PROBABILISTIC DISCRIMINATIVE REGION DESCRIPTOR FOR TIME SERIES CLASSIFICATION

by

Pınar Süngü İşiaçık

B.S., Mathematics, Koç University, 2019

B.A., Economics, Koç University, 2019

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computational Science and Engineering

Boğaziçi University

2022

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my advisor *Mustafa Gökçe Baydoğan* for his patience, motivation, enthusiasm, and sharing of his immense knowledge. After every conversation we had, I came away enlightened and feeling fortunate for his guidance. Without his support, understanding and trust in every step throughout the process, this thesis would have never been possible.

I would like to express my gratitude for the valuable comments and constructive criticism I received from Mert Edalı and Ümit Işlak who were an important part of my thesis committee.

I would also like to thank the members of my Computational Science and Engineering department for sharing their profound knowledge and methodology which has contributed greatly to my success and helped me to determine a career path and future.

My life partner Burak İşiaçık, thank you for your love, trust, and understanding. Thank you for providing me the continued moral support and encouragement to pursue my dreams. My family supported me every step of the way from the very beginning of my educational journey. I am mostly grateful to my mother Arife Süngü, and my father Erkan Süngü for their love and support.

I am deeply indebted to my friends, Fırtına Küçük, Aybüke Akgül, Can Dağıdır, Sueda Akınay, Elif Karatekin for their help, stimulating suggestions and encouragement. Thanks to my friends Sümeyye Ağaç, Büşra Oğuzoğlu, Cansu Yılmaz and Mehmet Burak Kurutmaz for their support and invaluable collaboration while navigating the maze of this Master's degree.

ABSTRACT

PROBABILISTIC DISCRIMINATIVE REGION DESCRIPTOR FOR TIME SERIES CLASSIFICATION

Detecting discriminivative regions is a recent promising concept in many different domains for various dataset types such as image, text and time series. In time series domain, time series might be large and high dimensional because of the developing storage capacities. Although computational capacities are improved, storage and computation costs are increased. Therefore recent attempts are focused on the decreasing the computational and run time complexities. To decrease the complexity of the models, instead of using raw data, construction of the new feature representation by using the distinctive sub-sequences of the time series is the most common approach. Discriminative sub-sequences are called as shapelets in time series reflect the characteristics of the class of time series. Shapelets provide interpretable results and shapelet-based classifiers have superior accuracy on many time series datasets. Many researchers have proposed shapelet extraction methodologies for classification purpose. This study proposes a novel local feature extraction framework for time series and shapelet-based time series classification pipeline. Proposed framework provides model selection flexibility to describe the time-observation space to find local discriminative regions. After obtaining the discriminative regions, shapelets are extracted on the time-observation space by thresholding the class probability estimates to construct a new feature representation. New feature representation is calculated by the Euclidean distance between shapelets and time series. Finally, a classifier is trained by the new feature representation. Experimental results show that shapelet-based time series classification by using proposed Probabilistic Discriminative Region Descriptor (PDRD) provides competitive results on benchmark datasets.

ÖZET

ZAMAN SERİLERİ SINIFLANDIRMASI İÇİN OLASILIKSAL AYIRT EDİCİ BÖLGE BULUCU

Son yıllarda, resim, metin ve zaman serileri gibi çeşitli veri kümeleri için ayırt edici bölgelerin saptanması gelişmekte olan bir kavramdır. Gelişen veri depolama teknikleri ile birlikte, modellenebilecek zaman serilerinin büyüklüğü ve boyutu da arttı. Bilgisayarların hesaplama kapasiteleri de geliştirilmesine rağmen artan depolama ve hesaplama maliyetleri araştırmacıları daha optimum yöntemleri keşfetmeye zorlamaktadır. Zaman serisi sınıflandırma problemlerinde modellerin karmaşıklığını ve hesaplama maliyetlerini azaltmak için ham verinin tamamını kullanmak yerine zaman serilerindeki ayırt edici alt serileri kullanarak yeni öznitelik gösterimi oluşturulmaktadır. Bu öznitelik gösterimi de serinin ait olduğu sınıfı temsil etmektedir. Zaman serilerinde ayırt edici alt seriler, şekilcikler olarak adlandırılır. Birçok zaman serisi veri kümesinde, şekilcik kullanılarak kurulan sınıflandırma modelleri üstün performans göstermektedir. Ayrıca, şekilciklerin kolaylıkla açıklanabilir ve yorumlanabilir olması da kullanımını arttıran en önemli özelliklerindendir. Bu tezde, zaman serileri üzerinde ayırt edici bölgelerin zaman-gözlem uzayı üzerinde keşfedilmesi ve bu uzay kullanılarak şekilcik çıkarımıyla zaman serisi sınıflandırma akışı sunulmuştur. Uzayın tanımlanması için model seçiminin esnek olması da sunulan akışın bir avantajıdır. Ayırt edici bölgeler elde edildikten sonra, yeni bir öznitelik gösterimi oluşturmak için zaman-gözlem uzayında sınıflara ait, bir sınıflandırma modeliyle tahminlenen olasılık değerleri eşik değer ile filtrelenerek şekilcikler çıkarılır. Yeni öznitelik gösterimi oluşturulduktan sonra sınıflandırma modeli eğitilerek, zaman serisi sınıflandırma problemi çözülür. Deney sonuçlarına göre, önerilen olasılıksal ayırt edici bölge bulucu (PDRD) kullanılarak şekilcik çıkarma yöntemiyle yapılan sınıflandırma, referans veri setleri üzerinde rekabetçi sonuçlar sağlar.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

c_i	Class label of time series i
h	Length of shapelet
l	Location
L	Location set
m	Length of time series
p	Probability
s	Shapelet
S	Shapelet set
t	Time
T	Number of observations in time series
x	Observation
x_i^t	Observation of time series i on time t

LIST OF ACRONYMS/ABBREVIATIONS

1D	One Dimensional
2D	Two Dimensional
CSSL	Class-Specific Shapelets Learning
DTW	Dynamic Time Warping
ECG	electroCardioGram
FS	Fast Shapelet
GAM	Generalized Additive Model
GRSF	Generalized Random Shapelet Forest
LS	Learned Shapelet
LSS	Local Self-Similarity
PDRD	Probabilistic Discriminative Region Descriptor
SAX	Symbolic Aggregate Approximation
SIFT	Scale Invariant Feature Transform
SMTS	Symbolic representation for MTS
SURF	Speeded-Up Robust Feature
ST	Shapelet Transform
TSBF	Time Series Bag of Features

1. INTRODUCTION

Recently, time series classification has been a subject of great interest because of its various important real-world applications ranging from medicine, finance, learning sciences to seismology. For example, earthquake, explosion or other natural events produce seismogram data by recording of the ground motion [1]. Seismograph at a measuring station is defined by a function of time. Seismogram data is used to identify seismic events [1]. For detecting abnormal heart rhythms, electroCardioGram (ECG) aims to capture temporal patterns in heart signals [2].

A time series is a set of sequential observations, they might be large and high dimensional depending on the time interval. With the developing data storage technologies, more data can be accessed for modeling. However, while computational capacities have been strengthened, the need to reduce both computation and storage costs emerges. For that purpose, detecting, describing, and matching local features have attracted significant interest among the computer vision, pattern recognition and data mining communities. Detecting discriminative regions is helpful to find a suitable representation by reducing the dimension of the data. Many researchers have contributed various representation frameworks not only for time series datasets but also different types of datasets. For example in image domain, extracting local invariant features is one of the significant problems for the complex images like including distortions, complex backgrounds, low resolution. Scale Invariant Feature Transform (SIFT) [3], Speeded-Up Robust Features (SURF) [4] and Local Self-Similarity (LSS) [5] are the most known contributions proposed to detect local features on images. These methods are based on matching features and measuring the similarities. Although there is a dimension difference between images and time series, it is possible to connect image and time series representations based on the detecting discriminative regions and measuring the distance perspectives. Images have x axis and y axis and xy grid is filled by RGB space, so they are two dimensional (2D). Univariate time series are single observations recorded sequentially and time-dependent so they are one dimensional (1D).

In time series domain, discriminative regions are also 1D and extraction of them can be handled by detecting distinguishing characteristic sub-sequences which are named as shapelets. Main aim to extract shapelets is learning a new representation, and shapelet transforms can be used for clustering, classification or anomaly detection purposes. For example, extracting representative features from the time series with normal classes and learning the decision boundary with the new representation are steps of the anomaly detection pipeline. In prediction, anomalous observations will be misclassified [6]. These new representations are obtained by distance computation of sliding the shapelets across the time series to find the closest match. Therefore, distance metric is also important to obtain discriminative feature representation.

There are two main ways to calculate distance, namely elastic distance measures and measures like Euclidean distance. The challenge with shape mining and matching is that being invariant to distortions such as scale, shifting and rotation [7]. In Figure 1.1, there is a shifting example. Time series 2 is shifted 8 time unit, so discriminative region depends only on the time. Euclidean distance gives the same measurement for time series 2 and shifted version of it in different times, so it does not capture the shifting distortion.

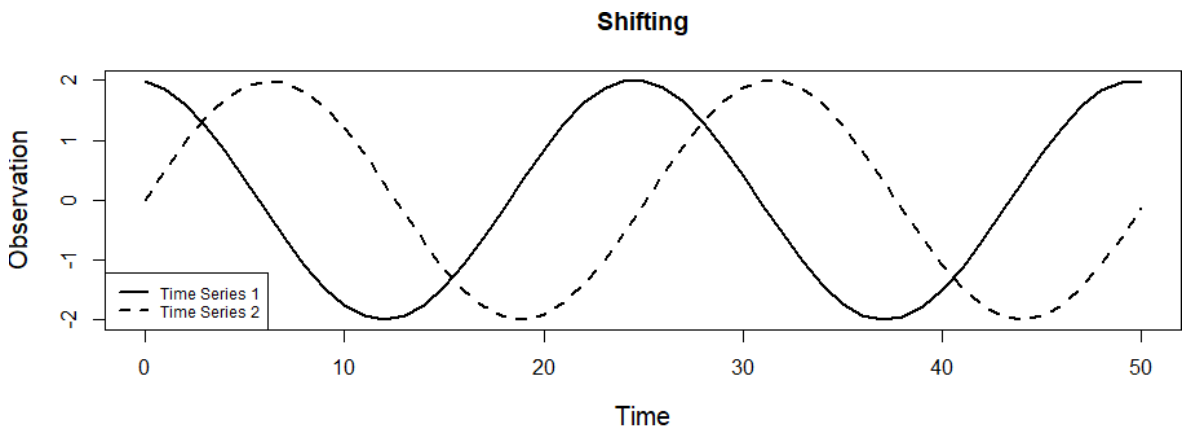


Figure 1.1. Shifting Example.

These problems are commonly handled by using elastic distance measures like Dynamic Time Warping (DTW) [8]. DTW is a more robust distance metric and requires higher computational power when it is compared by Euclidean distance. Feature rep-

resentations like Symbolic Aggregate approXimation (SAX) [9], Bag-of-Patterns Representation [10], Symbolic Representation for Multivariate Time Series (SMTS) [11] are the most appropriate approaches for such purposes. The improvements over interpretation capability of SMTS is proposed by [12] by using the local importance information. Shape-based methods that also aim to detect the discriminative region on time series are faster than methods that use elastic distance, but they are sensitive to distortions [12]. Also, DTW distance is not easy to interpret [13]. There is a trade-off between computation, interpretation and handling the distortions for shape-based similarity measures. Shape-based approaches attract greater attention because of their interpretation and computation advantages. For example, Time Series Bag of Features (TSBF) [13] generates shapelets and learns class probability estimates for each shapelets. Then, bag of feature representation are built from these probabilities to train a classifier. Class-Specific Shapelets Learning (CSSL) [14] generates and updates shapelets for each class. Also, class-shared shapelets are eliminated since they are useless to discriminate the class. Easy interpretation and being representative of the class are increased the attention over shapelet-based time series classification methods. First shapelet approach is proposed by [15] which calculates the information gain over the all possible shapelet candidates. Also, Shapelet Transform (ST) [16] and Generalized Shapelet Forest (GRSF) [17] are pruning all possible shapelet candidates based on the information gain. Calculating and pruning the all possible shapelet candidates is a computationally heavy process, so recent approaches are focused on reducing the computational complexity by optimization and piecewise approximation such as Learned Shapelets (LS) [18] and Fast Shapelets (FS) [19], respectively.

This work proposes a flexible local feature extraction framework called *Probabilistic Discriminative Region Descriptor (PDRD)* for time series and shapelet-based time series classification pipeline. The proposed shapelet-based time series classification method contains 3 steps. Firstly, shapelet discovery aims to extract shapelet by using class probability estimates. Then, new feature representation is constructed by calculating the Euclidean distance between shapelets and time series. Finally a classifier is trained by using the new feature representation. In Figure 1.2, steps of proposed pipeline are seen.

