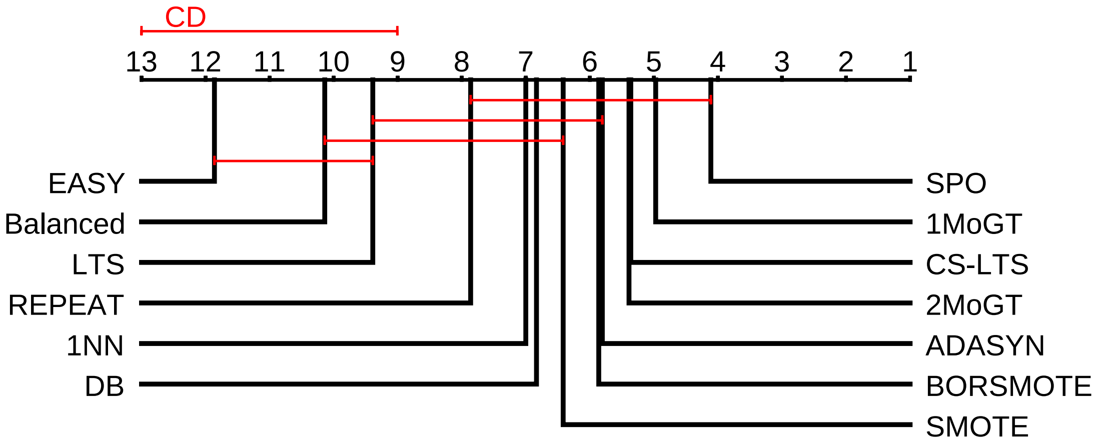


FIGURE 3.6: Critical difference diagram showing average rank of CS-LTS against all baseline methods on 18 imbalanced datasets.

except 1 dataset (*TwoPatterns3*) in case of LTS which is a draw. Comparing with other baseline methods it is seen that CS-LTS has achieved similar accuracy as baseline methods on more than one datasets (such as *wafer0* and *wafer1*). CS-LTS achieves comparable results with almost all of the over-sampling methods except for *sleaf1* and *TwoPatterns3* dataset. Results of CS-LTS on *Sleaf1* and *TwoPatterns3* certainly outperform LTS by huge margins; however, due to overlapping data-points in the feature space, it is hard for a linear model to achieve high classification accuracy in these two datasets. Compared to under-sampling methods (Easy and Balanced), CS-LTS is better than these baseline methods on most of the datasets. Another comparable method is the 1-Nearest Neighbor method (1-NN) which is known to be a good classifier for time-series classification problems. However, 1-NN computationally suffers from high dimensionality, hence it is time consuming compared to the proposed method. Moreover, CS-LTS is an easier-to-interpret method as compared to 1-NN which makes it more desirable to domain experts.

CS-LTS is an algorithmic approach to solve the imbalanced time-series classification problem whereas the state-of-the-art methods in this field are data manipulation methods that use over-sampling and under-sampling techniques, which act as a pre-processing step to solve the high imbalance time-series classification problem. Fig. 3.6 shows the critical difference diagram amongst all the baseline methods and CS-LTS.

3.9 Conclusion

In this chapter, the novel perspective of learning generalized shapelets for time-series classification via a logistic loss minimization is adapted, and the time-series classification framework is extended to a cost-sensitive framework that can handle highly imbalanced time-series datasets. In contrast to the baseline model, whose prediction accuracy is biased towards the abundant negative class, the proposed CS-LTS does not suffer from class imbalance problem. Extensive experiments on 36 real-world time-series datasets reveal the proposed method is a good alternative to the baseline model. It can handle both balanced and imbalanced time-series datasets and achieve better or comparable results against the current state-of-the-art methods.

CHAPTER 4

LEVERAGING SUBSEQUENCE-ORDERS FOR TIME-SERIES CLASSIFICATION

4.1 Introduction

A majority of shapelet discovery methods are focused on univariate time-series data. Only a handful of methods (e.g., (Cetin et al., 2015; Ghalwash et al., 2013b; Grabocka et al., 2016)) 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. 4.1, 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.

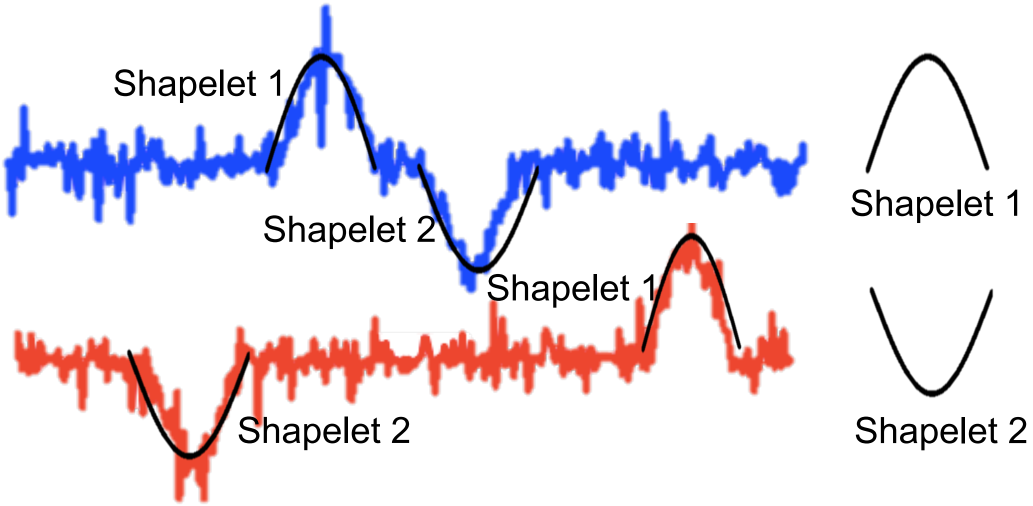


FIGURE 4.1: The blue univariate time-series is from class 1, and the red univariate time-series is from class 2. Shapelet 1 and Shapelet 2 could misclassify either classes, as they are present in both classes. However, considering pairwise shapelet-orders allows to differentiate the blue from the red time-series.

In this chapter, a novel scheme, named *TimeGap-based-orders*, to extract informative orders among pairwise shapelets by considering the time gap between any pairs of shapelets is explored. Based on this scheme, a novel model, Pairwise Shapelet-Orders Discovery (PSOD), which extracts both informative shapelets and shapelet-orders and incorporates the shapelet-transformed space with shapelet-order space for time-series classification is proposed. The 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.

4.2 Contribution

The main contributions of this chapter 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 significantly improved on average over baseline methods.

4.3 Related work

In the field of time-series classification, extracting shapelets to perform classification has recently received extensive attention (Hou et al., 2016; Roychoudhury et al., 2017; Xing et al., 2011; Zhang et al., 2016). The minimum distance between a shapelet and a time-series, namely shapelet transformation (Hills et al., 2014; Lines et al., 2012), 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 (Ghalwash et al., 2014; Ye and Keogh, 2009a) conduct an exhaustive search of all possible subsequences, which is often intractable for large datasets. Numerous methods (Grabocka et al., 2016; Karlsson et al., 2016; Mueen et al., 2011; Rakthanmanon and Keogh, 2013; Ye and Keogh, 2009a) 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 (Grabocka et al., 2014; Hou et al., 2016; Zhang et al., 2016) are learned from the data. The above approaches are mainly designed for univariate time-series datasets. A few studies (e.g. (Bostrom and Bagnall, 2017; Cetin et al., 2015; Ghalwash et al., 2013b; Grabocka et al., 2016; Karlsson et al., 2016)) 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. (Mueen et al., 2011) 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. (Patri et al., 2015) 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, a formal generalized method is proposed 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 (chung Fu, 2011) from time-series. A common approach is to first discretize (Das et al., 1998; Leigh et al., 2002; Ting et al., 2006) 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 (Shokoohi-Yekta et al., 2015). Tatavarty et al. (Tatavarty et al., 2007) 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, this study focuses on discovering **discriminative** shapelets and the temporal gap among them in both univariate and multivariate time-series data for classification. To the best of my knowledge, this study 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, the proposed approach is applicable to both univariate and multivariate time-series datasets.

4.4 Method Preliminaries

A time-series dataset composed of I training instances is denoted as $\mathbf{T} \in \mathbb{R}^{I \times D \times L}$. Instances are considered to have d $(1 \leqslant d \leqslant D)$ dimensions where each $\mathbf{T}_i$ $(1 \leqslant i \leqslant I)$ is of length L (for notation convenience one can 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, definitions of some terminologies used in this study are introduced.

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,

$$m_{i,k} = \min_{q=1,\ldots L-l+1} \sqrt{\frac{1}{l}\sum_{p=1}^{l}(\mathbf{T}_{i,d,p+q-1} - s_{d,p}^k)^2}, \tag{4.1}$$

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 (Hills et al., 2014). 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,

$$B_i^k = \operatorname*{argmin}_{q} \sqrt{\frac{1}{l}\sum_{p=1}^{l}(\mathbf{T}_{i,d,p+q-1} - s_{d,p}^k)^2} \tag{4.2}$$

The benefit of using shapelet-orders to correctly classify time-series has been shown in Fig. 4.1. 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, a scheme is proposed, 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,

$$g_i(s^{k_1}, s^{k_2}) = B_i^{k_1} - B_i^{k_2}. \tag{4.3}$$

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 d is omitted in their notations.

4.5 Model Description:

In this section, **P**airwise **S**hapelet-**O**rders **D**iscovery (PSOD) model for extracting informative shapelets and pairwise shapelet-orders for time-series classification is introduced. 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. First the process of identifying candidate shapelets is discussed, followed by the identification of candidate

Algorithm 2 Selection of a random candidate shapelet

```
0: procedure SEARCH
   Input: $T \in \mathbb{R}^{I \times D \times L}$, Accepted shapelet list $\mathcal{A}$, Rejected shapelet list $\mathcal{R}$, Distance threshold $\varepsilon_d$
0:   Draw random series: $i \sim \mathcal{U}\{1, \cdots, I\}$;
0:   Draw random dimension: $d \sim \mathcal{U}\{1, \cdots, D\}$;
0:   Draw random shapelet length: $l \sim \mathcal{U}\{1, \cdots, L\}$;
0:   Draw random start point: $p \sim \mathcal{U}\{1, \cdots, L - l + 1\}$;
0:   Randomly selected candidate: $s^k \leftarrow \mathbf{T}_{i,d,p:p+l-1}$;
0:   if $s^k$ is not similar to any previously accepted shapelets in $\mathcal{A}$ as well as any rejected shapelets in $\mathcal{R}$ then;
0:     Return $s^k$
0:   else
0:     $\mathcal{R} = \mathcal{R} \cup s^k$
0:   end if
0: end procedure=0
```

orders. In each case, a confidence measure is introduced to evaluate the quality of a candidate shapelet as well as a candidate order respectively.

4.5.1 *Randomized shapelet candidate extraction:*

Inspired from the huge speed up by Grabocka et al. (Grabocka et al., 2016), a similar shapelets extraction approach to randomly select subsequences from time-series, and then evaluate it by computing classification accuracy has been considered. The steps to randomly select a candidate shapelet is summarized in Algorithm 2.

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 (Grabocka et al., 2016), 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 $\mathcal{R}$.

4.5.2 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,

$$p_{ij} = \frac{e^{\alpha||\mathbf{m}_i - \mathbf{m}_j||^2}}{\sum_{z=1\cdots I, z\neq i} e^{\alpha||\mathbf{m}_i - \mathbf{m}_z||^2}} \quad , \qquad p_{ii} = 0 \tag{4.4}$$

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,

$$p^{\mathbf{S}}_{i,c} = \sum_{Y_j = c} p_{ij} \tag{4.5}$$

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 $\mathbf{T}_i$.

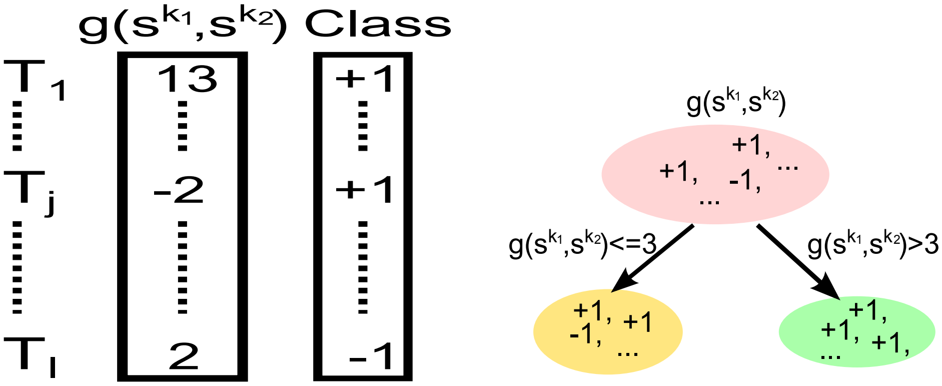


FIGURE 4.2: An example of finding a TimeGap-based-order candidate.

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}, the potential orders between s^{k_1} and s^{k_2} is extracted using TimeGap-based-order scheme introduced in Sec. 5.4.

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. 4.3. 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 $g(s^{k_1}, s^{k_2}) \leqslant h$, and the right subset contains the instances which satisfy $g(s^{k_1}, s^{k_2}) > h$. An illustrated procedure is shown in Fig. 4.2. 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 $g(s^{k_1}, s^{k_2}) \leqslant h$ 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. 4.2, 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 $g(s^{k_1}, s^{k_2}) \leqslant h$ shall be assigned to the class of $+1$. In this example, $o : g(s^{k_1}, s^{k_2}) \leqslant h$, and $Y_o = +1$.

The precision of the candidate order o of *Class* = c is defined as

$$P(Class = c|o \text{ exists}) = \frac{P(o \text{ exists}|Class = c)\; P(Class = c)}{P(o \text{ exists})} \tag{4.6}$$

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,

$$\mathbb{C}(o) = P(Class = c|o \text{ exists}) \times P(Class = c \cap o \text{ exists}) \tag{4.7}$$

Both terms in Eq. A.5 are probabilities, thus the confidence measure for order o is a value between 0 and 1. Eq. A.1 and Eq. A.5 are detailed in Appendix A.

4.5.3 Updating class membership confidence using orders:

Let **O** denote order space. The class membership confidence $p^{\mathbf{O}}_{i,c}$ 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 $\mathbf{T}_i$,

$$p^{\mathbf{O}}_{i,c} = \mathbb{C}\left(\bigcup_{Y_{o_n}=c \,\cap\, o_n \text{ occurs in } \mathbf{T}_i} o_n\right) \tag{4.8}$$

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

$$\begin{aligned} p^{\mathbf{O}}_{i,c} &= \mathbb{C}(o_1 \cup o_2) = \mathbb{C}(o_1) + \mathbb{C}(o_2) - \mathbb{C}(o_1 \bigcap o_2) \\ &= \mathbb{C}(o_1) + \mathbb{C}(o_2) - \mathbb{C}(o_1) * \mathbb{C}(o_2) \end{aligned} \tag{4.9}$$

In a general case when there are multiple orders, Eq. 4.8 can be calculated according to the inclusion-exclusion principle of probability.