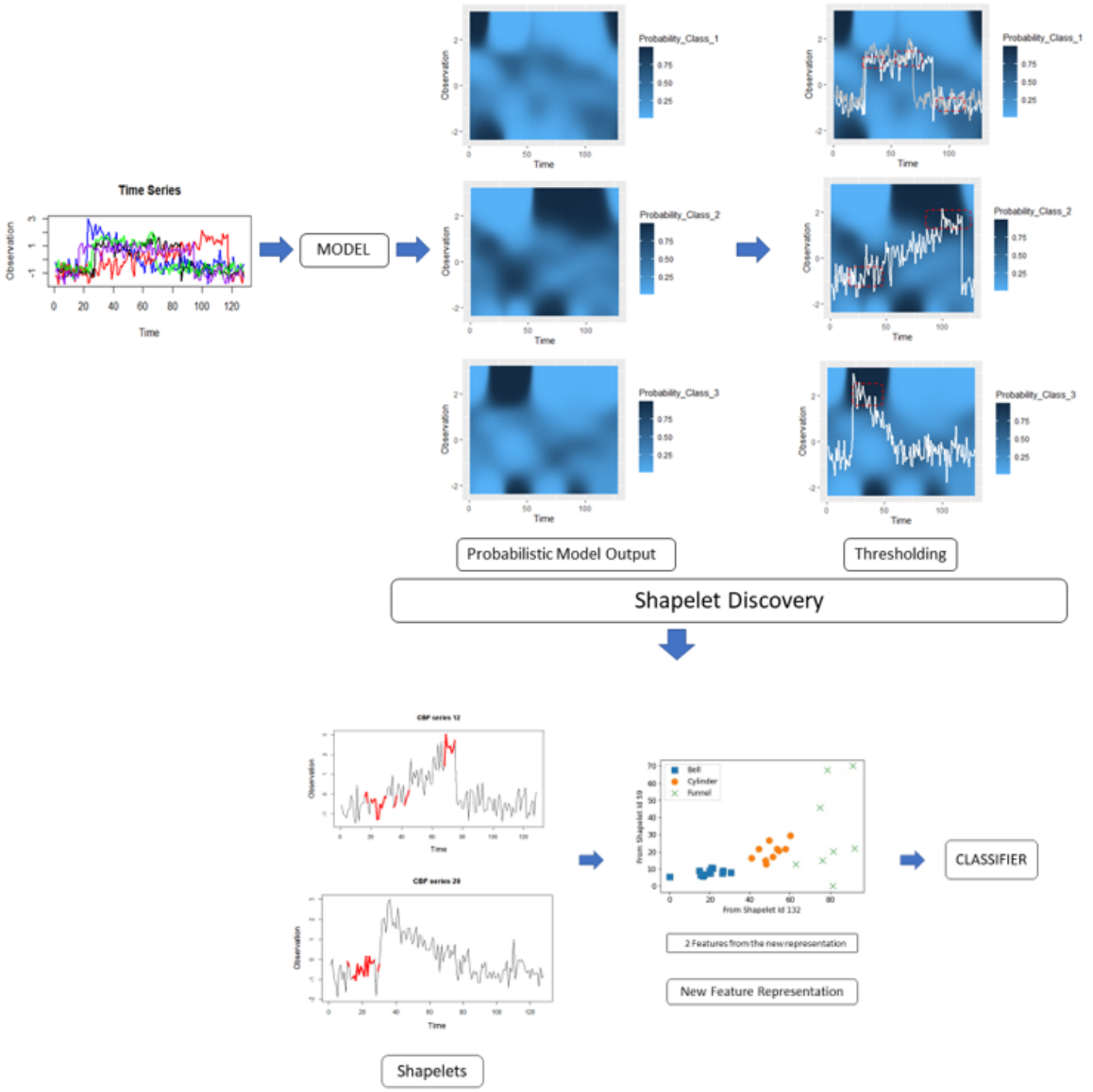


Figure 1.2. PDRD Pipeline.

Extracting local feature information only depends on a model which can produce probabilistic output such as logistic regression, decision trees, random forest, Generalized Additive Model (GAM) etc. Once class probability estimations are obtained for each time series observation, shapelets are extracted by filtering probabilities with a given threshold. Furthermore, new feature representation is obtained by calculating Euclidean distance and location information with sliding window approach between shapelets and time series. Finally, any classifier can be used for the time series classification purpose since new feature representation is in a tabular format, not time dependent. The novel part of the pipeline is that the estimation of the class probabil-

ities on time-observation space and extracting shapelets from these space by using a given threshold. After shapelets are obtained, calculation of Euclidean distance with sliding window approach to have the new representation and classification steps are similar to the state-of-the-art shapelet-based classification methodologies. Therefore, handling the distortions might be still problematic with the use of Euclidean distance, but it reduces the computationally complexity in the shapelet discovery step. Also, the interpretability and the visualization of the shapelets are notably different than the traditional shapelet interpretations.

The rest of the thesis is organized as follows: In Section 2, background information on tree-based learning methods, GAM and the usage of them in this thesis are provided. Section 3 provides a detailed literature review on discriminative region detection and shapelet-based classifications. In Section 4, a detailed explanation for the pipeline and interpretation are presented. Section 5 introduces the data used in this thesis, the experiments and compares the methods, and Section 6 provides the conclusion and future work.

2. BACKGROUND

In the proposed framework, identifying discriminative regions can be modeled by an algorithm that can produce probabilistic outputs. In this thesis, decision tree, random forest and generalized additive model (GAM) are presented. This section introduces the basics of these models before introducing the method.

2.1. Tree-Based Learning

Decision trees (DT) are a type of supervised machine learning model and can be used for both regression and classification problems. Decision trees are defined as a collection of sequential decision rules. Nodes and leaves are main entities of decision trees. Nodes are constructed as splitting the observations based on a given decision rule. These rules are generally defined by minimizing the impurity in the child nodes until the given threshold is satisfied.

Algorithm allocates the data into two branches by using all possible splits namely, with divide and conquer approach. These splits aim to minimize the sum of the squared deviations from the mean and splits are iteratively updated until each node reaches the minimum error for the regression trees. For classification trees, accuracy performance is calculated by gini score, entropy and chi-square etc. Gini score demonstrate the quality of the split based on the class prediction, entropy shows the purity of a node, chi-square calculates the statistical significance of differences between the parent and internal nodes.

Decision trees can be also used for time series classification problems after transforming the time series into a single matrix. In the single matrix, observations and time indexes are considered as instances and their classes are assumed to be the same as the time series classes. For example, sample from Cylinder-Bell-Funnel (CBF) which is one of the most popular synthetic datasets with three classes, namely "Cylinder", "Bell", and "Funnel" from Bagnall *et al.* [20]. CBF is a simulated data where each

class is obtained by standard normal noise added an offset term. Sample from each class can be seen in Figure 2.1.

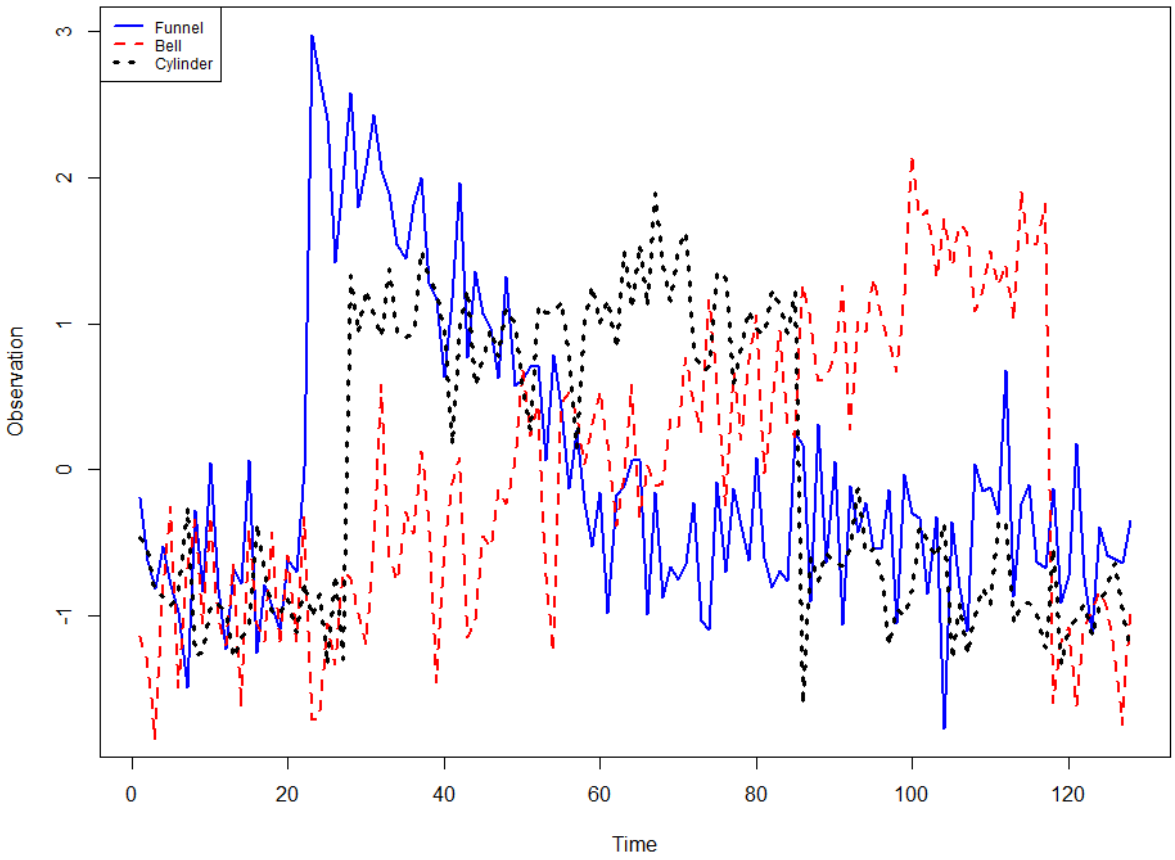


Figure 2.1. Sample from each class of CBF.

Distinguishing the time series observations based on their classes is the main purpose of time series classification problems. Feature-based approaches try to capture discriminative regions for the new representation, then classification step follows. For example in Figure 2.1, observations between 1 and 1.5 and times between 35 and 40 for cylinder, observations above 2 and times between 55 and 60, also values between 0 and 0.5, times between 35 and 40 for bell, observation around 2 and times between 25 and 40 for funnel are discriminative regions. Feature-based approaches aim to find these regions algorithmically. Before proceeding to train an example of decision tree, time series are transformed into a new sequential representation in which observations in one row can be represented by multiplexing class and time. Table 2.1 illustrates a sample of transformed CBF time-observation representation.

Table 2.1. Transformed representation of the time series observations.

Time	Observation	Class
1	-0.4642	1
2	-0.5550	1
3	-0.8428	1
4	-0.8658	1
.	.	.
.	.	.
.	.	.
1	-0.5687	2
2	-0.8881	2
3	-0.0479	2
4	-0.0380	2
.	.	.
.	.	.
.	.	.

After obtaining the transformed representation of the time series, a decision tree classifier can be trained by using the class variable as a categorical response and time and observation as features. In Figure 2.2, a decision tree for CBF dataset is seen. There are final decisions on leaf nodes, and decision rules are on interior nodes. At each node majority class, number of instance per classes, percentage of the data points are represented row by row, respectively. For example, left sub-tree shows that observation between 0.27 and 1.4 concludes with class 1 in the first row of the leaf node, namely cylinder. As it is seen in Figure 2.1, cylinder is between times 25 and 70 and the observations above 0.27 and below 1.4. In the second row of the left leaf node implies that there are 624 samples from class 1, 368 samples from class 2 and 192 samples from class 3. The last row shows that 31% of the dataset follows the left sub-tree. Also, right sub-tree shows that observation below 0.27 and above 0.89 with time after 69 give the class 3, funnel. It is also obvious in Figure 2.1 from funnel class. Since

the decision tree is trained by train dataset, and samples are only one series from each class, it reflects the aggregation of time series in CBF train dataset.