Algorithm 3 Pairwise shapelet-orders discovery - training

```
procedure PSODD-TRAIN
Input: $T \in \mathbb{R}^{I \times D \times L}$, Labels $Y \in \mathbb{C}^{I}$
Initialize: Accepted Shapelets list $\mathcal{A} \leftarrow \varnothing$, Accepted order list $\mathcal{O} \leftarrow \varnothing$, Rejected shapelet list $\mathcal{R} \leftarrow \varnothing$;
  for iteration = 1: $\mathbb{N}_1^{ILQD}$ do
    $acc \leftarrow$ ACCURACY($\mathcal{A}, \mathcal{O}$ );
    $s^k \leftarrow$ SEARCH();
    $\{\mathcal{A}, \mathcal{O}\} \leftarrow$ EVALUATE($s^k, acc$);
  end forReturn $\mathcal{A}, \mathcal{O}$;
end procedure=0
```

The initial class membership confidence for each time-series instance is computed in the shapelet-transformed space using Eq. 4.5. 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,

$$P(Y_i = c|\mathbf{M}, \mathbf{O}) = p_{i,c}^{\mathbf{S}} \times p_{i,c}^{\mathbf{O}} \tag{4.10}$$

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

$$P(Y_i = c|\mathbf{M}, \mathbf{O}) = p_{i,c}^{\mathbf{S}} \times \frac{1}{C} \tag{4.11}$$

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

4.5.4 Pairwise shapelet-orders discovery

The training phase of PSOD is now introduced. The pictorial representation of the framework is present in Appendix B and the pseudo code is outlined in Algorithm 3. The model begins by searching a candidate shapelet using SEARCH() function (outlined in Algorithm 2), and then evaluating the candidate shapelet s^k as well as the potential pairwise shapelet-orders using EVALUATE() (outlined in Algorithm 4). This process

Algorithm 4 Evaluate candidate shapelets

```
0: procedure EVALUATE
   Input: Accepted shapelet list $\mathcal{A}$, Accepted order list $\mathcal{O}$, Rejected shapelet list $\mathcal{R}$,
   current accuracy $acc$, a candidate shapelet $s^k$;
0:   tempAcc1 ← ACCURACY($\mathcal{A}$, $\mathcal{O}$, $s^k$)
0:   tempAcc2 = 0, tempOrder = 0;
0:   for m = 1, $\cdots$, $|\mathcal{A}|$ do
0:     Extract a order candidate $o_m$ between $s^k$ and $s^m$.
0:     tempAcc2 ← ACCURACY($\mathcal{A}$, $\mathcal{O}$, $s^k$, $o_m$);
0:     if tempAcc2 >tempAcc1 then
0:       tempAcc1 = tempAcc2
0:       tempOrder = $o_m$
0:     end if
0:   end for
0:   if tempAcc1 $>$ $acc$ and tempOrder ! = 0 then
0:     $\mathcal{O} \leftarrow \mathcal{O} \bigcup$ tempOrder;
0:     $\mathcal{A} \leftarrow \mathcal{A} \bigcup \{s^k\}$;
0:   else
0:     if tempAcc1 $>$ $acc$ and tempOrder == 0 then
0:       $\mathcal{A} \leftarrow \mathcal{A} \bigcup \{s^k\}$;
0:     end if
0:   end if
   Return $\mathcal{A}, \mathcal{O}$
0: end procedure=0
```

(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.

In Algorithm 3 line 7, EVALUATE() returns an updated list of accepted shapelets and an updated list of orders. In EVALUATE() (Algorithm 4), a candidate shapelet s_k is first evaluated (line 3) by calculating the classification accuracy (outlined in Algorithm 5). 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 5), 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. 4.10, otherwise, it is computed according to Eq. 4.11. Note that the class membership confidence is calculated for each class (Algorithm 5, line 5), and the predicated class of instance $\mathbf{T}_i$ is the class with the highest probability (Algorithm 5, line 12).

The pseudocode of the testing phase of PSOD is outlined in Algorithm 6. 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 is computed based on Eq. 4.4, and the class membership confidence for $\mathbf{T}'_i$ in shapelet-transformed space is computed using Eq. 4.5. 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. 4.10, otherwise, it will be updated based on Eq. 4.11. The class with the highest membership confidence will be selected as the final predicted class.

4.5.5 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 $\mathcal{O}(IL^2)$. In Eq. 4.5, the 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 $f = \frac{\#Accepted\ shapelts + \#Rejected\ shapelets}{IL^2}$.

Algorithm 5 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
          $P(Y_i = c|\mathbf{M}, \mathbf{O}) = p^{\mathbf{S}}_{i,c} \times \frac{1}{C}$;
        end if
      end if
    end for
    $\hat{Y}_i = \mathrm{argmax}_c P(Y_i = c|\mathbf{M}, \mathbf{O})$;
    if $\hat{Y}_i == Y_i$ then
      $acc = acc + 1$;
    end if
  end for
  $acc = acc/I$ ;
  $\mathcal{A} \leftarrow \mathcal{A} \backslash \{s^k\}$, $\mathcal{O} \leftarrow \mathcal{O} \backslash \{o\}$
  Return $acc$
end procedure=0
```

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 4.1, PSOD's training time is empirically compared with state-of-art shapelet based methods on different datasets.

Algorithm 6 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
            $P(Y_i = c|\mathbf{M}, \mathbf{O}) = p^{\mathbf{S}}_{i,c} \times \frac{1}{C}$;
          end if
        end if
      end for
    end for
    $\hat{Y}_i = \text{argmax}_c \, P(Y_i = c|\mathbf{M}, \mathbf{O})$;
  end for
  Return $\hat{Y}$
end procedure=0
```

4.6 Experimental Evaluation

The proposed model[1] 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 (Bagnall et al., 2017). Details about each univariate datasets can be viewed on repository's website[2]. Moreover, 6 multivariate datasets are chosen that were used in (Grabocka et al., 2016) to highlight the advantage of shapelet-orders in real-world multidimensional time-series datasets.

[1] Code Link for PSOD: https://bitbucket.org/shoumikrc/psod/src

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

4.6.1 Baselines and experimental setup

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, proposed method, PSOD, was compared with 5 state-of-art shapelet-based time-series classification models.

- Scalable Shapelet Discovery (SSD) (Grabocka et al., 2016): This method is a fast procedure to extract random shapelets from the time-series dataset. The proposed 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) (Grabocka et al., 2014): 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) (Rakthanmanon and Keogh, 2013): 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[3] (NS) (Patri et al., 2014): This is an naive extension of the FS algorithm where a d-dimensional multivariate time-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[3] (SF) (Patri et al., 2014): 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

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

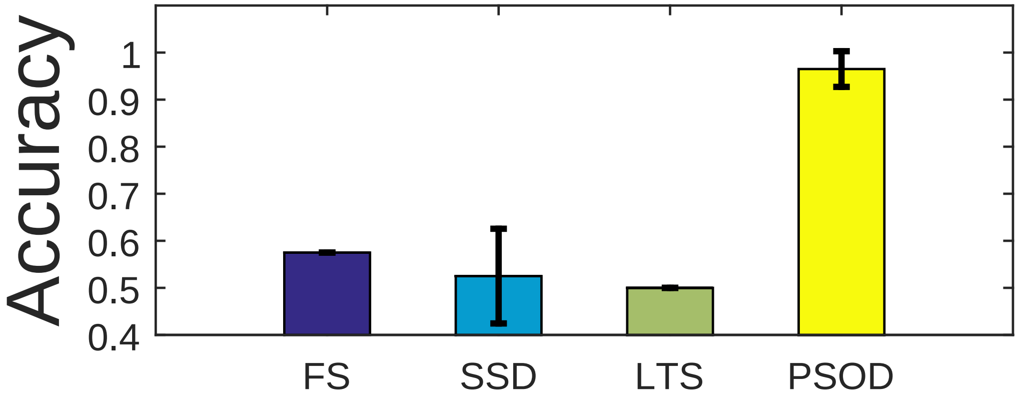


FIGURE 4.3: Average accuracy of FS, SSD, LTS and PSOD on synthetic dataset where orders between shapelets exist in the data.

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.

4.6.2 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. 4.1. In both synthetic datasets, the start-time of patterns were randomly selected, and the remaining points in the time-series follow Gaussian distribution $\mathcal{N}(0, 0.05)$. Moreover, noise sampled from a product distribution comprised of $\mathcal{N}(0, 0.05)$ and $\mathcal{U}(0, 0.25)$ was also added 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. 4.3 shows the classification accuracy obtained by all baselines and PSOD on

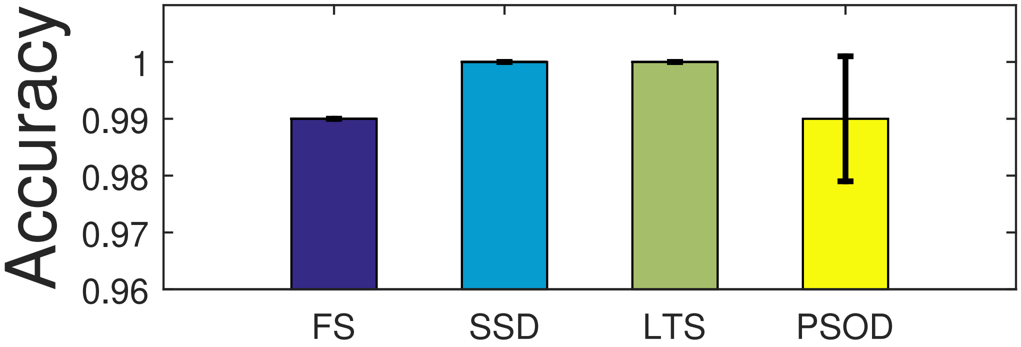


FIGURE 4.4: Average accuracy of FS, SSD, LTS and PSOD on synthetic dataset where no order between shapelets exists in the data.

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 shapelets (Fig. 4.4). Therefore, the benefit of taking shapelet-orders into account was more evident when temporal dependency among pairs of shapelets could differentiate the class.

4.6.3 Analysis of percentile (P) parameter

The sensitivity of the distance threshold ϵ_d in Algorithm 2 was evaluated 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. 4.5 larger percentile values result in more candidate subsequences being rejected with a degradation in accuracy (shown in Fig. 4.6).

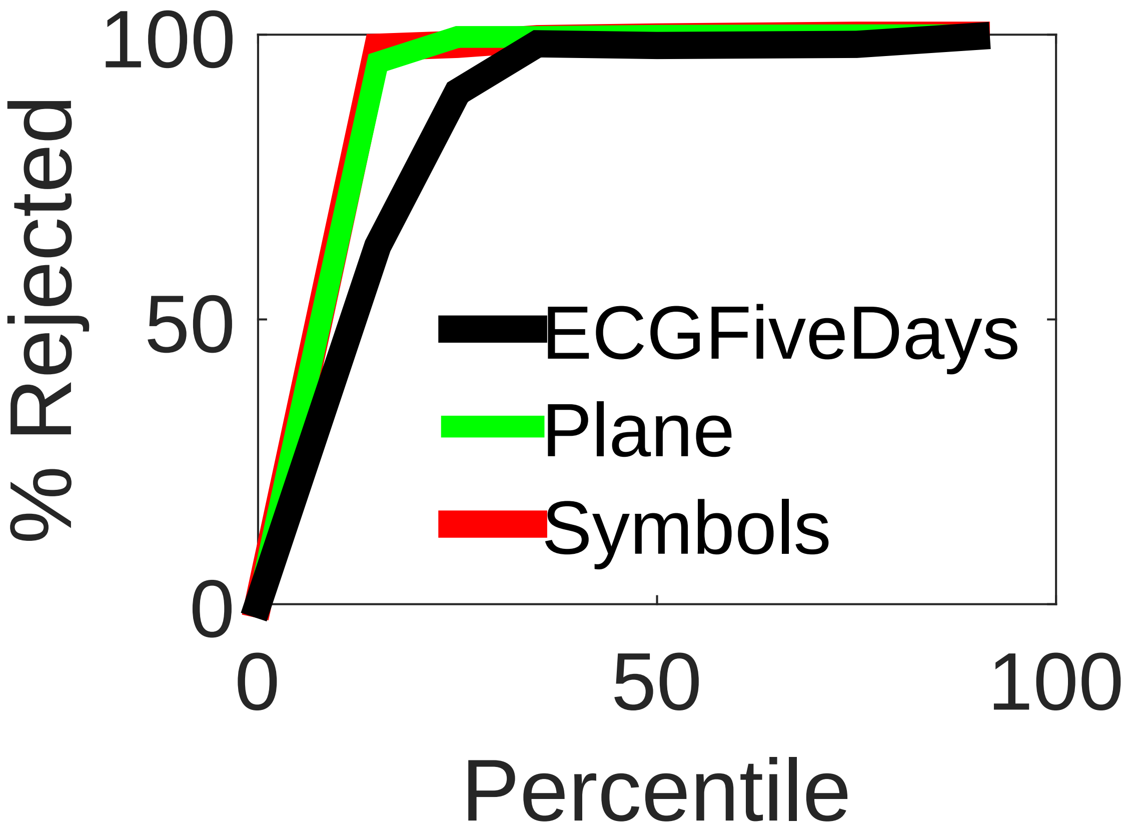


FIGURE 4.5: Percentile Vs. % Shapelet rejected

4.6.4 Results on real-world univariate datasets

First, the performance of PSOD was compared 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 4.1. 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

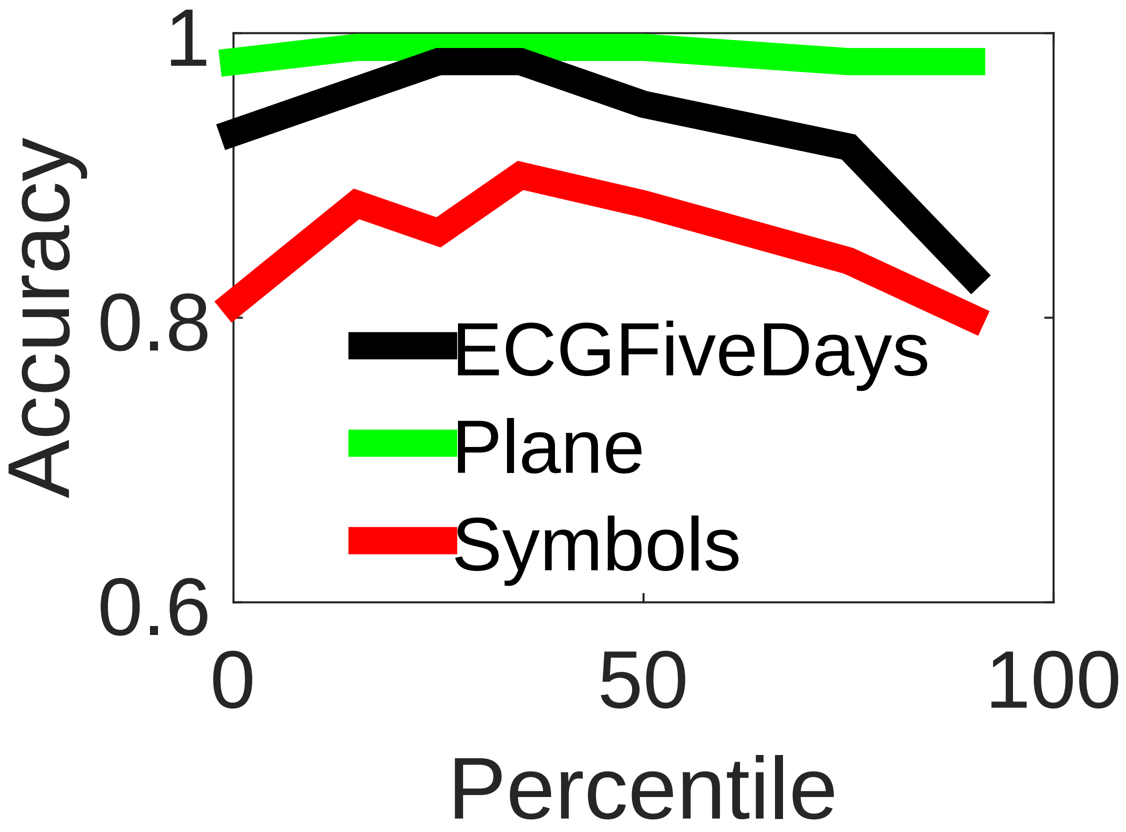


FIGURE 4.6: Percentile Vs. Accuracy

FIGURE 4.7: The effect of varying percentile parameter P on (a) number of shapelets rejected and (b) accuracy.

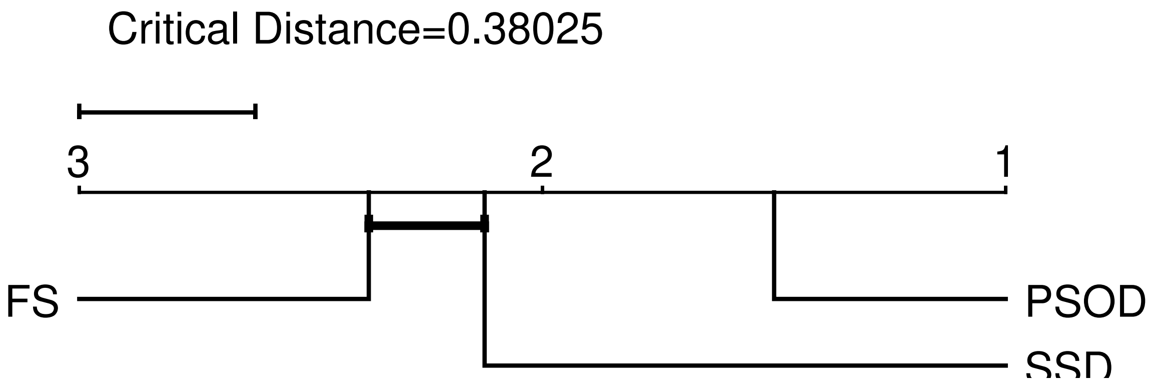


FIGURE 4.8: Statistical significance

FaceAll, which have multiple classes, PSOD still outperform FS and SSD with respect to accuracy. Table 4.1 revealed that (1) although PSOD attained slightly inferior results than

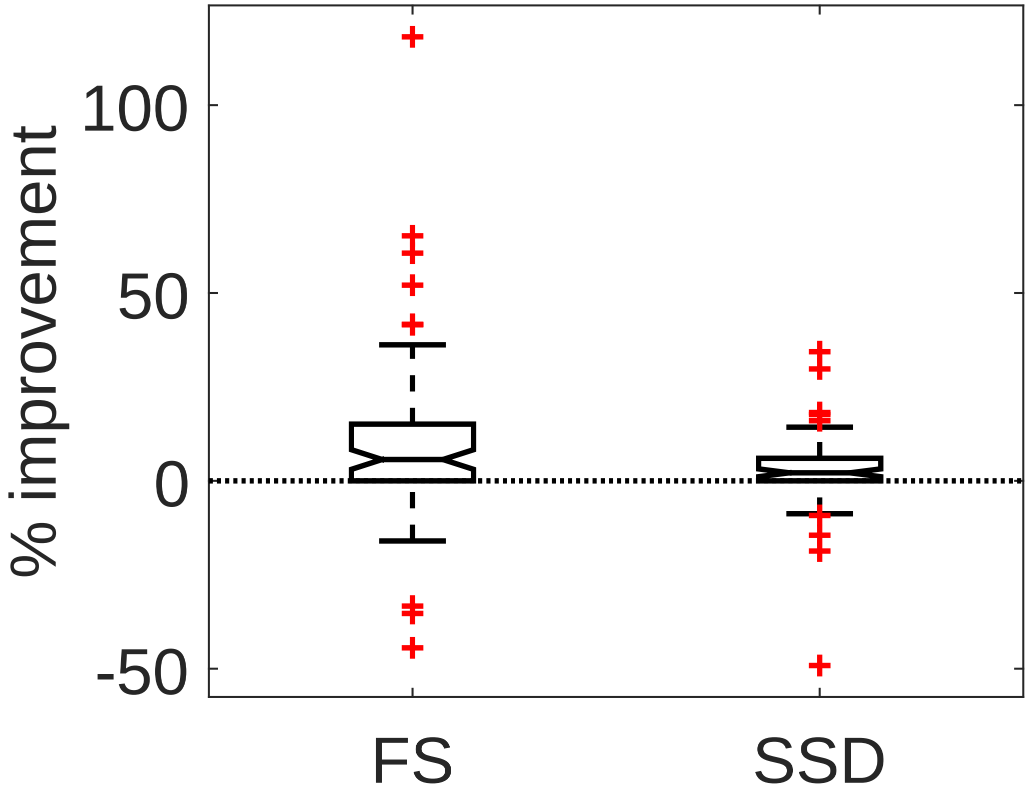


FIGURE 4.9: Accuracy improvement

Table 4.1: 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*)

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, the effectiveness of PSOD was evaluated on 75 real-world univariate datasets

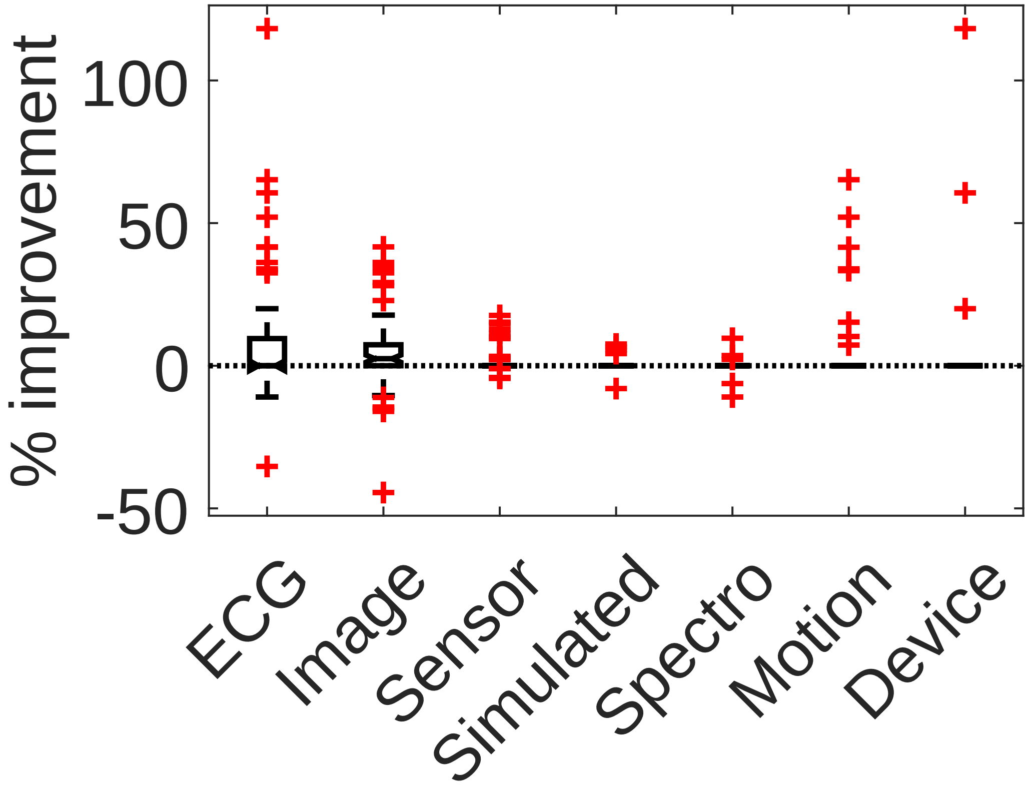


FIGURE 4.10: Improvement over FS

Table 4.2: 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[4]	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**

obtained from 7 categories namely ECG, Image, Sensor, Simulated, Spectro, Motion and Device. The proposed PSOD was compared against FS and SSD only, since LTS is very costly for longer time-series datasets. The significance test, calculated based on (Demšar, 2006), shows that PSOD is significantly better than FS and SSD at the 5% level (see Fig 4.8). The percentage of improvement of PSOD over FS and SSD (plotted in Fig. 4.9)

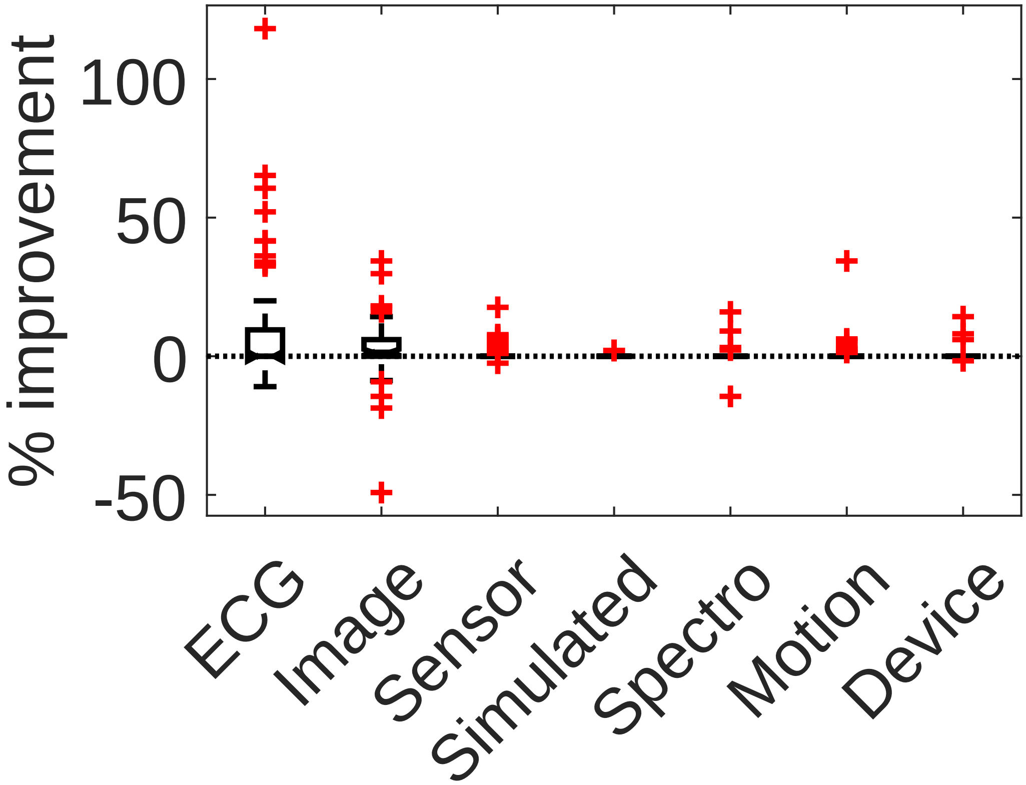


FIGURE 4.11: Improvement over SSD

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. 4.10 and Fig. 4.11 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 4.6.6). Table 4.1 and Fig. 4.6.4 clearly indicate the superiority of PSOD. For more detailed results on individual univariate datasets from different categories please refer to Appendix C.

4.6.5 Results on multivariate datasets

The proposed model was further assessed on 6 real-world multivariate time-series datasets. Their characteristics and the average accuracy of 5 trials are shown in Table 4.2. PSOD was compared with three multivariate time-series classification techniques namely NS, SF and SSD. Table 4.2 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.

4.6.6 Discussion

From the experiments, it was 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 extraction 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.

4.7 Conclusion

In this study, a novel order-generation scheme, *TimeGap-based-orders*, to capture temporal dependency among shapelets is proposed, and a novel model PSOD aimed to

[4] Balanced binary datasets were used.

extract both informative shapelets and shapelet-orders is presented. From the extensive experimental results, it can be inferred that 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.

CHAPTER 5

LEVERAGING TEMPORAL DEPENDENCY AMONG LEARNED SHAPELETS

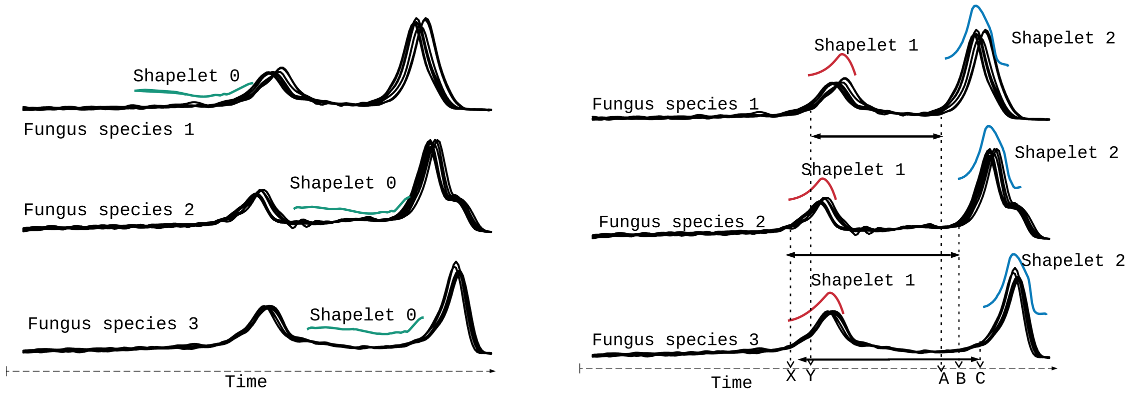
5.1 Introduction

The plethora of time-series data collected from a wide range of domains has significantly increased research interest among data-miners in the realm of time-series classification (Bagnall et al., 2017). Temporal-ordered data collected at equal intervals are available from sensor-based domains such as the internet of things (IoT) (Patri et al., 2014), image outlines (Ye and Keogh, 2009a), spectro-analysis of food products (Bagnall et al., 2012), trajectories of human motion (Mueen et al., 2011), etc. Moreover, with the increasing popularity of wearable devices, smart homes, industrial environment monitoring devices and healthcare devices are just a few under the umbrella domain of IoT that produce time-series data. In all the aforementioned domains a robust time-series classification model is imperative that can handle temporal data more akin to real-world challenges.

Among the numerous approaches that have been investigated for developing time-series classification models, shapelets based methods have garnered much popularity due to their simplicity and interpretable nature. Shapelets (Ye and Keogh, 2009a) are short discriminative temporal patterns (subsequences) that encode local variation information.