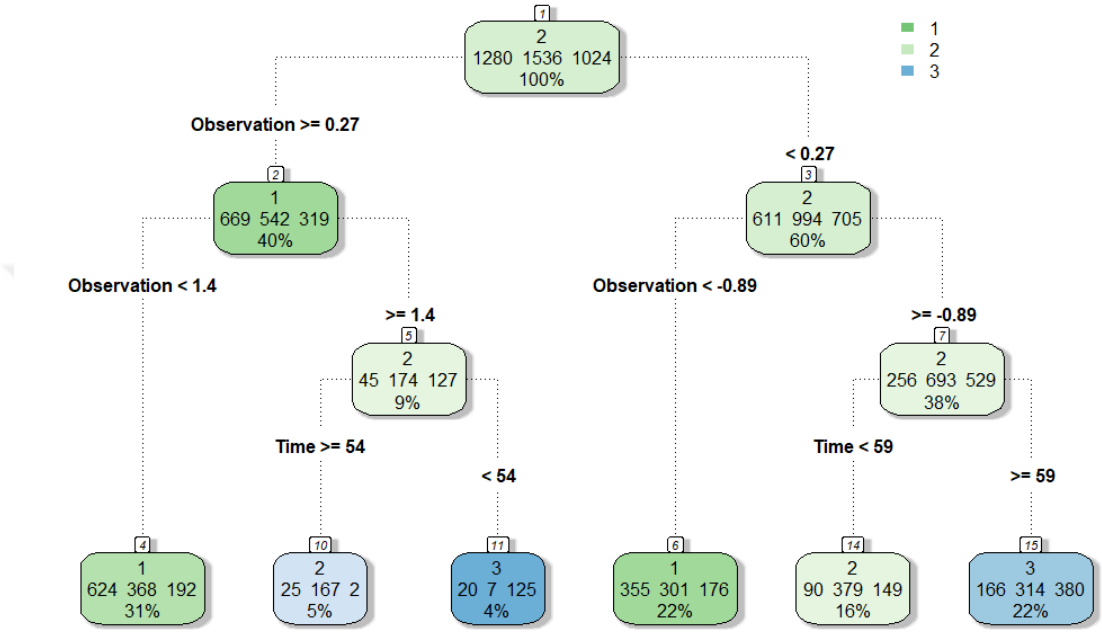
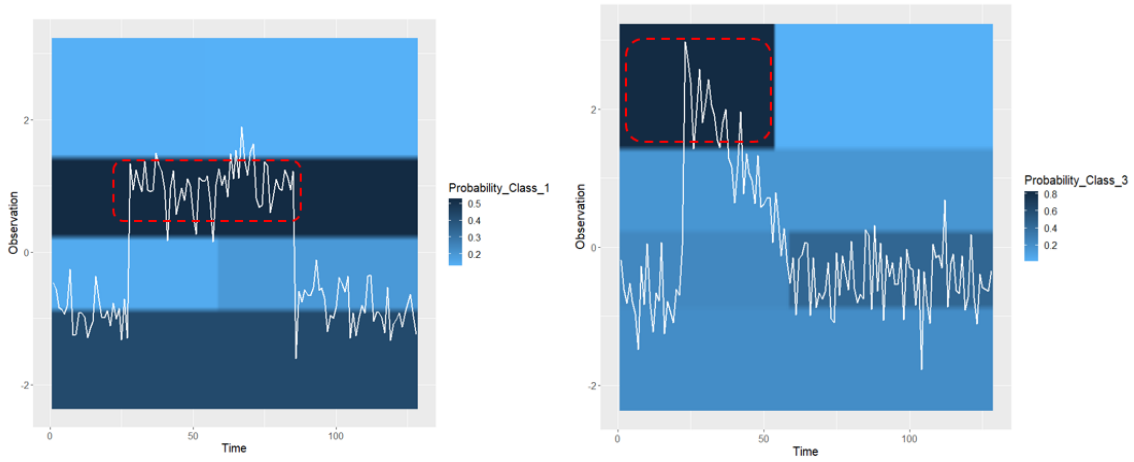


Figure 2.2. Classification Tree of CBF.

Decision tree nodes direct to the dashed box on the response surfaces in Figure 2.3 for cylinder and funnel classes, respectively. In other words, increasing the quality of the response surface would be helpful to find more discriminative regions. For that purpose, increasing the model complexity or using models that can produce continuous probabilistic outputs might be some solutions. Based on the stability perspective, decision trees might be problematic because they are sensitive to small changes such as change of data and different kind of new representations in time series. Random forest is an example of more complex and stable model when compared to the decision tree.

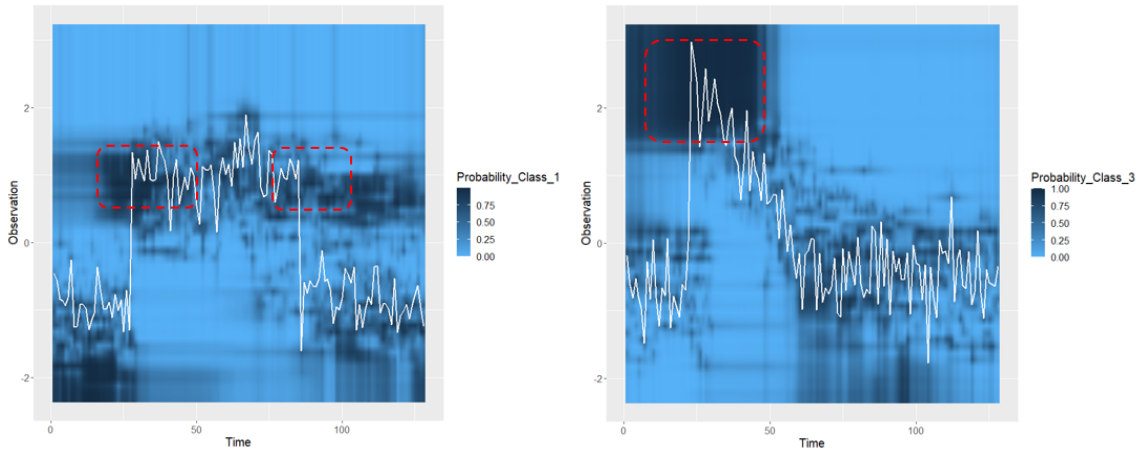


(a) Cylinder Discriminative Region. (b) Funnel Discriminative Region.

Figure 2.3. Response surface of decision tree.

Random forest (RF) is an ensemble learning method for regression and classification tasks. It is simply a collection of decision trees. Outputs of many decision trees are combined by mean of the nodes for regression and majority voting methodology when it is used for classification. Also, variable importance calculation of random forest ensures easy interpretation. Variable importance is calculated by the mean decrease in impurity called as gini importance. Split criterion improvements at each split in each tree, represent the importance of the split variable. This measure is aggregated for each variable on all trees to define the importance of the variable [21].

Both decision trees and random forest are non-parametric models, so they have no assumptions, provide flexibility and higher performance. As it is seen in Figures 2.3 and 2.4, they capture the almost the same region for cylinder and funnel classes. However, it is obvious that discriminative regions are more precise and detailed.



(a) Cylinder Discriminative Region. (b) Funnel Discriminative Region.

Figure 2.4. Response surface of random forest.

2.2. Generalized Additive Model (GAM)

A *generalized additive model (GAM)* [22] is a generalized linear model with a linear predictor involving a sum of smooth functions of covariates. For the sake of simplicity, the general model structure is introduced for linear regression. For classification problems, only the distribution of response variable is changed based on the distribution family which is defined as

$$g(\mu_i) = A_i\theta + f_1(x_{1i}) + f_2(x_{2i}) + f_3(x_{3i}, x_{4i}) + ..., \tag{2.1}$$

where $\mu_i = E(Y_i)$ and $Y_i \sim EF(\mu_i, \Phi)$ and Y_i is a response variable, $EF(\mu_i, \Phi)$ is the exponential family distribution with mean μ_i and scale parameter Φ, A_i is a row of the model matrix for a strictly parametric model components, θ is the related parameter vector, and the f_j are smoother of the covariates, x_k [23].

Tensor product smooths provide modeling the responses to interactions of multiple covariates with different units. Instead of creating N-dimensional basis for N covariates, computing the tensor product $X_1 \otimes X_2 \otimes X_3 \otimes ... \otimes X_N$, gives a matrix correspond to their respective basis dimensions. Therefore, tensor product basis functions are constructed by products of marginal basis functions. This allows to consider the directions of all covariates. In the given problem, time and observations are dependent

to each other. Tensor product basis of time and observation allows to move them together, dependently. GAM with tensor interaction and using only tensor interaction instead of adding smooth covariates are defined as

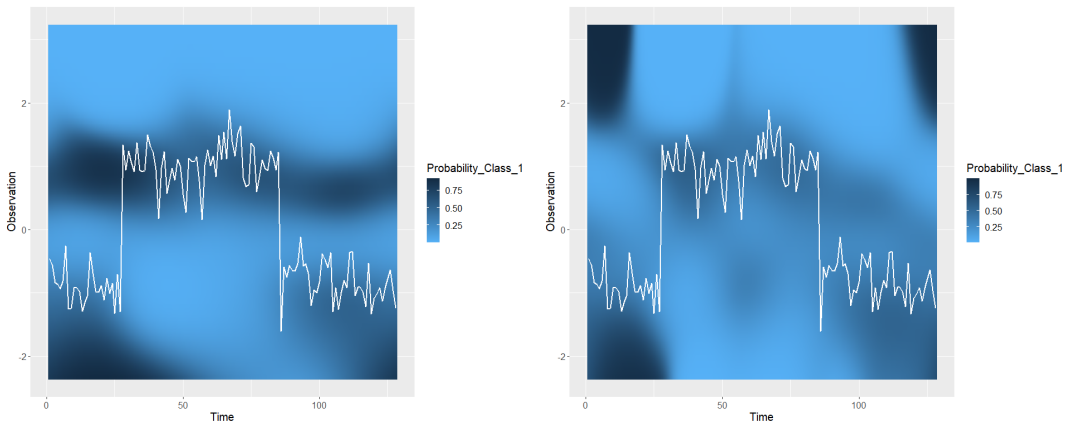
$$Y = \beta_1 + \beta_2 f_1(X) + \beta_3 f_2(T) + \beta_4 f_3(X,T) + \epsilon, \tag{2.2}$$

where f_1, f_2, f_3 are smoothing functions, Y response variable, X, T covariates and ϵ is independent $N(0, \sigma^2)$ random variable.

$$Y = \beta_1 + \beta_2 f(X,T) + \epsilon, \tag{2.3}$$

where f is tensor product smoothing function, Y response variable and ϵ is independent $N(0, \sigma^2)$ random variable.

Modeling Equation (2.3) has both run time and complexity advantages over the Equation (2.2). In addition to these advantages, response surface of the model Equation (2.3) is more discriminative than the model Equation (2.2). For example, the comparison of response surfaces for the class named cylinder of CBF dataset is seen in Figure 2.5 by keeping all other variables fixed.



(a) Tensor interaction with covariates. (b) Tensor interaction without covariates.

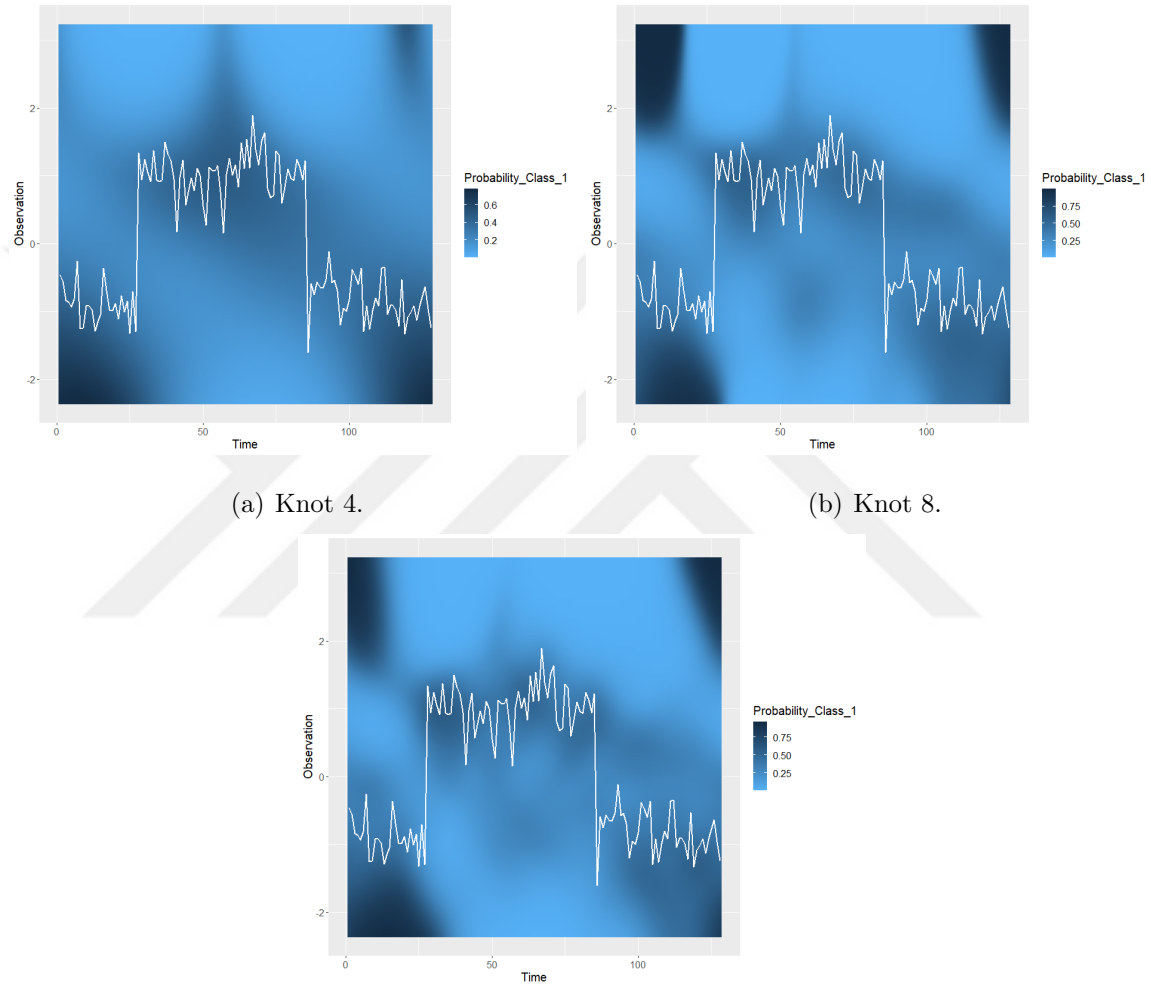
Figure 2.5. Response surface comparison of tensor interaction with covariates and without covariates for cylinder.

Cylinder sample is plotted on the response surfaces and darker regions are discriminative for cylinder class and they are consistent with given arguments for the decision tree. Tensor interaction without covariates has run time and complexity advantages over tensor. As it is seen that response surface of tensor interaction without covariates is almost similar to response surface of tensor interaction with covariates response. Therefore, using the tensor product smooths for time-observation interaction is important to detect the discriminative patterns that are informative about the class based on the above construction of the basis and direction discussions. In GAM modeling, basis expansion is a set of functions, so interpolation of these piece-wise functions is handled by spline interpolation which also provides continuous response surfaces.

Spline interpolation is used for smoothing the piece wise polynomials of curves which joins multiple polynomial curves, and that local regions are called as knots to save the continuity [23]. The number of knot is a parameter needed to optimize. Among many alternatives smoothing splines and knots for modeling, choosing thin plate (tp) spline and cyclic cubic (cc) regression spline brings following opportunities. Using thin plate spline for observations is low dimensional isotropic smoothers which means that rotation of the covariate coordinate system does not change by smoothing. Also, it does not require optimize number of knots because thin plate handles the dimension/rank reduction by truncated eigen-decomposition. The reason behind the using cyclic cubic spline is that the times neighbouring knots are connected by sections of cubic polynomial constrained to be continuous up to and including second derivative at the knots ie. cc spline wraps at the smallest and largest times [23]. For cc spline knot optimization is still necessary.

For example, the comparison of response surfaces for the class cylinder of CBF dataset depending on the knot of cc spline and tensor interaction without covariates is seen in Figure 2.6 by keeping all other variables fixed. Increasing number of the knots to 16 increases the complexity of the model, but when it is compared with 8 knots does not improve the response surface. Thus, optimization of the number of knots should be done. Cylinder is between times 25 and 70 and the observations above 0.27 and

below 1.4 discriminative. As it is seen in Figure 2.6, increasing the number of knots from 4 to 8 target area is getting darker. The difference between 8 knots and 16 knots is negligible. Thus, knot optimization is required to have an optimized model.



(a) Knot 4.

(b) Knot 8.

(c) Knot 16.

Figure 2.6. Response surface comparison of knots with 4, 8 and 16 for cylinder.

Restricted maximum likelihood (REML) is one of the methods for producing unbiased estimates of variance and covariance. GAM uses the REML to estimate the coefficients by penalized likelihood maximisation and estimation of smoothing parameters [23]. Using multinomial family of GAM, in other words multinomial logistic regression, is important for the proposed time-observation interaction surface. The idea behind the *multinomial logistic regression* is that each response variable Y_i is one

of $C + 1$ category labels, $0, 1, 2, .., C$. Category 0 is assumed as a referance category, and η^c is a linear predictor associated with each of the remaining C categories. The probability of Y_i taking the value 0 is then $\frac{1}{1+\sum_k exp(\eta_i^c)}$, while the probability that it takes the value $j \neq 0$ is $\frac{exp(\eta^j)}{1+\sum_k exp(\eta_i^c)}$ [23]. Prediction of response variable produces $C+1$ columns matrix of class probabilities for multinomial logistic regression.

3. RELATED WORK

In this section, related work in detection of interest regions and the discriminative sub-sequences named as shapelets on time series data are briefly discussed. Detecting discriminative regions resolves the problem of high dimensionality because its main aim is constructing new representation for the given dataset by reducing the dimension. It is seen in text datasets by describing the interesting or context based keywords with simply bag-of-words approach in one dimension (1D), in image datasets detecting the edges, corners and extracting local matching features by using some popular algorithms like SIFT [3], SURF [4] etc. to create bag-of-visual-words in two dimension (2D). Also these discriminative regions are categorized by using symbolic word dictionary like SAX, [9], symbolic representation for MTS (SMTS) [11], Indirect Structure Testing Software (ISTS) as an extension of SMTS [12] etc. in time series datasets. One may consider that all shapelet methodologies for time series classification problem are algorithms to detect the interest regions in 1D.

The main difference of shape-based similarity methods is distance selection. Dynamic Time Warping (DTW) measure and Euclidean distance metric are the most common distance measures for time series clustering and classification. Although methods with DTW distance measure can handle rotation-invariant similarity, and distortions on time series, it is not computationally efficient. The reason behind this inefficiency is that DTW is not a distance metric in which must satisfy symmetry, self-identity, non-negativity, and triangular inequality. DTW does not satisfy triangular inequality property that is why it is called as DTW measure and it measures the minimum cumulative distance between series. DTW has an advantage of non-linear mapping, and it is effectively used with the nearest neighbor classifier for obtaining the shape of time series [24]. These methods generally do not require new feature representation for clustering and classification, and the results are obtained by only distance calculation. However, extracting shapelets provides some opportunities for obtaining the new representation and generally Euclidean distance is used because of its computational advantage. Converting time series into new tabular representation gives an

opportunity to use state-of-the-art classifiers for the classification. Because of these, shapelet extraction with Euclidean distance is used for the new feature representation in the proposed framework like many other shapelet based time series representation methods.

The literature on shapelet discovery generally focuses on finding the best sub-sequences, shapelets, among all possible sub-sequences of the time series. Shapelet extraction and their use for a classification purposes is proposed by [15]. All possible sub-sequences are introduced as potential shapelet candidates so calculating information gain for all possible shapelet candidates is not feasible. Therefore, improvements on the domain are focused on finding the best shapelets without searching all possible candidates and consequently being faster. Early abandoning and entropy pruning are applied to improve the running time of brute force method [15].

Searching for the best k shapelets from time series and calculating the Euclidean distance between transformed version of the data and possible shapelets proposed by [16]. It also requires to enumerate over the all possible shapelets. Random sampling over the all possible shapelet space does not cause the decreasing accuracy performance [25].

Feature representation by reducing the dimension and transforming the time series data into symbolic words to improve the run time is proposed in Symbolic Aggregate Approximation (SAX) [9]. Fast Shapelets (FS) [19] is the improved version of SAX. Although it does not ensure accuracy improvement, provides faster methodology by searching shapelets over random masked SAX representation. Since FS decreases the dimension of the time series, it is computationally efficient. Also it uses the advantage of discrete representation by using hashing [19] to reduce the complexity.

Generalized Random Shapelet Forest [17] prunes the shapelet space to find the best shapelet set. For that purpose it requires number of targeted shapelets, minimum and maximum length of the desired shapelet candidates to generate the shapelet set. Then, Euclidean distance calculation between time series and the shapelets in the gen-

erated shapelet set follows to construct given number of trees. Thus, information gain is calculated from each split. Although accuracy performance of GRFS is satisfactory, it requires to tune minimum length, maximum length and sample size and parameter optimization step has high computational cost.

Instead of searching for the best shapelets among all the shapelet candidates, feasibility of learning them is discussed by using stochastic gradient descent optimization in Learned Shapelets [18]. However, learning shapelets requires high computational cost, large number of parameters to tune and wisely chosen initial shapelets which has increasing effect on the accuracy. Learning-based methods obtain higher classification accuracy than search-based methods. Nevertheless, learned shapelets are fixed after training so they can not overcome the distortions of local patterns at test dataset. Since there is no constraint for learning based methods that the shapelets are similar to other subsequences, that reduces their interpretability, which is often the reason why shapelet-based methods are preferred.

In this work, the novel part is that local probabilities are learned by a supervised learning model which can produce probabilistic outputs to find discriminative regions on the time-observation space. Visualization of the class probability estimates over a time-observation space provides benefits in terms of understanding the possible temporal relations related to the class.

4. PROBABILISTIC DISCRIMINATIVE REGION DESCRIPTOR FOR TIME SERIES CLASSIFICATION

This section introduces discriminative region detection for univariate time series dataset which is a set of pairs of n time series and its class label, $D =< T_1, c_1 >, < T_2, c_2 >, < T_3, c_3 >, ..., < T_n, c_k >$. The proposed approach can handle time series of variable length, but for the sake of simplicity, assume that each time series T_i contains m ordered real values, denoted as $T_i = (T_{i1}, T_{i2}, ..., T_{im})^T$. In the first step, given series are represented as time (t), observation (x) pairs. That initial representation Φ_{Ntx2} is a transformed matrix represented as

$$\Phi_{NTx2} = \begin{array}{c} \begin{matrix} t & x \end{matrix} \\ \begin{pmatrix} 1 & x_1^1 \\ 2 & x_2^2 \\ . & . \\ . & . \\ . & . \\ t & x_N^t \end{pmatrix} \end{array}. \tag{4.1}$$

Then, the proposed framework requires a model that can produce probabilistic output to train the obtained initial representation Φ_{NTx2} and to get the probabilities of belonging to the classes for each time and observation pairs. Final output is probabilities for each time-observation pairs and for each class level. Higher probabilities present the discriminative regions on time-observation space. It is possible to visualize probabilities with any grid for each class label which provides easy interpretation and detecting important sub-regions. These sub-regions are also can be considered as a *shapelet* S which is a discriminative subsequence of a time series.

In the literature, subsequence S of length $h < m$ of time series T_i generally starting at position j can be written as $S = T_i^l = T_{ij}, T_{i(j+1)}, ..., T_{i(j+l+1)}$. However, shapelets do not have to be contiguous. Also, subsequences with some missing values

can be described as shapelets. Identifying discriminative regions methodology can be used for the shapelet extraction. Based on the model selection, in the proposed framework, shapelets might includes missing values or not. Flow can be represented as

$$\begin{matrix} t & x \end{matrix} \begin{pmatrix} 1 & x_1^1 \\ 2 & x_2^2 \\ . & . \\ . & . \\ . & . \\ t & x_n^t \end{pmatrix} \rightarrow MODEL \rightarrow \begin{matrix} c_1 & c_2 & . & . & . & c_k \end{matrix} \begin{pmatrix} {p_1}_1^1 & {p_2}_1^1 & . & . & . & {p_k}_1^1 \\ {p_1}_2^2 & {p_2}_2^2 & . & . & . & {p_k}_2^2 \\ . & . & . & . & . & \\ . & . & . & . & . & \\ . & . & . & . & . & \\ {p_1}_n^t & {p_2}_n^t & . & . & . & {p_k}_n^t \end{pmatrix}. \tag{4.2}$$

In the following subsections, shapelet discovery, new feature representation by using the obtained shapelets and classification steps are introduced by using discriminative region identification idea.

4.1. Shapelet Discovery

In this subsection, the first step of the framework, shapelet extraction and importance of model selection for the shapelet extraction are introduced.

Shapelet-based classification methods aim to find the best shapelet candidate set S. Therefore shapelet discovery is the most important step of the shapelet-based classification methods. The quality of the shapelets affects feature representation and consequently the accuracy of the classifier. Proposed shapelet discovery procedure starts with building a learner that can produce probabilistic outputs like decision trees, random forest, generalized additive models (GAM), and generalized linear models (GLM). Higher probabilities of observation and time interaction identify the discriminative regions on the time-observation space. After obtaining the probabilities, there is one parameter named as threshold to filter out the higher probabilities. Thus, time and observations of the remaining probabilities are extracted shapelets. This filtering causes gaps on the shapelets. That provides efficiency and feasibility on computation for any

size of dataset. Also, discriminative information from the beginning, middle, end or anywhere of the series is covered in only one shapelet. Proposed shapelet discovery algorithm is shown in Figure 4.1.