Classification models developed using these discriminative time-series subsequences are highly interpretable and generally achieve better classification accuracy than the approaches (Bagnall et al., 2017) which use the global properties of the time-series to determine the category of a time-series.

Numerous methods (such as, (Grabocka et al., 2014, 2016; Hills et al., 2014; Mueen et al., 2011; Xing et al., 2011; Ye and Keogh, 2009a)) have been proposed to discover shapelets for classification of time-series data with multiple real-world applications (Mirowski et al., 2016; Patri et al., 2014; Roychoudhury et al., 2015; Zakaria et al., 2012). At the most basic level, the shapelet discovery methods can be divided into two groups. The first group is search-based techniques (Ghalwash et al., 2014; Grabocka et al., 2016; Ye and Keogh, 2009a) that conduct an exhaustive or randomized search of all possible subsequences. Alternatively, instead of searching all possible subsequences, the second group focuses on learning generalized shapelets (Grabocka et al., 2014; Hou et al., 2016; Zhang et al., 2016) from data. In addition to discovering unique shapelets, in many scenarios, the temporal interactions between shapelets are also helpful (informative) for the classification of the category of the time-series (See Fig. 4.1 which will be discussed later). However, most shapelet discovery methods neglect the temporal interactions among shapelets as they assume that the extracted shapelets are independent of each other. A recent work (Roychoudhury et al., 2019) first presented a notion of time-gap based shapelet-orders to capture the temporal dependency among pairwise shapelets, and proposed a model called Pairwise Shapelet-Orders Discovery (PSOD) (Roychoudhury et al., 2019) to extract both informative shapelets and shapelet-orders. The PSOD model in an iterative manner evaluated the performance of a randomly selected subsequence and all possible shapelet-orders, and then stores the selected subsequence (including subsequence-orders) if they improve classification accuracy. However, one major drawback of PSOD was the random selection of subsequences for candidate shapelet that could lead to an extraction of non-optimal shapelets and eventually poor quality



(a) PSOD

(b) LOS

FIGURE 5.1: High-resolution melt curves of the rDNA internal transcribed spacer (ITS) region of numerous strains of three fungi species. Two shapelets can be visually discovered in the dataset. Both shapelets occur in all three species making it difficult to classify using traditional shapelet based methods.The shapelet 0 was extracted by the current state-of-the-art method PSOD (Roychoudhury et al., 2019), and shapelet 1 and shapelet 2 were discovered by the proposed model LOS. X and Y represent start-time of shapelet 1 in three species, and A, B, C represent start-time of shapelet 2 in three species.

shapelet-orders.

In this paper, we propose a novel model, ***L**earning pairwise **O**rders and **S**hapelet* (LOS), which leverages the time-gap based orders among generalized shapelets. The underlying hypothesis is that leveraging the temporal dependency information of near-to-optimal shapelets improves the quality of the shapelet-orders and further improves the classification performance. Let us explain this using an example. Fig. 4.1 plots the high-resolution melt curves of the rDNA internal transcribed spacer (ITS) region of numerous strains in three fungi species. The melt curve of the ITS region is unique to each species and the curve shapes are conserved across different strains of the same species. It is evident from the figure that each species has two shapelets in their waveform profiles. Using traditional shapelet based algorithms (such as LTS (Grabocka et al., 2014)), it would be impossible to differentiate among three species of fungi, as both red and blue shapelets occur in all three fungi species. It is observed that the difference in time-gaps between the two types of shapelets (see Fig. 5.1b) can be leveraged to uniquely identify a fungi species.

The temporal gap between blue shapelet and red shapelet in case of species 3 $(X - C)$ is larger than in species 2 $(X - B)$ which in turn is larger than the time-gap in species 1 $(Y - A)$. Thus, taking the temporal gap into consideration we can correctly identify different species of fungi from the melt curves.

The shapelets extracted by PSOD and LOS and their locations of occurrences for each fungi species are shown in Fig. 4.1. The PSOD model only extracted one shapelet, which is shapelet 0 (green subsequence in Fig. 5.1a), and therefore did not extract any order. The PSOD model's inability to extract shapelet-orders can be attributed to the randomized selection of subsequences for candidate shapelets. The randomized approach is susceptible to the selection of sub-optimal shapelets, and consequently affects the quality of shapelet-order. In contrast to PSOD, the proposed model LOS identified shapelet 1 and shapelet 2 (red and blue shapelets), and their corresponding time-gap based order information. The classification accuracy obtained by PSOD on the Fungi dataset is 0.33, and obtained by LTS, the shapelet-learning model, is 0.55. However, the proposed model reached an accuracy of 0.85 on the *Fungi* datset (See Table 5.1 in Sec. 5.6).

The proposed model significantly extends the learning shapelet framework (such as LTS (Grabocka et al., 2014)) to jointly learn generalized shapelets and extract informative shapelet-orders among pairwise shapelets. In order to demonstrate the novelty of the proposed model, we consider a variant of the proposed model. The alternative model named as LTS$_{+o}$ consists of two steps. The first step simply applies the shapelet learning model LTS to learn shapelets, and the second step applies a logistic regression model to learn the weights of learned shapelets and the orders among learned shapelets. Additionally, a randomized subsequence initialization for learning generalized shapelets is proposed, instead of the costly k-means algorithm that is used by all existing shapelet learning frameworks, allowing the proposed model to be more scalable than the traditional shapelet learning approaches. The experiments on both synthetic and real-world datasets shows the effectiveness and efficiency of the proposed model.

5.2 Contribution

The contributions of this chapter are the following:

- This study considers *temporal dependency information among pairs of learned generalized shapelets and generates pairwise shapelet-orders among learned shapelets* for use in time-series classification.
- *A model*, ***L**earning pairwise **O**rders and **S**hapelet* (LOS), is proposed to jointly learn the generalized shapelets (highly interpretable patterns) and extract shapelet-orders from data. An optimization problem is proposed which jointly learns shapelets and the classification hyperplane weights.
- A randomized subsequence initialization is considered to allow the model to scale to large datasets.

5.3 Related work

Extracting shapelets for time-series classification has garnered a lot of attention (Hou et al., 2016; Roychoudhury et al., 2017; Xing et al., 2011; Zhang et al., 2016) in the time-series community. The shapelet transformed data (Hills et al., 2014; Lines et al., 2012) which represents the minimum distance between shapelets and the time-series instances can be used as predictors in a traditional classification framework. Therefore, the overall success of time-series classification using shapelets is highly dependent on the discovery of the most discriminative subsequences.

The brute-force search based approaches (Ghalwash et al., 2014; Ye and Keogh, 2009a) for discovering shapelets is often intractable in case of datasets having very long time-series. A number of speedup techniques (Grabocka et al., 2016; Karlsson et al., 2016; Mueen et al., 2011; Rakthanmanon and Keogh, 2013; Ye and Keogh, 2009a) have been proposed to scale the search process for identifying discriminative subsequences from potential shapelet candidates. An alternative to the searching procedure for candidate shapelets is the learning of generalized shapelets. The Learning Time-series Shapelets (LTS) approach (Grabocka et al., 2014) jointly learns generalized shapelets along with weights of a logistic regression model using the shapelet-transformed data as predictors. Successively, (Hou et al., 2016) and (Zhang et al., 2016) have also proposed alternative procedures for learning generalized shapelets. These methods significantly improve prediction accuracy compared to other shapelet-searching-based time-series classification models. Additionally, randomized approaches (e.g., (Grabocka et al., 2016) and (Karlsson et al., 2016)) have also been proposed and have shown robust classification performance in terms of accuracy and speedup.

Most of the existing shapelet-based time-series classification methods focus on how to search (or learn) discriminative shapelets and ignore the temporal dependencies among shapelets, which is also an important feature in some time-series applications. Mueen et al. (Mueen et al., 2011) had proposed *Logical shapelets*, which are logical combinations of multiple shapelets. The shapelets' expressiveness were enhanced using conjunctive and disjunctive logical operations that helped to discover logical rules. However, these discovered rules failed to capture the temporal dependency among shapelets. Patri et al. (Patri et al., 2015) discussed how temporal dependency among shapelets on multivariate time-series datasets can improve classification performance. A concatenated univariate time-series was generated by inter-leaving time-series segments from multiple dimensions. However, this approach was only applicable to multivariate time-series datasets.

An alternative direction of analyzing interactions among temporal patterns has focused on extracting association rules among frequent patterns (chung Fu, 2011). A general approach is to first discretize (Das et al., 1998; Leigh et al., 2002; Ting et al., 2006) the time-series data into segments and convert each segment into a symbol. In the transformed symbolic domain high quality rules (Shokoohi-Yekta et al., 2015) are discovered. Tatavarty et al. (Tatavarty et al., 2007) proposed discovering temporal dependencies between frequently appearing patterns in multivariate time-series. Their work focused on discovering temporal associations among frequently occurring subsequences in different dimensions by transforming the time-series into a symbolic representation. On the contrary, we focus on learning **discriminative** and **unique** shapelets (patterns) and leverage the temporal gap among learned shapelets for classification.

Recently, Pairwise Shapelet Order Discovery (Roychoudhury et al., 2019) was proposed as a formal generalized method to extract temporal dependency among pairs of informative shapelets. The proposed method jointly extracts informative shapelets and the most informative pairwise shapelet-orders that enhance the confidence of prediction on both univariate and multivariate time-series. However, the randomized selection of time-series subsequences for candidate shapelets could lead to selection of non-optimal shapelets that could further lead to generation of poor quality shapelet-orders. In this paper, the learning generalized shapelets framework is extended by augmenting the shapelet-transformed space with the shapelet-order space, and propose the step of randomly initializing subsequences to increase the model's scalability.

5.4 Method Preliminaries

A multi-class time-series dataset composed of I training examples denoted as $\mathbf{T} \in \mathbb{R}^{I \times N}$ is considered, where each $\mathbf{T}_i$ $(1 \leqslant i \leqslant I)$ is of length N and the label for each time-series instance is a nominal variable $Y \in \{0, \cdots, C\}^I$. The objective is to learn K shapelets $\mathbf{S}$, each of length L, that are most discriminative in order to characterize the target class, and their order information. The shapelets are denoted as $\mathbf{S} \in \mathbb{R}^{K \times L}$. Next, some key terminologies are introduced that are used in this paper.

Definition 6. *Minimum distance* (M_i^k): The minimum distance M_i^k between the time-series $\mathbf{T}_i$ and a candidate shapelet S^k is the minimum distance between the candidate shapelet and any segment of length L extracted from $\mathbf{T}_i$, that is,

$$M_i^k = \min_{j=1,\ldots,J} \frac{1}{L} \sum_{l=1}^{L} (\mathbf{T}_{i,j+l-1} - S_l^k)^2 \tag{5.1}$$

where, $\mathbf{T}_{i,j+l-1}$ represents $(j+l-1)^{th}$ value in the instance $\mathbf{T}_i$ and S_l^k represents the l^{th} value in the candidate shapelet S^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 7. *Shapelet-transform space* (**M**): The minimum distance between candidate shapelets S^k and the time-series $\mathbf{T}_i$ indicates the degree of closeness between a candidate shapelet S^k and the time-series $\mathbf{T}_i$ examples. Given a set of I time-series examples and K shapelets, a matrix $\mathbf{M} \in \mathbb{R}^{I \times K}$ can be constructed which is composed of minimum distances M_i^k between the i^{th} series $\mathbf{T}_i$ and the k^{th} shapelet S^k. This representation is known as shapelet-transformed data (Hills et al., 2014). The representation **M** (shown in Eq. 5.2) reduces the dimensionality of the original time-series since the number of candidate shapelets K is less than the length of the time-series N.

$$\mathbf{M} = \begin{array}{c} \begin{matrix} S^1 & S^2 & \ldots & S^K \end{matrix} \\ \begin{pmatrix} M_1^1 & M_1^2 & \ldots & M_1^K \\ \vdots & M_2^2 & \ldots & M_2^K \\ \vdots & \vdots & \ddots & \vdots \\ M_I^1 & \ldots & \ldots & M_I^K \end{pmatrix} \end{array} \tag{5.2}$$

Definition 8. *Start-time* (B_i^k): The start-time of a candidate shapelet S^k in $\mathbf{T}_i$ is the point j from which the candidate shapelet S^k has the minimum distance to $\mathbf{T}_i$, that is, $B_i^k = \text{argmin}_j \frac{1}{L} \sum_{l=1}^{L} (\mathbf{T}_{i,j+l-1} - S_l^k)^2$. The matrix **B** is a matrix of start-times B_i^k where shapelet S^k has minimum distance with time-series example $\mathbf{T}_i$

$$\mathbf{B} = \begin{array}{c} \begin{array}{cccc} S^1 & S^2 & \dots & S^K \end{array} \\ \begin{pmatrix} B_1^1 & B_1^2 & \dots & B_1^K \\ \vdots & B_2^2 & \dots & B_2^K \\ \vdots & \vdots & \ddots & \vdots \\ B_I^1 & \dots & \dots & B_I^K \end{pmatrix} \end{array} \tag{5.3}$$

Definition 9. *Time-gap* $G_i(S^{k_1}, S^{k_2})$: Given two shapelets S^{k_1} and S^{k_2} and a time-series $\mathbf{T}_i$, the time-gap between two shapelets is the difference of start-times of the two shapelets in the $\mathbf{T}_i$ normalized by the length of the time-series, that is, $G_i(S^{k_1}, S^{k_2}) = \frac{1}{N}(B_i^{k_1} - B_i^{k_2})$.

Definition 10. *Shapelet-order space* (**G**): Given K shapelets and I time-series examples, the shapelet-order space $\mathbf{G} \in \mathbb{R}^{I \times \frac{K(K-1)}{2}}$ represents the time-gaps among K pairwise shapelets. K pairwise time-gaps results in $\frac{K(K-1)}{2}$ shapelet-orders which capture the temporal interaction among the K candidate shapelets.

$$\mathbf{G} = \begin{array}{c} \begin{array}{cccc} O^1 & O^2 & \dots & O^{\frac{K(K-1)}{2}} \end{array} \\ \begin{pmatrix} G_1(S^1, S^2) & G_1(S^1, S^3) & \dots & G_1(S^{K-1}, S^K) \\ \vdots & G_2(S^1, S^3) & \dots & G_2(S^{K-1}, S^K) \\ \vdots & \vdots & \ddots & \vdots \\ G_I(S^1, S^2) & \dots & \dots & G_I(S^{K-1}, S^K) \end{pmatrix} \end{array} \tag{5.4}$$

5.5 Model Description

The details of the ***L**earning pairwise **O**rders* and **S**hapelet (LOS) model is introduced. In the LOS model, the probability of classifying a time-series instance to a particular class category is estimated using a linear function which is a linear combination of two different spaces, shapelet-transformed space and shapelet-order space. First, the linear learning function is discussed, followed by the definitions of gradients to learn the generalized shapelets and the parameters for the classification model. Furthermore, a fast temporal subsequence initialization procedure is introduced for learning generalized shapelets. Finally, an alternative model is presented that extends the predictive function with a weighted linear combinations of the two feature spaces.

5.5.1 Learning Objective

A linear learning function is proposed (Eq. 5.5) where the predicted target value $\hat{Y_i}$ is estimated jointly from the shapelet-transform space and the shapelet-order space. For ease of explanation the learning model is described using binary targets ($Y \in \{0, 1\}$) with the fixed shapelet length of L. The minimum distances $M_i^k \in \mathbf{M}$ and the time-gaps $G_i^h \in \mathbf{G}$ are used as predictors for approximating the value of $\hat{Y_i}$.

$$\hat{Y_i} = \sum_{k=1}^{K} M_i^k W_M^k + \sum_{h=1}^{\frac{K(K-1)}{2}} G_i^h W_G^h, \tag{5.5}$$

where W_M^k and W_G^h are the linear weights (parameters) to be learned for determining the classification hyperplane. A logistic regression framework is leveraged to learn the parameters of the learning function. The logistic loss function $\mathcal{L}(Y, \hat{Y}) = -Y ln\sigma(\hat{Y}) - (1-Y)ln(1-\sigma(\hat{Y}))$ is used to estimate the loss between the true targets Y and the predicted targets $\hat{Y}$. A regularized logistic loss function denoted by $\mathcal{F}$ is setup as the optimization function and is defined as:

$$\underset{S,W_M,W_G}{\text{argmin}}\, \mathcal{F}(S, W_M, W_G) = \underset{S,W_M,W_G}{\text{argmin}} \sum_{i=1}^{I} \mathcal{L}(Y_i, \hat{Y_i}) + \lambda_{W_M} \|W_M\|^2 + \lambda_{W_G} \|W_G\|^2 \tag{5.6}$$

The objective of the learning algorithm is to learn generalized shapelets S^k, the weights W_M, W_G for the hyperplane. Once generalized shapelets are updated, their corresponding shapelet-order space **G** are updated as well. A Stochastic Gradient Descent (henceforth SGD) approach is adopted to solve the optimization problem. The SGD algorithm optimizes the parameters to minimize the loss function by updating through per instance of the training data. Thus, the per-instance decomposed objective function is denoted as $\mathcal{F}_i = \mathcal{L}(y_i, \hat{y}_i) + \frac{\lambda_{W_M}}{I}\sum_{k=1}^{K}(w_M^k)^2 + \frac{\lambda_{W_G}}{I}\sum_{h=1}^{\frac{K(K-1)}{2}}(w_G^h)^2$.

Gradients

The SGD algorithm requires definitions of gradients of the objective function with respect to shapelets S^k, hyperplane weights W_M and W_G. However, according to Def. 5.1 M_i^k is not a continuous function and thus non-differentiable. Instead, M_i^k can be approximated using its soft-minimum approximation which is defined as

$$M_i^k \approx \hat{M}_i^k = \frac{\sum_{j=1}^{J} D_{i,j}^k \exp(\alpha D_{i,j}^k)}{\sum_{\bar{j}=1}^{J} \exp(\alpha D_{i,\bar{j}}^k)} \tag{5.7}$$

where $D_{i,j}^k$ is defined as the distance between the j^{th} segment of series i and the k^{th} shapelet given by the formula $D_{i,j}^k = \frac{1}{L}\sum_{l=1}^{L}(\mathbf{T}_{i,j+l-1} - S_l^k)^2$. Similarly, the start-time B_i^k defined in Def. 8 is also not differentiable. However, the soft-minimum version of B_i^k expressed as

$$B_i^k \approx \hat{B}_i^k = \frac{\sum_{j=1}^{J} j \exp(\alpha D_{i,j}^k)}{\sum_{\bar{j}=1}^{J} \exp(\alpha D_{i,\bar{j}}^k)} \tag{5.8}$$

can be used instead.

Next, the gradient definitions required by the model to solve the optimization problem are introduced. The point gradient of the objective function for the i^{th} time-series with respect to shapelet S^k at point l is defined as

$$\frac{\partial \mathcal{F}_i}{\partial S_l^k} = \frac{\partial \mathcal{L}(Y_i, \hat{Y}_i)}{\partial \hat{Y}_i}\frac{\partial \hat{Y}_i}{\partial \hat{M}_i^k}\sum_{j=1}^{J}\frac{\partial \hat{M}_i^k}{\partial D_{i,j}^k}\frac{\partial D_{i,j}^k}{\partial S_l^k}. \tag{5.9}$$

The derived expressions for the partial derivatives of each component in Eq. 5.9 are defined

Algorithm 7 ***Learning pairwise* O*rders and* S*hapelets***

```
procedure LOS
Input: $T \in \mathcal{R}^{I \times N}$, Number of shapelets $K$, length of a shapelet $L$, Regularization parameter $\lambda_{W_M}$,$\lambda_{W_G}$, Learning rate $\eta$, maxIter
Initialize: Shapelets $S \in \mathbb{R}^{K \times L}$, classification hyperplane weights $W_M \in \mathbb{R}^{K}$, $W_G \in \mathbb{R}^{\frac{K(K-1)}{2}}$,
  for iterations = $\mathbb{N}_1^{maxIter}$ do
    for $i = 1, ..., I$ do
      for $k = 1, ..., K$ do
        $W_M^{k^{new}} \leftarrow W_M^{k^{old}} - \eta \frac{\partial \mathcal{F}_i}{\partial W_M^k}$
        for $l = 1, ..., L$ do
          $S_l^{k^{new}} \leftarrow S_l^{k^{old}} - \eta \frac{\partial \mathcal{F}_i}{\partial S_l^k}$
        end for
      end for
      for $h = 1, ..., \frac{K(K-1)}{2}$ do
        $W_G^{k^{new}} \leftarrow W_G^{k^{old}} - \eta \frac{\partial \mathcal{F}_i}{\partial W_G^k}$
      end for
    end for
  end for
Return $S, W_M, W_G$
end procedure=0
```

as,

$$\frac{\partial \mathcal{L}(Y_i, \hat{Y}_i)}{\partial \hat{Y}_i} = -(Y_i - \sigma(\hat{Y}_i)), \qquad \frac{\partial \hat{Y}_i}{\partial \hat{M}_i^k} = W_k$$

$$\frac{\partial \hat{M}_i^k}{\partial D_{i,j}^k} = \frac{\exp(\alpha D_{i,j}^k (1 + \alpha (D_{i,j}^k - \hat{M}_i^k)))}{\sum_{\bar{j}=1}^{J} \exp\left(\alpha D_{i,\bar{j}}^k\right)}, \qquad \frac{\partial D_{i,k,j}}{\partial S_l^k} = \frac{2}{L}(S_l^k - \mathbf{T}_{i,j+l-1})$$

The hyperplane weights W_M and W_G are learned by minimizing the objective function in Eq. 5.6 via SGD. The gradients for updating the weights W_M^k and W_G^h is defined as $\frac{\partial \mathcal{F}_i}{\partial W_M^k} = -(Y_i - \sigma(\hat{Y}_i))\hat{M}_i^k + \frac{2\lambda_{W_M}}{I} W_M^k$ and $\frac{\partial \mathcal{F}_i}{\partial W_G^h} = -(Y_i - \sigma(\hat{Y}_i))\hat{G}_i^h + \frac{2\lambda_{W_G}}{I} W_G^h$ respectively.

The steps of the proposed learning algorithm for time-series classification are shown in Algorithm 7. The pseudocode shows that the procedure updates all K shapelets and the weights W_M, W_G by a learning rate η.

5.5.2 Random initialization vs. k-means clustering

Gradient based optimization techniques are heavily dependent on good initialization of parameters especially when applied to a non-convex learning functions. Traditional shapelet learning frameworks (Grabocka et al., 2014; Hou et al., 2016; Zhang et al., 2016) have applied k-means clustering algorithm to issue the k centroids of all time-series segments of a given length as the initial subsequences for learning generalized shapelets. However, k-means clustering is known to be time consuming method and does not scale well to large time-series datasets. In order to speed-up the subsequence initialization process, a random initialization process is proposed which consists of generating sine and cosine waveforms of a given length with the frequency component randomly chosen from a uniform distribution between 0 and 1. The choice of initialized subsequences for shapelet learning is justified since it has been established in the time-series research community that clustering time-series subsequences leads to a discovery of cluster centers patterns that are sine waves (Keogh and Lin, 2005).

5.5.3 Extended Model (LOS$_\gamma$)

In the proposed linear predictor function (Eq. 5.5), the shapelet-transform space encodes the similarity between the learned shapelets and the time-series instances, and the shapelet-order space provides the temporal dependency information among the learned pairwise generalized shapelets. Without prior knowledge on the contribution of the shapelet-transform space and the shapelet-order space, it is difficult to leverage the importance of the features from these two spaces. Thus, the proposed LOS model is extended with two additional weight parameters (γ_1 and γ_2) as shown in Eq. 5.10. The extended model is termed as LOS$_\gamma$.

$$\hat{Y}_i = \sum_{k=1}^{K} \gamma_1 M_i^k W_M^k + \sum_{h=1}^{\frac{K(K-1)}{2}} \gamma_2 G_i^h W_G^h \tag{5.10}$$

The parameters γ_1 and γ_2 are the weights from each of the feature spaces and control the amount of influence the predictors of each space has on the final outcome of the target variable. The learning problem is re-formulated as a constrained optimization problem since the weight parameters $\gamma \in \{\gamma_1, \gamma_2\}$ values should always be positive. Moreover, the values of the parameters can be forced to be fractional by enforcing the parameter constrain of $\gamma_1 + \gamma_2 = 1$. These constraints allows us to measure the contributions of both parameters towards the final prediction value of $\hat{Y}$. Learning these weights provides an alternative capability in understanding the influences of the shapelet-transform space and the shapelet-order space respectively. The constrained optimization function can be written as

$$\begin{gathered}\operatorname*{argmin}_{S,W,\gamma} \mathcal{F}(S, W^M, W^G, \gamma) \\ \text{subject to } \gamma_1 + \gamma_2 = 1, \gamma_1 > 0, \gamma_2 > 0.\end{gathered} \tag{5.11}$$

The previous gradient definitions also gets updated due to this new re-formalization as

$$\frac{\partial \hat{Y}_i}{\partial \hat{M}_i^k} = \gamma_1 W_k, \qquad \frac{\partial \mathcal{F}_i}{\partial W_M^k} = -\gamma_1 (Y_i - \sigma(\hat{Y}_i))\hat{M}_i^k + \frac{2\lambda_{W_M}}{I} W_M^k,$$

$$\text{and } \frac{\partial \mathcal{F}_i}{\partial W_G^h} = -\gamma_2 (Y_i - \sigma(\hat{Y}_i))\hat{G}_i^h + \frac{2\lambda_{W_G}}{I} W_G^h.$$

The learning procedure for estimating the space contribution parameters in the proposed framework is a constrained optimization problem because we need to guarantee that $\gamma_1 > 0$, $\gamma_2 > 0$ and $\gamma_1 + \gamma_2 = 1$. However, the SGD algorithm can only be applied to solve unconstrained optimization problems. Thus, we convert the constrained optimization into an unconstrained optimization similar to (Radosavljevic et al., 2010) and apply the SGD algorithm to solve the optimization problem for learning the optimal space weight parameters. The γ_2 parameter is first written in terms of γ_1 ($\gamma_2 = 1 - \gamma_1$) and replaced in equation (5.11) changing the optimization problem to Eq. 5.12.

$$\underset{S,W,\gamma_1}{\operatorname{argmin}} \mathcal{F}(S, W^M, W^G, \gamma_1) \qquad \text{subject to} \quad \gamma_1 > 0 \tag{5.12}$$

The objective function is then minimized with respect to $\log \gamma_1$ instead of γ_1. As a result, the new optimization problem becomes unconstrained. The derivative of the new objective function with respect to $\log \gamma_1$ in gradient descent is computed as: $\frac{\partial \mathcal{F}_i}{\partial \log \gamma_1} = \gamma_1 \frac{\partial \mathcal{L}(Y_i, \hat{Y}_i)}{\partial \gamma_1}$, where $\frac{\partial \mathcal{L}(Y_i, \hat{Y}_i)}{\partial \gamma_1} = \sum_{k=1}^{K} M_i^k W_M^k - \sum_{h=1}^{\frac{K(K-1)}{2}} G_i^h W_G^h$.

5.6 Experimental Evaluation

The proposed model was evaluated extensively on 14 real-world datasets obtained from the UEA & UCR Time Series Classification Repository (Bagnall et al., 2017). Details about each dataset can be viewed on the repository's website[1]. Additionally, three synthetic datasets were used to highlight the advantage of leveraging shapelet-orders over shapelet information.

5.6.1 Baseline methods

The performance of the proposed LOS model was compared with the following baseline models.

- Fast Shapelets (FS) (Rakthanmanon and Keogh, 2013): This is an extension of the decision tree shapelet approach that speeds up the shapelet discovery task. The FS algorithm discretize and approximates the shapelets rather than a complete search at each node of the decision tree.
- Scalable Shapelet Discovery (SSD) (Grabocka et al., 2016): This method is a fast procedure to extract random shapelets from the time-series dataset.
- Learning Time-Series Shapelets (LTS) (Grabocka et al., 2014): The LTS learns shapelets instead of searching from the time-series data.

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

- Pairwise Shapelet Order Discovery (PSOD) (Roychoudhury et al., 2019): The PSOD method generalizes SSD by taking pairwise shapelet-orders into account.

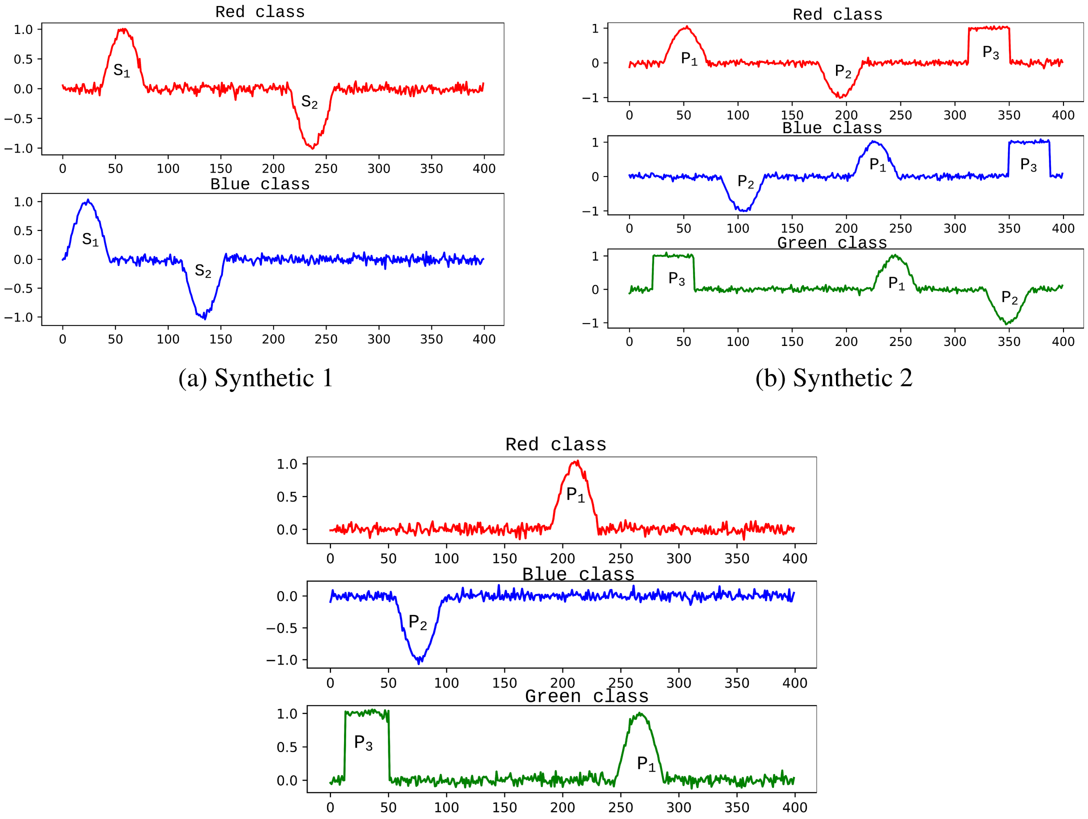
In addition, three variants of the proposed model is considered.