Input: Training set D, Class set C, Threshold Q where training set includes observation values x_t^n and time index t, Q is a percentile threshold list.
Output: Shapelet set S, Location set L.
Function *ShapeletDiscovery(D, C, Q).*
Train a learner that produces probabilistic outputs with dataset D, C.
Predict observation values x_t^n to get class probability estimates for all classes P_i where i in $1 \leq i \leq |C|$.
For class probability estimates P_i **do**
For threshold percentile Q_j **do**
Filter P_i based on a threshold Q_j where j in $1 \leq j \leq length(Q)$.
Generate shapelet s from observations of remaining probabilities P and starting time of the shapelet as location l.
Add shapelet s to the Shapelet set S.
Add location l to the Location set L.
End For
End For
Return Shapelet set S, Location set L.

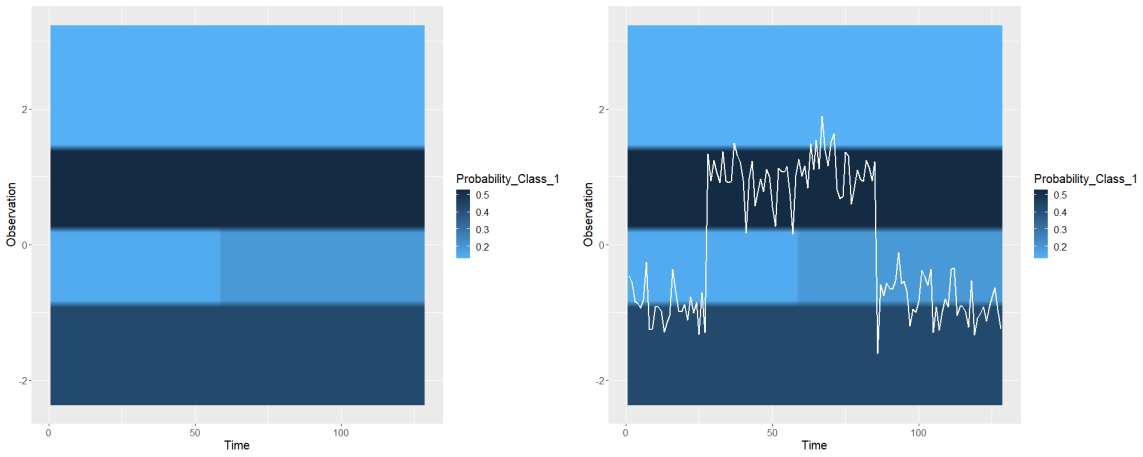
Figure 4.1. Pseudocode of the shapelet discovery with PDRD.

Model selection is one of the most important steps to define the time-observation space. The novel part of the framework is creating a response surface and filtering the discriminative regions on time-observation space, so how the response surface looks like when discrete and continuous models are chosen is discussed in these two subsections by using CBF dataset as an example.

4.1.1. Discrete Modeling

For the illustration of discrete modeling, non-parametric tree-based models are trained on CBF dataset, namely decision tree and random forest. Tree-based models learn the data with top-down and divide-and-conquer approaches, so time-observation space looks like discrete and axis parallel. In the following figures, there are response surfaces predicted by using decision tree for each classes to discriminate the space in a discrete manner. Darker regions are potential time and observations to define shapelets.

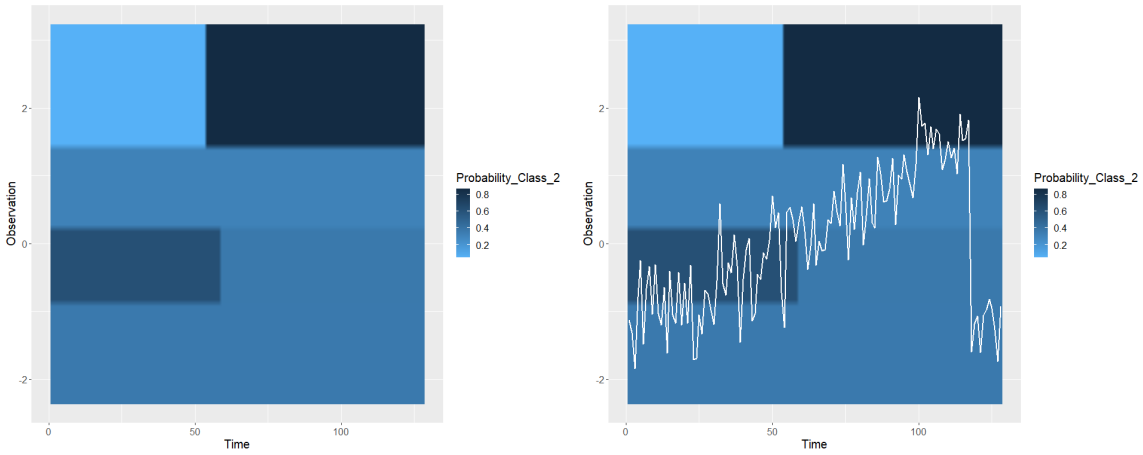
In Figure 4.2(a), discriminative areas for class 1 namely cylinder are seen for a grid and in Figure 4.2(b), one sample from cylinder class is plotted on the grid. The intersection of darker region and sample time series plotted as white line can be filtered as shapelet by a probability threshold of 0.5.



(a) Cylinder Response Surface. (b) Cylinder Sample on Response Surface.

Figure 4.2. Response surfaces of cylinder class from decision tree.

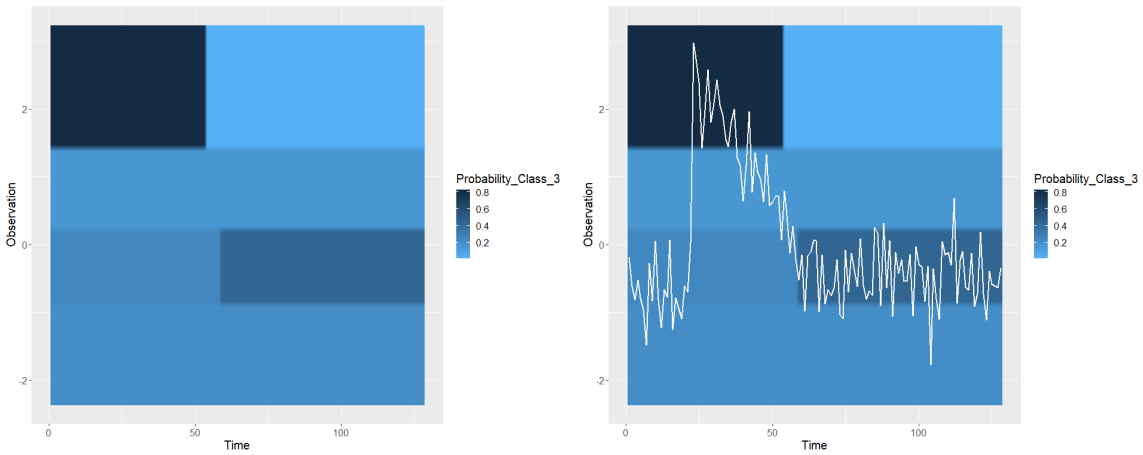
In Figure 4.3, discriminative areas for class 2 namely bell are seen for a grid and one sample from bell class is plotted on the grid. The intersection of darker region and sample time series plotted as white line can be filtered as shapelet by a probability threshold of 0.8.



(a) Bell Response Surface. (b) Bell Sample on Response Surface.

Figure 4.3. Response surfaces of bell class from decision tree.

In Figure 4.4, discriminative areas for class 3 namely funnel are seen for a grid and one sample from bell class is plotted on the grid. The intersection of darker region and sample time series plotted as white line can be filtered as shapelet by a probability threshold of 0.8.

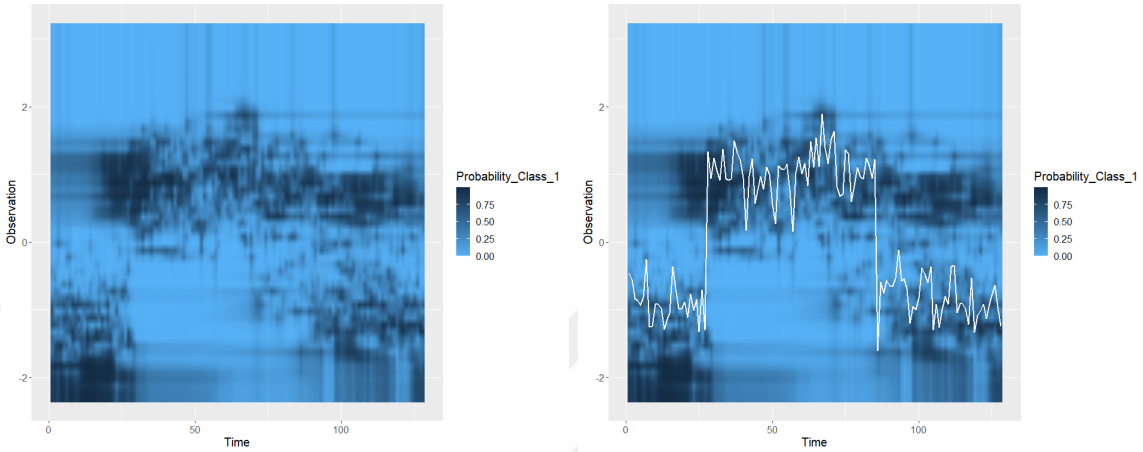


(a) Funnel Response Surface. (b) Funnel Sample on Response Surface.

Figure 4.4. Response surfaces of funnel class from decision tree.

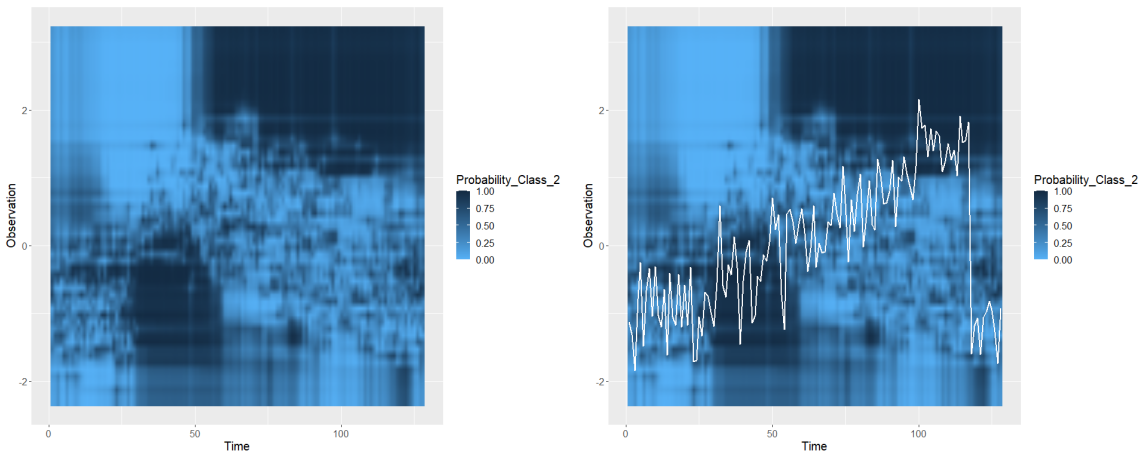
According to this approach, improving the response surface provides more discriminative shapelets. One way of improving the response surface is increasing the model complexity. For example, Figures 4.5, 4.6, and 4.7 are response surfaces pre-

dicted by random forest. As it is seen modeling with random forest provides more exhaustive discriminative regions.



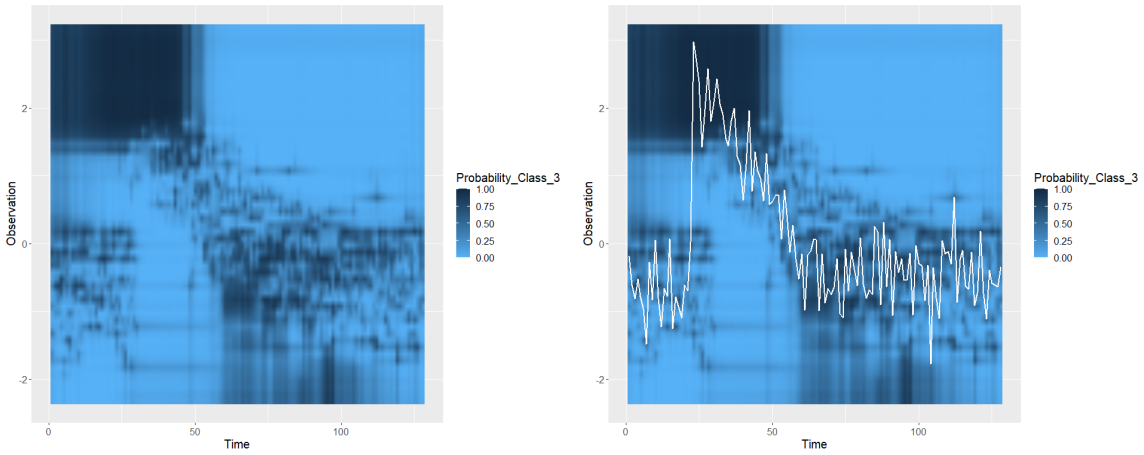
(a) Cylinder Response Surface. (b) Cylinder Sample on Response Surface.

Figure 4.5. Response surfaces of cylinder class from random forest.



(a) Bell Response Surface. (b) Bell Sample on Response Surface.

Figure 4.6. Response surfaces of bell class from random forest.



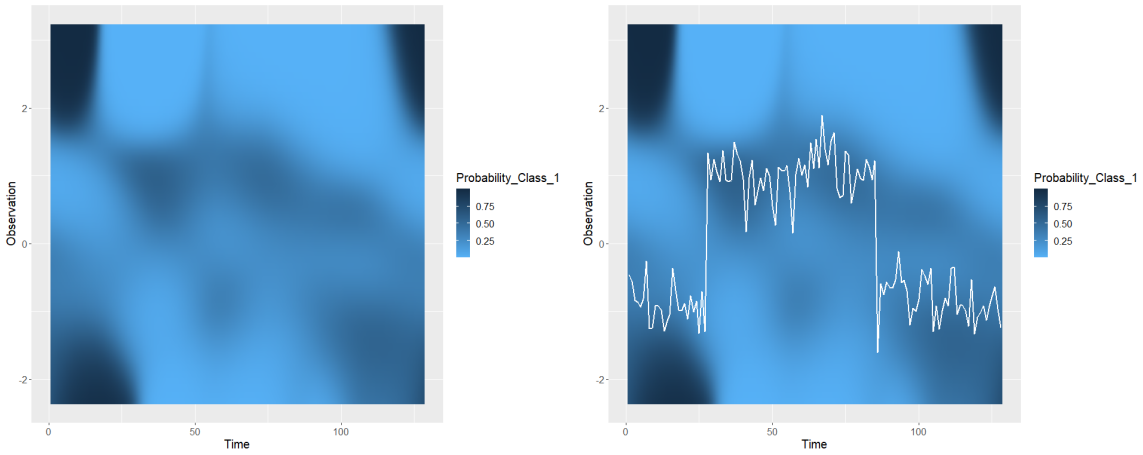
(a) Funnel Response Surface. (b) Funnel Sample on Response Surface.

Figure 4.7. Response surfaces of funnel class from random forest.

Improvements compared to decision trees for all classes are obviously seen. It is known that random forest is the extension of decision trees and has an advantage about giving accurate and precise results over decision trees. Increasing the threshold also helps to capture more accurate and discriminative regions on the space.

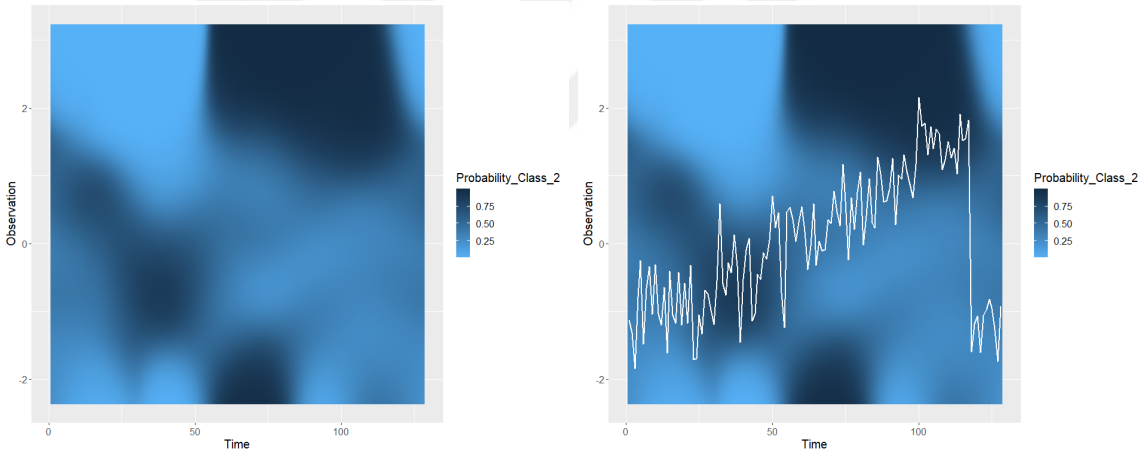
4.1.2. Continuous Modeling

For the illustration of the continuous modeling, GAM is used. The advantage of the GAM is using smoothers. The motivation behind thin plate spline behaves like penalized regression splines and they do not require knot optimization and cyclic cubic spline has advantage over periodic time series, and it is continuous up to the second derivative. The following Figures 4.8, 4.9, 4.10 illustrates the response surfaces obtained by the tensor product interaction of time with cyclic cubic spline, 8 knot and observation with thin plate spline.



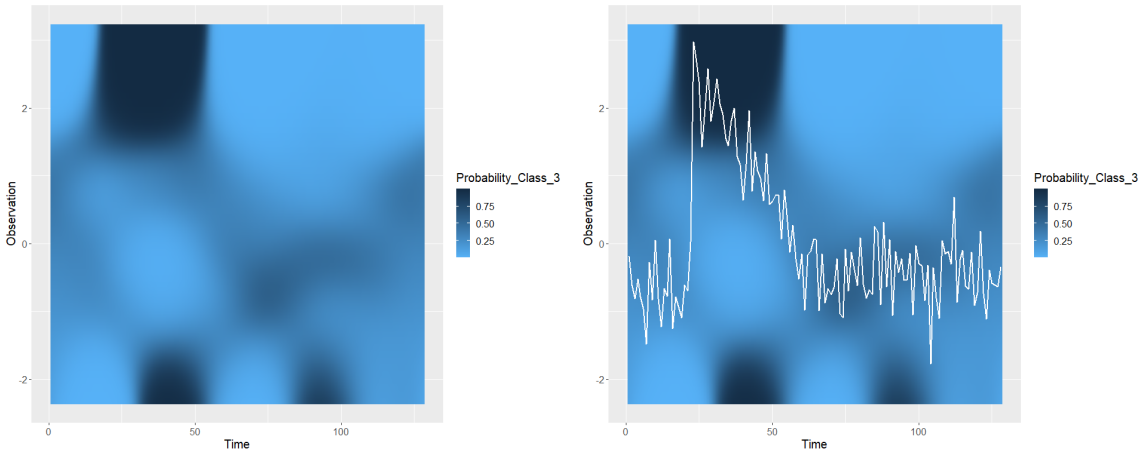
(a) Cylinder Response Surface. (b) Cylinder Sample on Response Surface.

Figure 4.8. Response surfaces of cylinder class from GAM.



(a) Bell Response Surface. (b) Bell Sample on Response Surface.

Figure 4.9. Response surfaces of bell class from GAM.



(a) Funnel Response Surface. (b) Funnel Sample on Response Surface.

Figure 4.10. Response surfaces of funnel class from GAM.

When discrete models and GAM are compared, it is obvious that continuous responses are smoother and better to capture higher thresholds. Therefore, obtaining shapelets from these continuous responses would be short, continuous and more discriminative sub-sequences as it is desired. Also, it is possible to extract some shapelets with missing values for both discrete and continuous modeling such as in Figure 4.11.

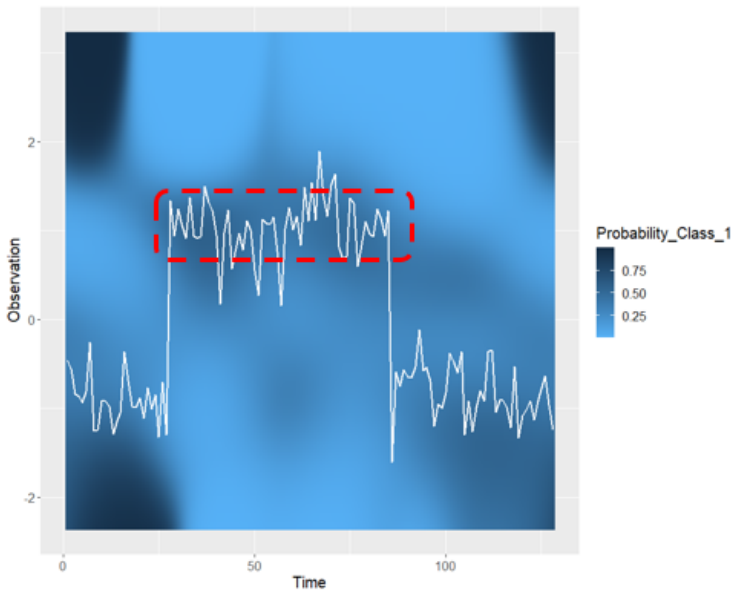


Figure 4.11. Obtaining Shapelets by Thresholding.

For example, one may put a probability threshold as 0.5 and it might be seen on both time between 15-20 and between 55-60. Thus, observations between time 20 and 55 will be missing. The proposed pipeline considers that situation also and does not have any distance calculation for that missing parts in the representation of time series step. In Figure 4.11, response surface of cylinder class of CBF dataset is seen with a sample from cylinder class. Thresholding provides the extraction of shapelets. For example, in the case of defined the probability threshold as 0.5, time and observations in the dashed box determine the shapelet in the Figure 4.11.

4.2. Representation of Time Series

In this subsection, the second step of the framework, new feature representation by using obtained shapelets is introduced. An algorithm which calculates the Euclidean distance between the shapelet candidate S_k and time series T_i with sliding a window of size 1. The process is illustrated in Figure 4.12.

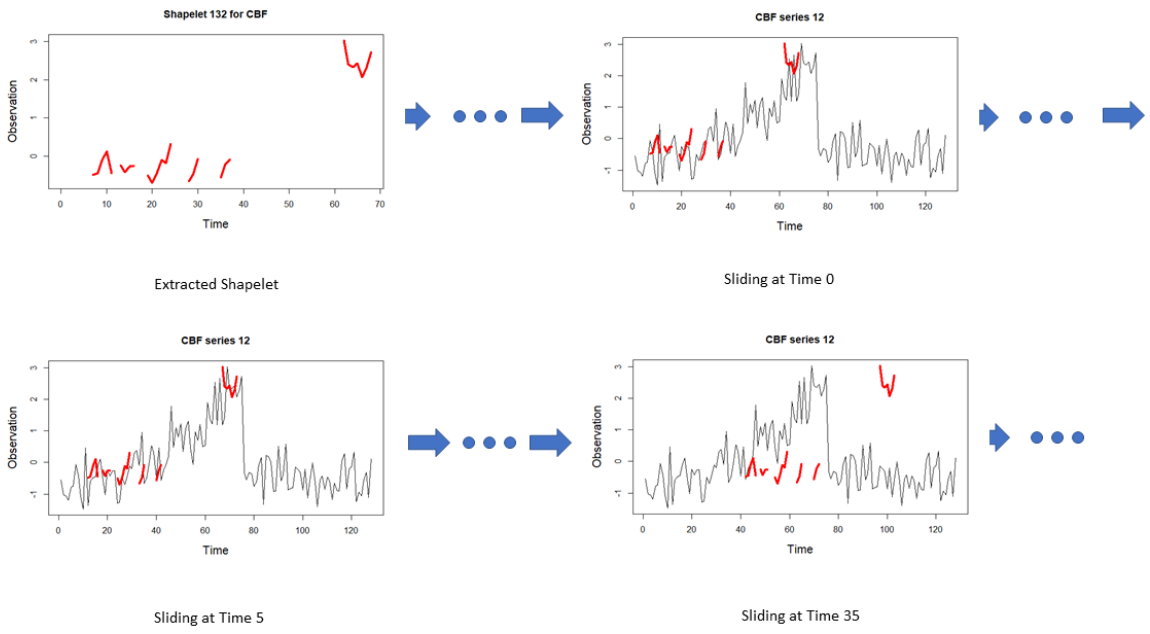


Figure 4.12. Distance Calculation Process.