- LTS$_{+o}$: The LTS$_{+o}$ model simply extends the LTS model. It first applies LTS to learn generalized shapelets, and then learn the weights of shapelets and the weights of shapelet-orders of the learned shapelets through a logistic regression framework.
- LOS$_{kmeans}$: The shapelets were initialized using k-means centroids of all subsequences.
- LOS$_{\gamma}$: Extending LOS by modeling weight parameters controlling the contribution from two feature spaces.

5.6.2 Experimental setup

For each method the average accuracy of 10 iterations was recorded. The default training and test sets were used in the first iteration in all experiments. Then the train and test sets were re-sampled in the following iterations. Each re-sample was same for each algorithm and the re-samples were forced to retain the initial train and test sizes. Moreover, the re-samples were stratified to retain the same class distribution as the default train and test sets.

The minimum length (L_{min}) of a shapelet is set to be 10 $\%$ of the length of the time-series examples. The $total\ number\ of\ segments$ was computed as L_{min} multiplied by the number of training time-series. The number of shapelets used as input for the optimization function was determined using $K = log(total\,number\,of\,segments)$. Three scales $\{L_{min}, 2 \times L_{min}, 3 \times L_{min}\}$ of subsequence lengths were investigated. The weight parameters W_M and W_G were initialized randomly around 0. γ_1 was randomly picked from a uniform distribution between 0 and 1. The parameter α, for the soft-max approximation of minimum distances and start-times was set to -100.



(a) Synthetic 1

(b) Synthetic 2

(c) Synthetic 3

FIGURE 5.2: Three synthetic datasets were used to highlight the advantage of leveraging shapelet-orders over shapelet information. Fig. 5.2a and Fig. 5.2b shows a binary class and multi-class dataset respectively where the order among shapelets exist and differentiate the class of the time-series examples. Fig. 5.2c shows a multi-class synthetic dataset where all the time-series examples can be identified by a unique pattern.

5.6.3 Synthetic datasets

Two groups of synthetic time-series datasets were generated. In the first group, two sets of synthetic datasets were generated where the orders of shapelets are different among different classes of time-series. In the first dataset, (Fig. 5.2a) binary class synthetic time-series examples were generated where two subsequences with a specific pattern were considered. The first pattern is $y = \sin x, x \in [0, \pi]$ and is denoted as S_1, and the second pattern is $y = -\sin x, x \in [0, \pi]$ and denoted as S_2. In this dataset, S_2 always occurs more than 100 time points after the occurrence of S_1 in the red class, whereas in the blue class

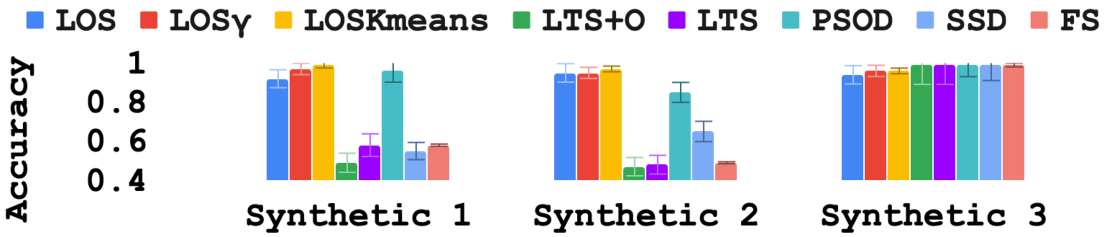


FIGURE 5.3: Average accuracy of synthetic time-series datasets.

the time difference between the two patterns is always less than 100. In the second dataset (Fig. 5.2b), three patterns P_1, P_2 and P_3 were generated. A multi-class time-series dataset was generated where in the red class P_1 always occurred before P_2 and P_3 always occurred after P_2. In a similar fashion for the blue class in Fig. 5.2b pattern P_2 always occurs before pattern P_1 and P_3 always occurs after pattern P_1. In case of the green class time-series pattern P_3 always occurs before P_1 which is followed by pattern P_2. In the second group, a multi-class synthetic time-series dataset was generated. The red class only contained P_1, the blue class only contained P_2 and the green class had both P_1 and P_3 although the order among P_1 and P_3 did not matter and both patterns were randomly placed. In all synthetic datasets, the start-time of patterns were randomly selected, and the remaining points in the time-series follow Gaussian distribution $\mathcal{N}(0, 0.05)$. Additionally, noise sampled from a product distribution comprised of $\mathcal{N}(0, 0.05)$ and $\mathcal{U}(0, 0.25)$ was added 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.

5.6.4 Results on synthetic datasets

Fig. 5.3 shows the average classification accuracy obtained by all baselines, LOS and all LOS variants on the synthetic datasets where shapelets have different orders. LOS, LOS$_\gamma$ and LOS$_{Kmeans}$ have significantly outperformed all baselines in terms of classification accuracy. PSOD attains a high average accuracy of 0.96 on Synthetic 1 dataset however, due to the random selection of candidate shapelets, PSOD looses performance in the more complicated Synthetic 2 dataset (average accuracy of 0.85) where three unique patterns are present in the data and examples of each class have a combination of temporal dependency among the three shapelets. LOS, LOS$_\gamma$ and LOS$_{Kmeans}$ on the other hand can handle the more complicated scenario in Synthetic 2 dataset to attain an a high average accuracy between $0.96 - 0.99$. The LTS$_{+o}$ model learns shapelets and considers shapelet-order as well. However, it executes in a two separate steps. The accuracy obtained by LTS$_{+o}$ is a near random guess. This verifies the superiority of LOS, which jointly learns shapelets and orders. As expected, the accuracy obtained by FS, SSD and LTS on Synthetic 1 and Synthetic 2 datasets are random, as there is no unique subsequence that can differentiate the categories of the time-series examples.

All baselines and the variants of LOS model perform comparable on the Synthetic 3 datasets where there is no order among shapelets (as shown in Fig. 5.2c). Therefore, the benefit of taking shapelet-orders into account was more evident when temporal dependency among pairs of shapelets could differentiate the class. Also leveraging the shapelet-orders among learned shapelets allows us to improve the classification accuracy due to the extraction of shapelet-orders among generalized shapelets.

5.6.5 Results on real-world datasets

The performance of LOS was compared against all the baseline methods on 14 real-world datasets that have a variety of properties in terms of time-series length and the number of classes. The average classification accuracy is reported in Table 5.1. The first group of datasets are binary class datasets. LOS attained the better or comparable accuracy in 4 out 6 binary class datasets when compared to baseline models. Although, on the *Wine* dataset FS is clearly a winner, LOS attains better performance against SSD, LTS and PSOD. On the second group of datasets, which have multiple classes, LOS$_{Kmeans}$ either outperformed or attained comparable performance as the baselines with respect to accuracy.

Table 5.1: Average accuracy on real-world time-series datasets.

Dataset	N	C	LOS	LOS_γ	LOS_{KMeans}	LTS_{+O}	LTS	$PSOD$	SSD	FS
Binary class datasets										
DistPOutCo.	80	2	**0.79**	0.75	0.75	0.77	0.77	0.68	0.73	0.72
PowerCons	144	2	**0.88**	0.85	0.82	0.88	0.87	0.86	0.83	0.83
GunPoiMVsF	150	2	**0.99**	0.98	0.98	0.98	0.98	0.97	0.96	0.94
Wafer	152	2	**0.99**	0.97	0.97	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**
Wine	234	2	0.59	0.57	0.52	0.54	0.52	0.50	0.50	**0.75**
Worm2Class	900	2	0.68	0.60	0.68	0.71	**0.75**	0.57	0.65	0.68
Multi-class datasets										
DistPXTW	80	6	0.64	0.64	**0.65**	0.62	**0.65**	**0.65**	0.57	0.62
2Patterns	128	4	0.96	0.96	**0.97**	0.70	**0.97**	0.96	**0.97**	0.92
ECG5000	140	5	0.92	0.92	**0.94**	0.90	0.92	0.93	0.92	0.92
Plane	144	7	0.98	0.98	**0.99**	0.94	**0.99**	0.98	0.96	**0.99**
Fungi	201	18	0.69	0.79	**0.85**	0.63	0.55	0.33	0.28	0.56
Meat	448	3	0.91	0.90	**0.92**	0.73	0.89	0.91	0.91	0.83
Beef	470	5	0.57	0.55	**0.59**	0.48	0.57	0.55	0.5	0.56
OliveOil	570	4	0.66	0.53	**0.73**	0.41	0.41	0.44	0.65	**0.73**

5.6.6 Training time analysis

The proposed LOS and LOS$_\gamma$ not only improves accuracy, but also reduces the amount of time it requires to train the model, thus allowing the proposed models to scale to large time-series datasets. We first analyzed how the accuracy and running time of these two models vary compared to LOS$_{Kmeans}$, LTS$_{+o}$ and LTS with respect to increasing the number of shapelets (K) on Synthetic 1 dataset. In Fig. 5.4, the accuracy of five methods was plotted as the number of shapelets (K) is increased and also the training time (Fig. 5.5). We observed that at the beginning, the accuracy of LOS (blue line), LOS$_\gamma$ (red line) and LOS$_{Kmeans}$ (yellow line) increases as the number of shapelets learned in the model is increased, and then converges to a stable value. However, with the increasing number of shapelets the performance of LTS$_{+o}$ (green line) and LTS (purple line) in terms classification accuracy do not improve. The primary reason for this is that Synthetic 1 dataset has order in the time-series (see Fig. 5.2a). Moreover, increasing K only slightly increases the running time for LOS (blue line) and LOS$_\gamma$ (red line) in Fig. 5.5. LOS$_{Kmeans}$, LTS$_{+o}$ and LTS that use the k-means clustering algorithm for the initialization

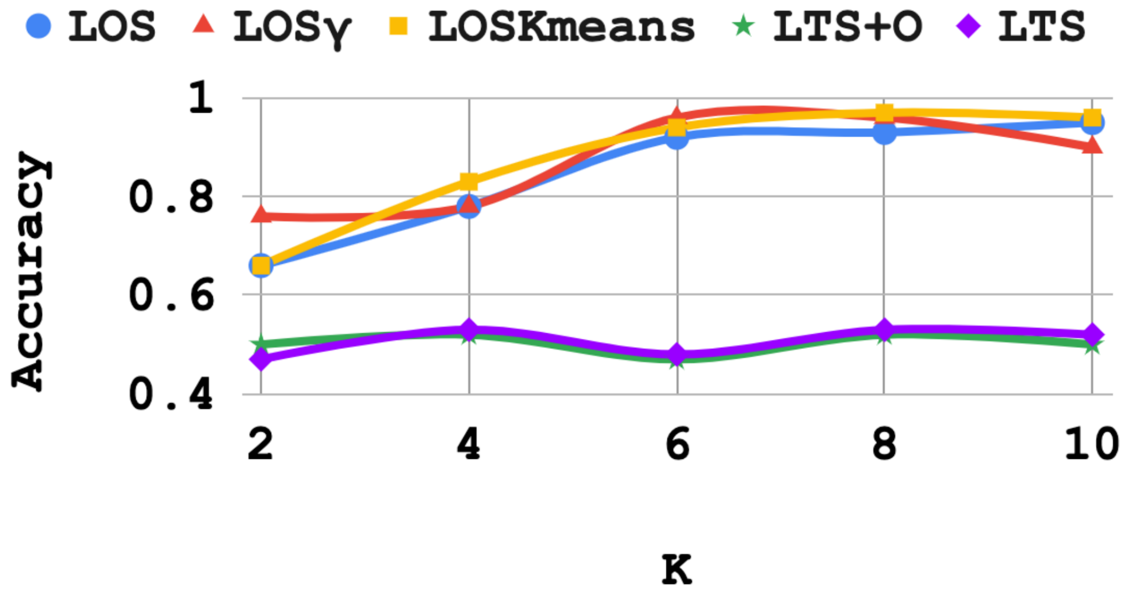


FIGURE 5.4: Analysis of Test set accuracy on Synthetic 1 dataset with increasing K.

of subsequences increase the training time by 4 folds for every value of K and becomes costlier with increasing K value.

In Fig. **??** the training time vs. test accuracy is compared on three real time-series datasets of different lengths. The blue bar indicates the test accuracy and the black dot represents the running time for a model. The results indicate that the proposed LOS and LOS$_\gamma$ attains similar or better accuracy with faster training time when compared to LOS$_{Kmeans}$, LTS and LTS$_{+o}$. For example, in the *Worm2Class* dataset (Fig. 5.6), both LOS and LOS$_{Kmeans}$ achieved an accuracy of 0.68 however, the running time of LOS was 14 minutes compared to 30 minutes for LOS$_{Kmeans}$. For the $Beef$ dataset (Fig. 5.8), LOS$_{Kmeans}$ achieved an accuracy of 0.59 with a run-time of 12.5 minutes, whereas, LOS achieved a comparable accuracy of 0.57 with reduced run-time of 8.8 minutes. LOS and LOS$_{+o}$ always take more time than LOS. Table 5.1, Fig. 5.6, Fig. 5.7 and Fig. 5.8 together show that LOS can achieve the best (or the second best) accuracy with the smallest training time.