Extracted shapelets are sliding over both train and test samples to calculate the Euclidean distance between shapelet and the part of the time series with the same length of the shapelet. Therefore, it is obvious that algorithm ends whenever the end point of the shapelet is touched the end of the time series. Therefore, having the start and end point of shapelets long provides another computational advantage. In the literature, short shapelets are desired and the proposed methodology does not conflict that argument because of the missing values. One may observe that proposed shapelet discovery algorithm provides many short sub-shapelets in one shapelet and avoids unnecessary distance calculations.

Once distance calculations are obtained, minimum of these distances is a feature value from the shapelet defined as

$$Dist_{i,k} = \min_{j=1,\dots J} \frac{1}{L} \sum_{l=1}^{L} (T_{i,j+l-1} - S_{k,l})^2, \tag{4.3}$$

where T is time series and S is shapelet. Location of the minimum distance is also a feature value that comes from where the minimum value is calculated as

$$Loc_{i,k} = \arg\min_{j=1,\dots J} \frac{1}{L} \sum_{l=1}^{L} (T_{i,j+l-1} - S_{k,l})^2, \tag{4.4}$$

where T is time series and S is shapelet.

New feature representation is a concatenation of the minimum distance values and location of minimum distances. For example, minimum distance is obtained between time 15 and 23 where 15 is another feature to represent the target series in other words leftmost time index. Since the number of shapelets is dynamic based on the datasets, number of features is also dynamic. Let number of shapelets is K, then the proposed approach guarantees that we will have $2K$ features for each series with the K of them minimum Euclidean distance values and K of them come from the location of these values. Then, the new feature representation is obtained such as in Table 4.1 for classification purpose.

Once the new feature representation is obtained as in Table 4.1, the task is converted to the traditional supervised classification learning problem which aims to predict a class label. Logistic regression, support vector machine (SVM), decision tree and random forest are some popular machine learning algorithms for supervised classification problems. Natively, logistic regression and SVM do not support multiclass classification, they require to transformation into multiple binary classification problems. Decision trees and random forest are better solutions for multiclass classification problems. In this thesis, random forest is chosen as a classifier because of its scalability and stability. Also, the proposed new feature representation contains high number of features. Since random forest is an ensemble learning method, better to predict class labels for datasets with high dimensional features. In additional, variable importance calculation of random forest assures easy interpretation.

Table 4.1. New Feature Representation.

	Distances				Location				
Series	D_1	D_2	...	D_K	L_1	L_2	...	L_K	Class
1	0.01	0.48	...	0.91	10	12	...	25	1
2	0.61	0.34	...	0.28	5	18	...	28	1
...	...	...	...	...	...	...	...	...	
N-2	0.32	0.48	...	0.17	78	38	...	22	3
N-1	0.28	0.55	...	0.25	75	45	...	13	3
N	0.54	0.43	...	0.11	81	58	...	15	2

5. EXPERIMENTS

In this section, the importance of the model selection for the proposed framework is presented by comparing discrete and continuous modeling with experimental settings for feature representation. After new feature representation is obtained, the classification of time series is only supervised classification problem so random forest, a tree-based ensamble method, is preferred for the classification of time series because of its proved success in classification problems. Then the best performing state of the art methods Learning Shapelets (LS) [18], Fast Shapelets (FS) [19], Shapelet Transform (ST) [16] and Generalized Random Shapelet Forest (GRSF) [17] are compared with both choosen discrete and continuous models based on classification accuracy and competitive running time by evaluating datasets from Bagnall *et al.* [20].

PDRD with Decision Tree, Random Forest and GAM are tested on 76 time series data from [26]. The dataset collection is varied in terms of number of class, training size, test size, and length of time series. Sample from 76 time series is seen in Table 5.1.

Table 5.1. Characteristics of the time series datasets.

	Class	Train cases	Test cases	Length
Adiac	37	390	391	176
Arrowhead	3	36	175	251
Beef	5	30	30	470
BeetleFly	2	20	20	512
BirdChicken	2	20	20	512
Car	4	60	60	577
CBF	3	30	900	128
ChlorineConc	3	467	3840	166
Coffee	2	28	28	286

Table 5.1. Characteristics of the time series datasets. (cont.)

	Class	Train cases	Test cases	Length
Computers	2	250	250	720
CricketX	12	390	390	300
CricketY	12	390	390	300
CricketZ	12	390	390	300
DiatomSize	4	16	306	345
DistPhalanxAge	3	400	139	80
DistPhalanxOut	2	600	276	80
DistPhalanxTW	6	400	139	80
Earthquakes	2	322	139	512
ECG200	2	100	100	96
ECG5000	5	500	4500	140
ECGFiveDays	2	23	861	136
FaceAll	14	560	1690	131
FaceFour	4	24	88	350
FacesUCR	14	200	2050	131
FiftyWords	50	450	455	270
Fish	7	175	175	463
FordA	2	3601	1320	500
GunPoint	2	50	150	150
Ham	2	109	105	431
Haptics	5	155	308	1092
Herring	2	64	64	512
InlineSkate	7	100	550	1882
ItalyPowerDemand	2	67	1029	24
LargeKitchen	3	375	375	720
Lightning2	2	60	61	637
Lightning7	7	70	73	319
Mallat	8	55	2345	1024

Table 5.1. Characteristics of the time series datasets. (cont.)

	Class	Train cases	Test cases	Length
Meat	3	60	60	448
MedicalImages	10	381	760	99
MidPhalanxAge	3	400	154	80
MidPhalanxOutline	2	600	291	80
MidPhalanxTW	6	399	154	80
MoteStrain	2	20	1252	84
OliveOil	4	30	30	570
OSULeaf	6	200	242	427
Phalanges	2	1800	858	80
Phoneme	39	214	1896	1024
Plane	7	105	105	144
ProxPhalanxAge	3	400	205	80
ProxPhalanxOut	2	600	291	80
ProxPhalanxTW	6	400	205	80
Refr.Devices	3	375	375	720
ScreenType	3	375	375	720
ShapeletSim	2	20	180	500
SmallKitchen	3	375	375	720
SonyRobot1	2	20	601	70
SonyRobot2	2	27	953	65
Strawberry	2	613	370	235
SwedishLeaf	15	500	625	128
Symbols	6	25	995	398
SyntheticControl	6	300	300	60
ToeSegmentation1	2	40	228	277
ToeSegmentation2	2	36	130	343
Trace	4	100	100	275
TwoLeadECG	2	23	1139	82

Table 5.1. Characteristics of the time series datasets. (cont.)

	Class	Train cases	Test cases	Length
TwoPatterns	4	1000	4000	128
UWaveAll	8	896	3582	945
UWaveX	8	896	3582	315
UWaveY	8	896	3582	315
UWaveZ	8	896	3582	315
Wafer	2	1000	6164	152
Wine	2	57	54	234
WordSynonyms	25	267	638	270
Worms	5	181	77	900
WormsTwoClass	2	181	77	900
Yoga	2	300	3000	426

PDRD is implemented by R programming language, source codes can be found in [27]. Experimentation is done in Windows 10 with 16 GB RAM, AMD Ryzen 5 4600H with Radeon Graphics. Finally, interpretability of proposed method is explained on some of the popular datasets.

Experiments for the shapelet extraction are done by three GAM models with 4, 8 and 16 knots to get the scale-space and shapelets are obtained by using thresholds as 70%, 80% and 90% percentile of all probabilities. The motivation behind building three GAM models is to capture the details by increasing the knots.

Table 5.2 shows the accuracy comparisons of PDRD with decision tree, random forest, GAM and state of the art accuracy results namely, Shapelet Transform (ST), Learning Shapelets (LS), Fast Shapelets (FS), Generalized Random Shapelet Forest (GRSF). The datasets with star (*) represent that the PDRD-GAM experiments are done by the one-vs-rest methodology i.e., for K classes, K-binary GAM models are built.

Table 5.2. Classification accuracies of PDRD, ST, LS, FS and GRSF on the UCR datasets.

	PDRD						
	GAM	**RF**	**DT**	**ST**	**LS**	**FS**	**GRSF**
Adiac*	0.742	0.754	0.555	**0.783**	0.522	0.593	0.726
Arrowhead	0.703	0.651	0.611	0.737	**0.846**	0.594	0.717
Beef	0.700	0.767	0.633	**0.900**	0.867	0.567	0.700
BettleFly	0.800	**0.950**	0.750	0.900	0.800	0.700	0.870
BirdChicken	0.800	0.650	**0.850**	0.800	0.800	0.750	0.810
Car	0.767	0.767	0.700	**0.917**	0.767	0.750	0.843
CBF	0.968	0.969	0.972	0.974	**0.991**	0.940	0.978
ChlorineConc	0.666	0.660	0.570	**0.700**	0.592	0.546	0.669
Coffee	0.964	0.893	0.929	0.964	**1.000**	0.929	0.929
Computers	0.724	**0.756**	0.740	0.736	0.584	0.500	0.739
CricketX*	0.713	0.697	0.677	**0.772**	0.741	0.485	0.771
CricketY*	0.738	0.723	0.700	**0.779**	0.718	0.531	0.756
CricketZ*	0.759	0.677	0.659	**0.787**	0.741	0.464	0.756
DiatomSize	0.915	0.902	0.912	0.925	**0.980**	0.866	0.969
DistPhalanxAge	0.719	0.719	0.734	0.770	0.719	0.655	**0.852**
DistPhalanxOut	0.786	0.772	0.779	0.775	0.779	0.750	**0.826**
DistPhalanxTW*	0.683	0.691	0.676	0.662	0.626	0.626	**0.797**
Earthquakes	0.748	0.748	0.748	0.741	0.741	0.705	**0.818**
ECG200	0.800	0.870	0.850	0.830	**0.880**	0.810	0.828
ECG5000	**0.990**	0.936	0.936	0.944	0.932	0.923	0.940
ECGFiveDays*	0.990	0.818	0.753	0.984	**1.000**	0.998	0.999
FaceAll*	0.753	0.762	0.653	**0.779**	0.749	0.626	0.749
FaceFour	0.784	0.739	0.682	0.852	0.966	0.909	**0.998**
FacesUCR*	0.887	0.882	0.662	0.906	**0.939**	0.706	0.861
FiftyWords*	**0.749**	0.741	0.692	0.705	0.730	0.481	0.723
Fish*	0.851	0.840	0.651	**0.989**	0.960	0.783	0.959

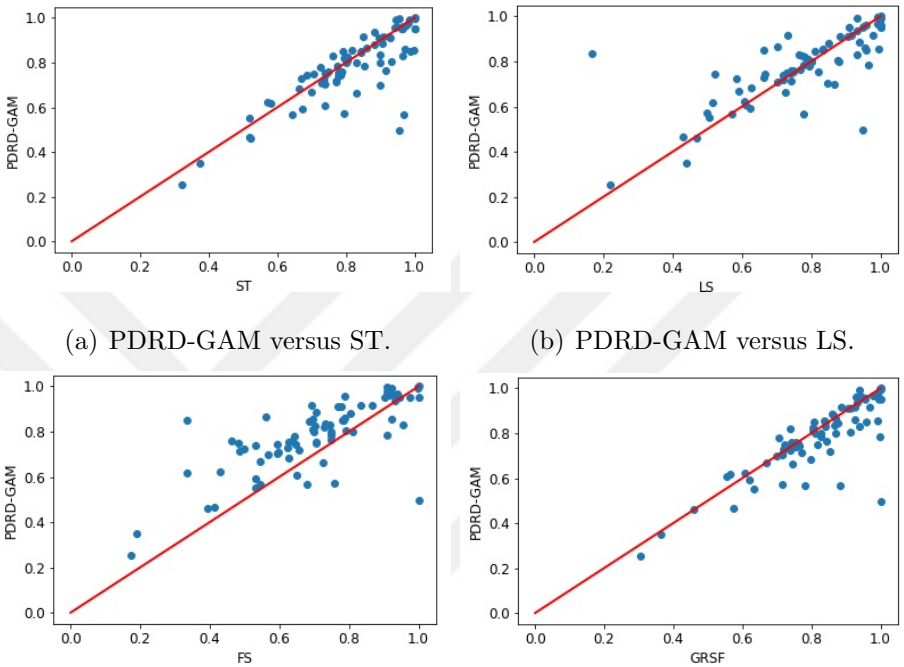
Table 5.2. Classification accuracies of PDRD, ST, LS, FS and GRSF on the UCR datasets. (cont.)