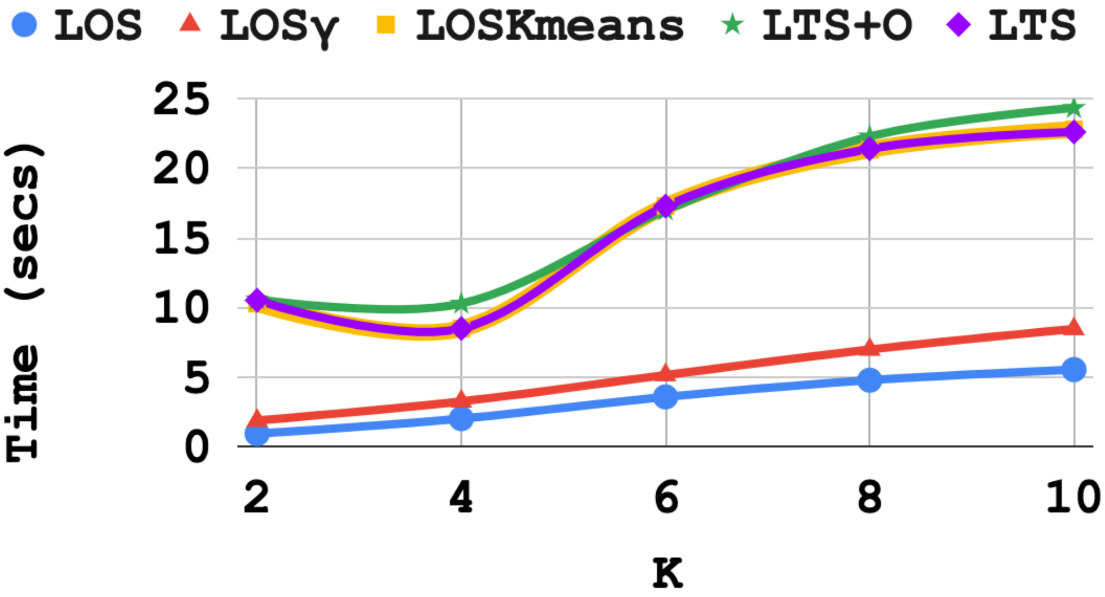


FIGURE 5.5: Analysis of Training time on Synthetic 1 dataset with increasing K.

5.7 Conclusion

A novel perspective of jointly learning generalized shapelets and leveraging shapelet-orders that capture the temporal dependency among pairwise learned shapelets is proposed for time-series classification. The shapelet-transformed space is augmented with the shapelet-order space, and both shapelets and shapelet-orders are jointly learned from the data. Many applications account for class discrimination via the temporal dependency of shapelets as discussed in the paper. From the experimental results on the synthetic datasets, we found that the LOS and its variants produce more accurate classification results compared to PSOD which is the state-of-the-art method for leveraging shapelet-orders in the shapelet based time-series classification models. Moreover, we significantly improved the scalability of the model by proposing a random initializing technique. Experiments on the real datasets show improved or comparable average classification accuracy when compared to baseline methods. In the future, we plan to extend the proposed learning framework for multivariate time-series datasets in order to capture shapelet-orders across

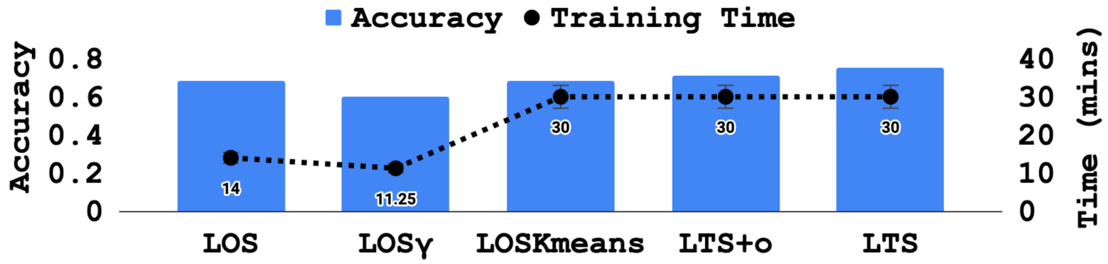


FIGURE 5.6: Accuracy vs. training time comparison for Worm2class dataset.

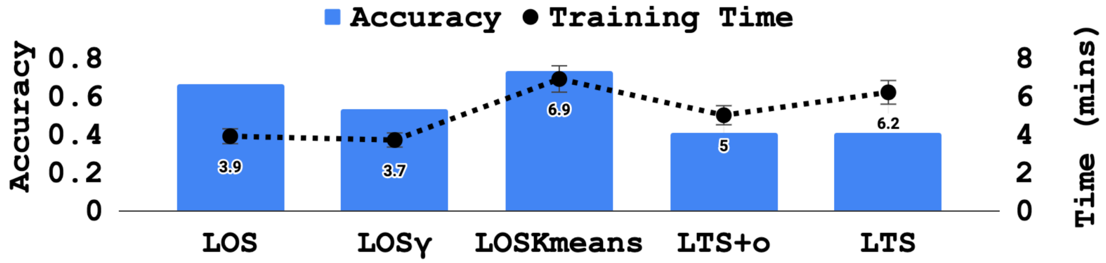


FIGURE 5.7: Accuracy vs. training time comparison for OliveOil dataset.

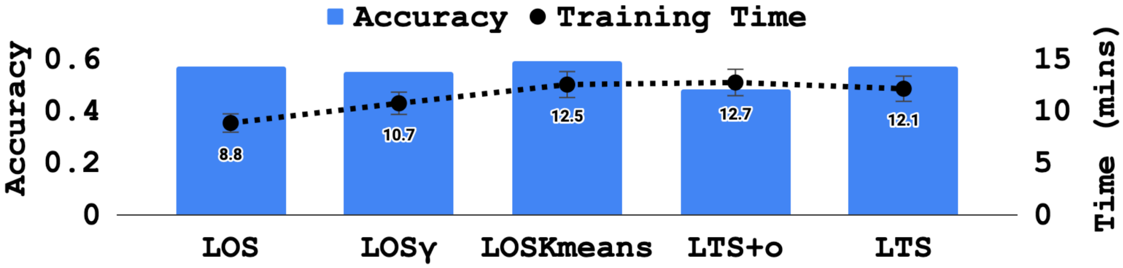


FIGURE 5.8: Accuracy vs. training time comparison for Beef dataset.

multiple dimensions.

CHAPTER 6

CONCLUSION

In this dissertation, four different time-series classification models are proposed to solve two real-world challenges pertaining to shapelet based time-series classification frameworks. Two proposed models provide solutions to the cost-sensitive learning problem (Chapter 2 and Chapter 3 in highly imbalanced time-series datasets. The extraction of temporal dependency information among subsequences and leveraging such information for time-series classification is explored in Chapter 4 and Chapter 5. The proposed models are supervised learning problems that focus on improving the current state-of-the-art methods on challenges related to real-world time-series datasets. In each chapter, we present a structured report that includes:

- a detailed background to motivate the target problem and lead to the main idea of the proposed approach;
- a comprehensive related work section listing the major state-of-the-art approaches of the given problem;
- a thorough method description to provide the technical details of the method and algorithm;
- a rigorous experimental evaluation to analyze the advantages and disadvantage of the

proposed approach.

In Chapter 2 an uncertainty based cost-sensitive early time-series classification model is proposed to suppress false and detect true cardiac arrhythmia alarms from ECG signals. Experiments on two life threatening cardiac arrhythmia datasets from Physionet's MIMIC II repository provide evidence that the proposed method is capable of identifying patterns that can distinguish false and true alarms using on average 60% of the available time series' length. Using temporal uncertainty estimates of time series predictions, confidence was estimated in the early classification predictions, therefore providing a cost-sensitive prediction model for ECG signal classification. The results from the proposed method are interpretable, providing medical personnel a visual verification of the predicted results. In conducted experiments, moderate false alarm suppression rates were achieved (34.29% for Asystole and 20.32% for Ventricular Tachycardia) while keeping near 100% true alarm detection, outperforming the state-of-the-art methods, which compromise true alarm detection rate for higher false alarm suppression rate, on these challenging applications.

In Chapter 3 a cost-sensitive time-series classification learning framework is proposed by extending the generalized shapelet learning framework to handle highly imbalance time-series datasets. First, the effectiveness of the proposed method is demonstrated on two case studies from the previous chapter, with the objective to detect true alarms from life threatening cardiac arrhythmia dataset from Physionet's MIMIC II repository. The results show improved true alarm detection rates over the current state-of-the-art method. Additionally, the proposed method is compared to the state-of-the-art learning shapelet method on 16 balanced dataset from the UCR time-series repository. The results show evidence that the proposed method (CS-LTS) outperforms the state-of-the-art method. Finally, extensive experiments were performed across an additional 18 highly imbalanced time-series datasets. The results provided evidence that the proposed method achieved comparable results with the state-of-the-art sampling/non-sampling based approaches for highly imbalanced time-series datasets. However, the proposed CS-LTS method is highly interpretable which is an advantage over many other methods.

In Chapter 4 and Chapter 5, the temporal dependencies among shapelets are explored. Two models are proposed, Pairwise Shapelet-Orders Discovery (PSOD) and Learning pairwise Orders and Shapelets (LOS), which extracts both informative shapelets and shapelet-orders and incorporates the shapelet-transformed space with shapelet-order space for time-series classification. The two proposed models are contrasting approaches in the time-series classification paradigm. The PSOD is a search-based greedy procedure to extract unique shapelets and identify orders among the selected shapelets. On the other hand, LOS is an optimization-based approach to extract shapelet-orders among learned generalized shapelets. However, in both the hypotheses, the extracted pairwise shapelet-orders could increase the confidence of the prediction and further improve the classification performance. In case of PSOD, 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. However, the drawback of a randomized candidate shapelet selection procedure is highlighted in Chapter 5 and a novel model, LOS, is proposed to alleviate the issue of non-optimal shapelet selection. The proposed LOS model was more accurate and faster than the baseline alternatives when evaluated on both synthetic and real-world time-series datasets.

BIBLIOGRAPHY

Aboukhalil, A., Nielsen, L., Saeed, M., Mark, R. G., and Clifford, G. D. (2008), "Reducing false alarm rates for critical arrhythmias using the arterial blood pressure waveform," *Journal of Biomedical Informatics*, 41, 442 – 451.

Bagnall, A., Davis, L., Hills, J., and Lines, J. (2012), "Transformation based ensembles for time series classification," in *Proceedings of the 2012 SIAM international conference on data mining*, pp. 307–318, SIAM.

Bagnall, A., Lines, J., Bostrom, A., Large, J., and Keogh, E. (2017), "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances," *Data Min. Knowl. Discov*, 31, 606–660.

Behar, J., Oster, J., Li, Q., and Clifford, G. (2013), "ECG signal quality during arrhythmia and its application to false alarm reduction," *IEEE Transactions on Biomedical Engineering*, 60, 1660–1666.

Bostrom, A. and Bagnall, A. (2017), "A Shapelet Transform for Multivariate Time Series Classification," *CoRR*, abs/1712.06428.

Cao, H., Li, X., Woon, D. Y., and Ng, S. (2011), "SPO: Structure Preserving Oversampling for Imbalanced Time Series Classification," in *11th IEEE International Conference on Data Mining, ICDM 2011, Vancouver, BC, Canada, December 11-14, 2011*, pp. 1008–1013.

Cao, H., Li, X., Woon, D. Y., and Ng, S. (2013), "Integrated Oversampling for Imbalanced Time Series Classification," *IEEE Trans. Knowl. Data Eng.*, 25, 2809–2822.

Cao, H., Tan, V. Y. F., and Pang, J. Z. F. (2014), "A Parsimonious Mixture of Gaussian Trees Model for Oversampling in Imbalanced and Multimodal Time-Series Classification," *IEEE Trans. Neural Netw. Learning Syst.*, 25, 2226–2239.

Cetin, M. S., Mueen, A., and Calhoun, V. D. (2015), "Shapelet Ensemble for Multi-dimensional Time Series." in *SDM*, pp. 307–315.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002), "SMOTE: Synthetic Minority Over-sampling Technique," *J. Artif. Int. Res.*, 16, 321–357.

Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A., and Batista, G. (2015), "The UCR Time Series Classification Archive," .

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

Das, G., Lin, K.-I., Mannila, H., Renganathan, G., and Smyth, P. (1998), "Rule Discovery from Time Series," in *SIGKDD*, pp. 16–22.

Demšar, J. (2006), "Statistical Comparisons of Classifiers over Multiple Data Sets," *Jour. of Machh. Learn. Res.*, 7, 1–30.

Drew, B. J., Harris, P., Zègre-Hemsey, J. K., Mammone, T., Schindler, D., Salas-Boni, R., Bai, Y., Tinoco, A., Ding, Q., and Hu, X. (2014), "Insights into the Problem of Alarm Fatigue with Physiologic Monitor Devices: A Comprehensive Observational Study of Consecutive Intensive Care Unit Patients," *PLoS ONE*, 9.

Ghalwash, M., Radosavljevic, V., and Obradovic, Z. (2013a), "Early Diagnosis and Its Benefits in Sepsis Blood Purification Treatment," in *IEEE International Conference on Healthcare Informatics, ICHI 2013, 9-11 September, 2013, Philadelphia, PA, USA*, pp. 523–528.

Ghalwash, M., Radosavljevic, V., and Obradovic, Z. (2013b), "Extraction of Interpretable Multivariate Patterns for Early Diagnostics," in *ICDM*, pp. 201–210.

Ghalwash, M. F., Radosavljevic, V., and Obradovic, Z. (2014), "Utilizing Temporal Patterns for Estimating Uncertainty in Interpretable Early Decision Making," in *SIGKDD*, pp. 402–411.

Goldberger, A. L., Amaral, L. A. N., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. (2000 (June 13)), "PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals," *Circulation*, 101, e215–e220.

Grabocka, J., Schilling, N., Wistuba, M., and Schmidt-Thieme, L. (2014), "Learning Time-series Shapelets," in *SIGKDD*, pp. 392–401.

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

Guo, H. and Viktor, H. L. (2004), "Learning from Imbalanced Data Sets with Boosting and Data Generation: The DataBoost-IM Approach," *SIGKDD Explor. Newsl.*, 6, 30–39.

Han, H., Wang, W.-Y., and Mao, B.-H. (2005), "Borderline-SMOTE: A New Over-sampling Method in Imbalanced Data Sets Learning," in *Proceedings of the 2005 International Conference on Advances in Intelligent Computing - Volume Part I*, ICIC'05, pp. 878–887.

He, H. and Garcia, E. A. (2009), “Learning from Imbalanced Data,” *IEEE Trans. on Knowl. and Data Eng.*, 21, 1263–1284.

He, H., Bai, Y., Garcia, E. A., and Li, S. (2008), “ADASYN: Adaptive synthetic sampling approach for imbalanced learning,” in *Proceedings of the International Joint Conference on Neural Networks, IJCNN 2008, part of the IEEE World Congress on Computational Intelligence, WCCI 2008, Hong Kong, China, June 1-6, 2008*, pp. 1322–1328.

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

Hou, L., Kwok, J. T., and Zurada, J. M. (2016), “Efficient Learning of Timeseries Shapelets,” in *AAAI*, pp. 1209–1215.

Jr., J. P. K. (2012), “Clinical alarm hazards: a “top ten” health technology safety concern,” *Journal of Electrocardiology*, 45, 588 – 591.

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

Keogh, E. and Lin, J. (2005), “Clustering of Time-Series Subsequences is Meaningless: Implications for Previous and Future Research,” *Knowl. Inf. Syst.*, 8, 154–177.

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

Li, Q. and Clifford, G. D. (2012), “Signal quality and data fusion for false alarm reduction in the intensive care unit,” *Journal of Electrocardiology*, 45, 596 – 603.

Lines, J., Davis, L. M., Hills, J., and Bagnall, A. (2012), “A Shapelet Transform for Time Series Classification,” in *SIGKDD*, pp. 289–297.

Liu, X.-Y., Wu, J., and Zhou, Z.-H. (2009), “Exploratory Undersampling for Class-imbalance Learning,” *Trans. Sys. Man Cyber. Part B*, 39, 539–550.

López, V., Fernández, A., García, S., Palade, V., and Herrera, F. (2013), “An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics,” *Information Sciences*, 250, 113 – 141.

Mirowski, T., Roychoudhury, S., Zhou, F., and Obradovic, Z. (2016), “Predicting Poll Trends Using Twitter and Multivariate Time-Series Classification,” in *SocInfo*, pp. 273–289, Springer.

Mueen, A., Keogh, E., and Young, N. (2011), “Logical-shapelets: An Expressive Primitive for Time Series Classification,” in *SIGKDD*, pp. 1154–1162.

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

Patri, O. P., Kannan, R., Panangadan, A. V., and Prasanna, V. (2015), “Multivariate Time Series Classification Using Inter-leaved Shapelets,” in *NIPS Time Series Workshop*.

Radosavljevic, V., Vucetic, S., and Obradovic, Z. (2010), “Continuous Conditional Random Fields for Regression in Remote Sensing,” in *Proceedings of the 2010 Conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, pp. 809–814, Amsterdam, The Netherlands, The Netherlands, IOS Press.

Rakthanmanon, T. and Keogh, E. (2013), “Fast Shapelets: A Scalable Algorithm for Discovering Time Series Shapelets,” in *SDM*, pp. 668–676.

Roychoudhury, S., Ghalwash, M. F., and Obradovic, Z. (2015), “False alarm suppression in early prediction of cardiac arrhythmia,” in *BIBE*, pp. 1–6.

Roychoudhury, S., Ghalwash, M. F., and Obradovic, Z. (2017), “Cost Sensitive Time-Series Classification,” in *ECML/PKDD*, pp. 495–511.

Roychoudhury, S., Zhou, F., and Obradovic, Z. (2019), “Leveraging Subsequence-orders for Univariate and Multivariate Time-series Classification,” in *Proceedings of the 2019 SIAM International Conference on Data Mining*, pp. 495–503, SIAM.

Saeed, M., Villarroel, M., Reisner, A., Clifford, G., Lehman, L.-W., Moody, G., Heldt, T., Kyaw, T., Moody, B., and Mark, R. (2011), “Multiparameter intelligent monitoring in intensive care II: A public-access intensive care unit database,” *Critical Care Medicine*, 39, 952–960.

Salas-Boni, R., Bai, Y., Harris, P. R. E., Drew, B. J., and Hu, X. (2014), “False ventricular tachycardia alarm suppression in the ICU based on the discrete wavelet transform in the ECG signal,” *Journal of Electrocardiology*, 47, 775 – 780.

Sayadi, O. and Shamsollahi, M. B. (2011), “Life-Threatening Arrhythmia Verification in ICU Patients Using the Joint Cardiovascular Dynamical Model and a Bayesian Filter,” *IEEE Transactions on Biomedical Engineering*, 58, 2748–2757.

Scalzo, F., Liebeskind, D. S., and Hu, X. (2013), “Reducing False Intracranial Pressure Alarms Using Morphological Waveform Features.” *IEEE Trans. Biomedical Engineering*, 60, 235–239.

Shokoohi-Yekta, M., Chen, Y., Campana, B., Hu, B., Zakaria, J., and Keogh, E. (2015), “Discovery of Meaningful Rules in Time Series,” in *SIGKDD*, pp. 1085–1094.

Sun, Y., Kamel, M. S., Wong, A. K. C., and Wang, Y. (2007), “Cost-sensitive Boosting for Classification of Imbalanced Data,” *Pattern Recogn.*, 40, 3358–3378.

Tatavarty, G., Bhatnagar, R., and Young, B. (2007), "Discovery of Temporal Dependencies between Frequent Patterns in Multivariate Time Series," in *2007 IEEE Symp. on CIDM*, pp. 688–696.

Ting, J., Fu, T.-C., and Chung, K. F.-L. (2006), "Mining of Stock Data: Intra- and Inter-Stock Pattern Associative Classification," in *DMIN*.

Xing, Z., Pei, J., and Yu, P. S. (2009), "Early Prediction on Time Series: A Nearest Neighbor Approach." in *International Joint Conferences on Artificial Intelligence*, pp. 1297–1302.

Xing, Z., Pei, J., Yu, P. S., and Wang, K. (2011), "Extracting Interpretable Features for Early Classification on Time Series." in *SDM*, pp. 247–258.

Ye, L. and Keogh, E. (2009a), "Time Series Shapelets: A New Primitive for Data Mining," in *SIGKDD*, pp. 947–956.

Ye, L. and Keogh, E. (2009b), "Time series shapelets: a new primitive for data mining," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 947–956, ACM.

Zakaria, J., Mueen, A., and Keogh, E. (2012), "Clustering Time Series Using Unsupervised-Shapelets," in *ICDM*, pp. 785–794.

Zhang, Q., Wu, J., Yang, H., Tian, Y., and Zhang, C. (2016), "Unsupervised Feature Learning from Time Series," in *IJCAI*, pp. 2322–2328.

Appendix A

PRECISION OF THE CANDIDATE-ORDER

The precision of the candidate order can be computed as follows

$$P(Class = c | o \text{ exists}) = \frac{P(o \text{ exists} | Class = c)\ P(Class = c)}{P(o \text{ exists})}, \tag{A.1}$$

where $P(o \text{ exists})$ is the probability that the candidate order o exists in the training data. $P(Class = c)$ is the probability of an instance belonging to class c, and $P(o \text{ exists} | Class = c)$ is the probability of an instance belonging to class c contains the order o. More specifically we compute the terms in Eq. A.1 in the following manner:

$$P(o \text{ exist}) = \frac{\text{\#training instances where } o \text{ occurs}}{\text{\#training instances}} \tag{A.2}$$

$$P(Class = c) = \frac{\text{\# training instances which labels are } c}{\text{\#training instances}} \tag{A.3}$$

$$P(o \text{ exists} | Class = c) = \frac{\text{\# training instances which labels are } c \text{ and order } o \text{ also occurs}}{\text{\# training instances which labels are } c} \tag{A.4}$$

The confidence of the candidate order o of class = c is defined as a product of the precision (Eq. A.1 of the candidate order and the probability of the intersection that the candidate order exists and belongs to class c, that is,

$$\mathbb{C}(o) = P(Class = c | o \text{ exists}) * P(Class = c \cap o \text{ exists}) \tag{A.5}$$

Both terms in Eq. A.5 are probabilities, thus the confidence measure for order o is a value between 0 and 1.

Appendix B

PSOD FRAMEWORK

The following figure is a diagrammatic representation of the proposed PSOD framework.

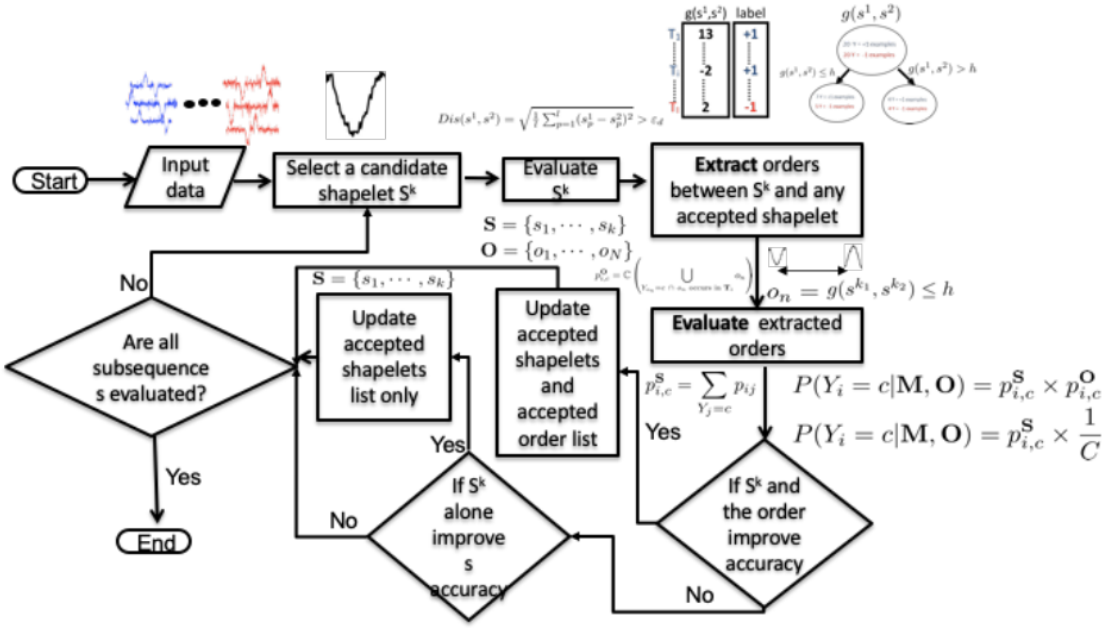

g(s¹,s²)
label
T₁
13
+1
Tᵢ
-2
+1
Tₗ
2
-1
$g(s^1,s^2)$
$g(s^1,s^2) \le h$
$g(s^1,s^2) > h$
$Dis(s^1,s^2) = \sqrt{\frac{1}{l}\sum_{p=1}^{l}(s_p^1 - s_p^2)^2} > \varepsilon_d$
Start
Input data
Select a candidate shapelet Sᵏ
Evaluate Sᵏ
Extract orders between Sᵏ and any accepted shapelet
$\mathbf{S} = \{s_1, \cdots, s_k\}$
$\mathbf{O} = \{o_1, \cdots, o_N\}$
$o_n = g(s^{k_1}, s^{k_2}) \le h$
Evaluate extracted orders
$p_{i,c}^{\mathbf{S}} = \sum_{Y_j = c} p_{ij}$
$P(Y_i = c|\mathbf{M}, \mathbf{O}) = p_{i,c}^{\mathbf{S}} \times p_{i,c}^{\mathbf{O}}$
$P(Y_i = c|\mathbf{M}, \mathbf{O}) = p_{i,c}^{\mathbf{S}} \times \frac{1}{C}$
No
Are all subsequence s evaluated?
Yes
End
$\mathbf{S} = \{s_1, \cdots, s_k\}$
Update accepted shapelets list only
Update accepted shapelets and accepted order list
Yes
No
If Sᵏ alone improve s accuracy
Yes
If Sᵏ and the order improve accuracy
No

FIGURE B.1: The framework of the proposed model PSOD (training phase).

Appendix C

DETAILED RESULTS ON 75 UNIVARIATE TIME-SERIES DATASETS

Table C.1 lists the classification accuracies of FS, SSD and PSOD on real-world univariate time-series datasets from 7 different categories. The summarized results of this table are shown in Fig. 4.8, Fig. 4.9, Fig 4.10 and Fig. 4.11 respectively in chapter 4.

Table C.1: Average classification accuracy for datasets from 7 different categories

Category	Dataset	FS	SSD	PSOD
ECG	CinCECG	**0.85**	0.53	0.55
	ECG200	0.75	**0.81**	**0.81**
	ECG5000	0.92	0.92	**0.94**
	ECGFiveDays	**0.99**	0.95	**0.99**
	TwoLeadECG	**0.92**	0.9	**0.92**
	50Worlds	0.48	**0.64**	0.62
	Adiac	0.54	**0.59**	0.3
	ArrowHead	0.57	0.7	**0.73**
	BeetleFly	0.65	0.71	**0.75**
	Birdchicken	**0.75**	0.66	0.63
	DiatomSizeRedu	0.87	**0.91**	0.74
	DistalPhalanxAG	0.64	0.67	**0.68**
	DistalPhalanxCorr	0.72	0.73	**0.74**
	DistalPhalanxTW	0.62	0.58	**0.66**
	FaceALL	0.62	**0.73**	**0.73**

Table C.1 continued from previous page

Category	Dataset	FS	SSD	PSOD
Image	FaceFour	**0.9**	0.75	0.8
	FacesUCR	0.7	0.85	**0.86**
	Fish	**0.77**	0.76	0.69
	handOutline	0.81	0.82	**0.86**
	Herring	0.53	0.51	**0.55**
	MedicalImages	0.6	0.68	**0.7**
	MiddlePhalanxoutlineAgeGrp	0.53	0.47	**0.61**
	MiddlePhjalanxoutlineCorrect	0.66	0.71	**0.73**
	MiddlePhalanxTW	0.46	0.44	**0.52**
	OsuLeaf	**0.67**	0.61	0.6
	PhallangedOutlineCorrect	0.72	0.73	**0.74**
	ProximalGroup	0.77	0.78	**0.83**
	proximalcorrect	**0.8**	0.72	0.78
	ProximalTW	0.7	0.71	**0.78**
	ShapesAll	0.58	**0.82**	0.79
	SwedishLeaf	0.6	**0.87**	0.85
	Symbols	**0.93**	0.81	0.86
	WordSynonyms	0.43	**0.59**	0.57
	Yoga	0.69	0.8	**0.73**
Sensor	Car	**0.73**	0.67	0.7
	Earthquake	0.71	0.68	**0.73**
	FordA	0.78	0.85	**0.88**
	FordB	0.72	0.77	**0.83**
	InsectWingbeat	0.47	0.45	**0.48**
	ItalyPowerDemand	**0.9**	0.82	0.86
	Lightning2	0.7	**0.8**	0.78
	Lightning7	0.63	0.65	**0.69**
	Motestain	0.77	0.77	**0.79**
	Phoneme	0.17	0.17	**0.2**
	Plane	**0.99**	0.96	0.98
	Sony1	0.68	0.77	**0.78**
	Sony2	0.79	0.85	**0.87**
	StarLightcurve	0.91	**0.94**	**0.94**
	Trace	**1**	0.98	0.99
	Wafer	**0.99**	**0.99**	**0.99**
Synthetic	shapeletSim	**1**	0.92	0.92
	synthetic controll	0.91	0.95	**0.97**
	Two Patterns	0.92	0.97	**0.99**
	ChlorineConcentraion	0.54	**0.57**	**0.57**

Table C.1 continued from previous page

Category	Dataset	FS	SSD	PSOD
Spectro	CBF	0.94	0.95	**0.98**
	Beef	0.56	0.5	**0.58**
	Ham	**0.64**	0.55	0.6
	Meat	0.83	**0.91**	**0.91**
	OliveOil	0.73	**0.76**	0.65
	Strawberry	0.9	0.9	**0.92**
	Wine	**0.75**	0.5	0.5
Motion	CricketX	0.48	0.72	**0.73**
	CricketY	0.53	0.69	**0.71**
	CricketZ	0.46	0.73	**0.76**
	GunPoint	**0.94**	**0.94**	**0.94**
	Haptics	0.39	0.32	**0.43**
	ToeSegm1	0.65	0.9	**0.92**
	ToeSegment2	0.69	0.87	**0.92**
	UWGestureX	0.69	0.73	**0.74**
	UWGestureY	0.59	0.64	**0.68**
	UWGestureZ	0.63	**0.67**	**0.67**
Device	Computers	0.5	**0.61**	**0.61**
	LargeKitchen	0.56	**0.76**	0.74
	RefrigenationDevice	0.33	0.5	**0.53**
	ScreenType	**0.4**	0.37	**0.4**
	SmallKitchenApplian	0.33	0.63	**0.72**