	PDRD						
	GAM	**RF**	**DT**	**ST**	**LS**	**FS**	**GRSF**
FordA*	0.860	0.807	0.770	**0.971**	0.957	0.787	0.927
GunPoint	0.953	0.953	0.953	**1.000**	**1.000**	0.947	0.995
Ham	0.743	0.724	0.762	0.686	0.667	0.648	**0.764**
Haptics*	0.461	0.468	0.419	**0.523**	0.468	0.393	0.460
Herring	0.594	0.578	0.609	**0.672**	0.625	0.531	0.619
InlineSkate*	0.349	0.338	0.267	0.373	**0.438**	0.189	0.365
ItalyPowerDemand	0.962	0.955	**0.967**	0.948	0.960	0.917	0.944
LargeKitchen*	0.864	0.864	0.821	0.859	0.701	0.560	**0.871**
Lightning2*	0.754	**0.820**	0.770	0.738	**0.820**	0.705	0.761
Lightning7*	0.781	0.781	0.781	0.726	**0.795**	0.644	0.707
Mallat*	0.952	0.965	0.957	0.964	0.950	**0.976**	0.940
Meat*	0.917	0.917	**0.933**	0.850	0.733	0.833	0.927
MedicalImages*	**0.730**	0.724	0.689	0.670	0.664	0.624	0.718
MidPhalanxAge	0.565	0.604	0.578	0.643	0.571	0.545	**0.780**
MidPhalanxOutline	0.821	**0.835**	0.794	0.794	0.780	0.729	0.740
MidPhalanxTW*	0.552	0.571	0.597	0.519	0.506	0.532	**0.632**
MoteStrain	0.913	0.897	0.893	0.897	0.883	0.777	**0.914**
OliveOil	0.833	0.800	0.700	**0.900**	0.167	0.733	0.867
OSULeaf*	0.566	0.521	0.517	**0.967**	0.777	0.678	0.883
Phalanges	0.828	0.833	0.809	0.763	0.765	0.744	**0.841**
Phoneme*	0.256	0.207	0.152	**0.321**	0.218	0.174	0.305
Plane*	0.952	0.971	0.971	**1.000**	**1.000**	**1.000**	**1.000**
ProxPhalanxAge	0.849	0.859	**0.863**	0.844	0.834	0.780	0.841
ProxPhalanxOut	0.880	0.856	0.835	**0.883**	0.849	0.804	0.862
ProxPhalanxTW*	**0.824**	0.810	0.785	0.805	0.776	0.702	0.804
Refr.Devices*	**0.619**	0.563	0.547	0.581	0.515	0.333	0.563

Table 5.2. Classification accuracies of PDRD, ST, LS, FS and GRSF on the UCR datasets. (cont.)

	PDRD						
	GAM	**RF**	**DT**	**ST**	**LS**	**FS**	**GRSF**
ScreenType*	0.467	0.456	0.440	0.520	0.429	0.413	**0.573**
ShapeletSim	0.494	0.528	0.567	0.956	0.950	**1.000**	**1.000**
SmallKitchen*	**0.848**	0.835	0.843	0.792	0.664	0.333	0.806
SonyRobot1	0.847	0.790	0.794	0.844	0.810	0.686	**0.878**
SonyRobot2	0.803	0.787	0.806	**0.934**	0.875	0.790	0.912
Strawberry*	0.951	0.951	0.905	**0.962**	0.911	0.903	0.957
SwedishLeaf*	0.910	0.915	0.832	**0.928**	0.907	0.768	0.906
Symbols*	0.936	**0.944**	0.934	0.882	0.932	0.934	0.933
SyntheticControl*	0.977	0.973	0.977	0.983	**0.997**	0.910	0.993
ToeSegmentation1	0.829	0.873	0.851	**0.965**	0.934	0.956	0.940
ToeSegmentation2	**0.915**	0.808	0.800	0.908	**0.915**	0.692	0.885
Trace*	**1.000**	**1.000**	0.980	**1.000**	**1.000**	**1.000**	**1.000**
TwoLeadECG	0.858	0.790	0.610	**0.997**	0.996	0.924	0.992
TwoPatterns*	**0.999**	0.998	0.871	0.955	0.993	0.908	0.998
UWaveAll*	0.956	**0.960**	0.950	0.942	0.953	0.789	0.955
UWaveX*	**0.808**	0.802	0.801	0.803	0.791	0.695	0.804
UWaveY*	0.709	0.714	0.714	**0.730**	0.703	0.596	0.714
UWaveZ*	0.757	0.753	**0.760**	0.748	0.747	0.638	0.742
Wafer*	0.995	0.994	0.991	**1.000**	0.996	0.997	**1.000**
Wine	0.574	0.741	0.630	**0.796**	0.500	0.759	0.715
WordSynonyms*	0.622	**0.625**	0.585	0.571	0.607	0.431	0.607
Worms*	0.610	0.584	0.649	**0.740**	0.610	0.649	0.556
WormsTwoClass	0.662	0.636	0.662	**0.831**	0.727	0.727	0.746
Yoga	**0.855**	0.848	0.818	0.818	0.834	0.695	0.838

In Figures 5.1, 5.2, 5.3 modeling PDRD framework with GAM, RF and DT for time series classification accuracies are compared with the state-of-the-art methods namely ST, LS, FS, GRSF, respectively.



(a) PDRD-GAM versus ST.

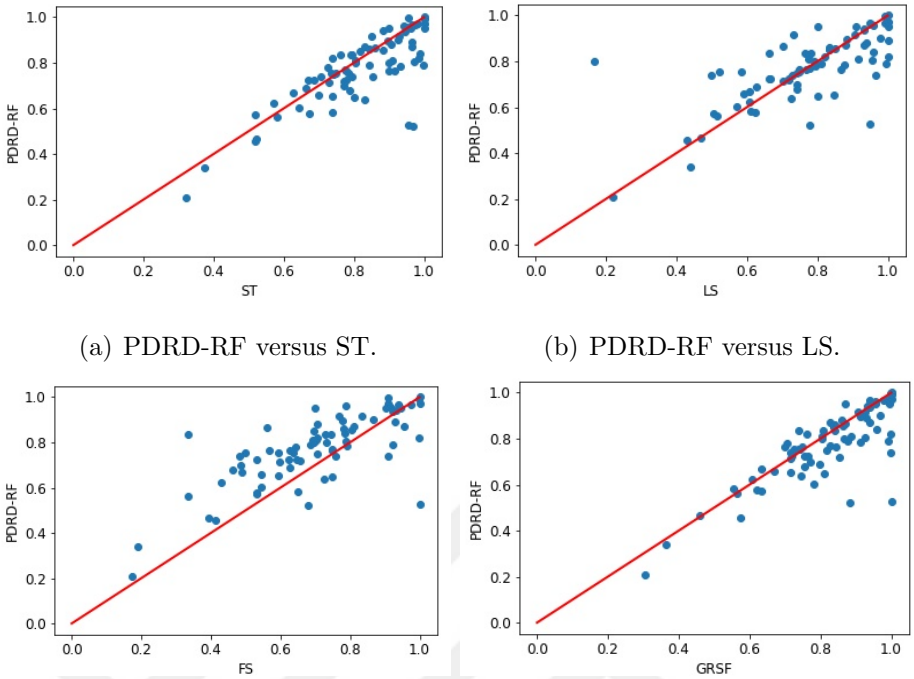
(b) PDRD-GAM versus LS.

(c) PDRD-GAM versus FS.

(d) PDRD-GAM versus GRSF.

Figure 5.1. Classification accuracies of PDRD-GAM versus ST, LS, FS, GRSF.

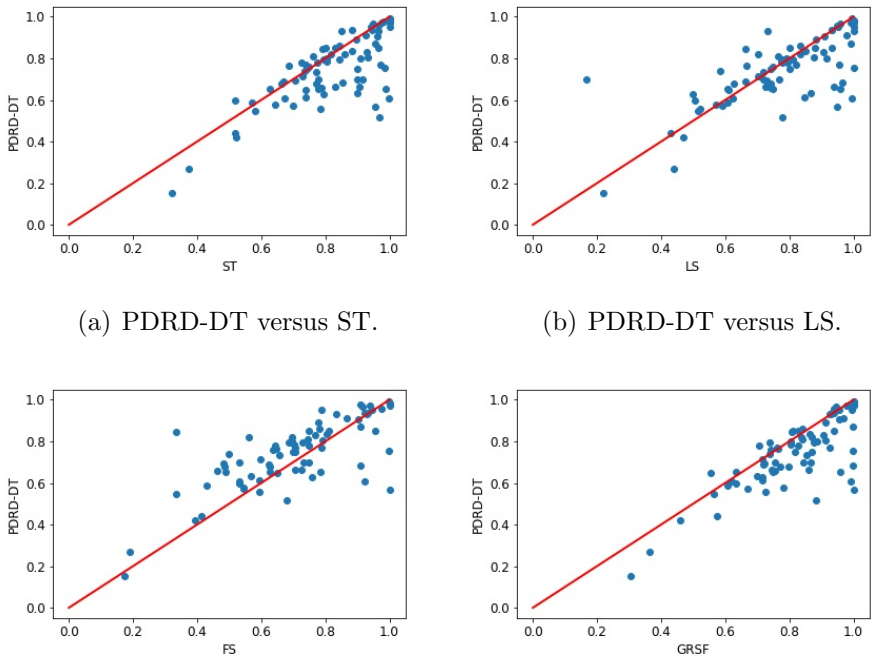
As it is seen that PDRD with GAM, RF and DT is mostly close to the state-of-the-art methods. PDRD with GAM, RF and DT give better resuts most of the datasets when it is compared with FS. For ST and LS comparisons, PDRD framework with GAM is competative. but PDRD with GAM is mostly the same as GRSF and for some of the datasets GRSF is better than the PDRD with GAM as it is seen in Table 5.2.



(a) PDRD-RF versus ST. (b) PDRD-RF versus LS.

(c) PDRD-RF versus FS. (d) PDRD-RF versus GRSF.

Figure 5.2. Classification accuracies of PDRD-RF versus ST, LS, FS, GRSF.

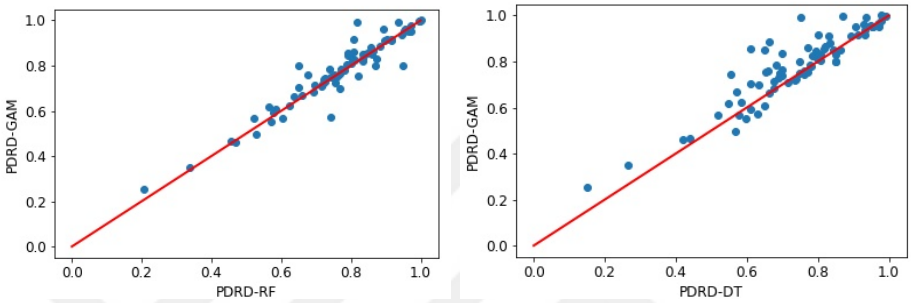


(a) PDRD-DT versus ST. (b) PDRD-DT versus LS.

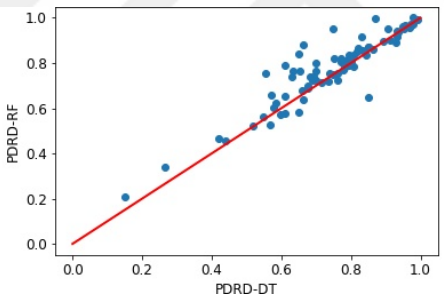
(c) PDRD-DT versus FS. (d) PDRD-DT versus GRSF.

Figure 5.3. Classification accuracies of PDRD-DT versus ST, LS, FS, GRSF.

In Figures 5.4(a), 5.4(b), classification accuracy comparisons between PDRD with GAM and RF and DT are seen, respectively. Modeling PDRD with GAM gives relatively better classification accuracies for most of the datasets than modeling PDRD with RF. PDRD with RF is better than PDRD with DT according to classification accuracies as it is seen in Figure 5.4(c).



(a) PDRD-GAM versus PDRD-RF. (b) PDRD-GAM versus PDRD-DT.



(c) PDRD-RF versus PDRD-DT.

Figure 5.4. Classification accuracies of PDRD methods.

In Table 5.3, shapelet extraction and classification run time (seconds) comparison between PDRD with GAM and RF and DT is seen, respectively. Classification run time beside classification time includes the construction time of the new feature representation for train and test datasets. Although modeling PDRD with GAM gives relatively better classification accuracies for most of the datasets, it is slower than modeling PDRD with RF and DT for both shapelet shapelet discovery and classification time.

Table 5.3. Shapelet and classification run time comparison of PDRD-GAM, PDRD-RF and PDRD-DT.

	Shapelet			Classification		
	GAM	RF	DT	GAM	RF	DT
Adiac	10005.9	64.8	115.0	102.8	48.2	1.2
Arrowhead	361.6	0.5	0.2	1.7	0.3	0.2
Beef	2254.1	0.9	0.2	3.6	0.5	0.2
BettleFly	105.9	0.6	0.1	0.3	0.1	0.1
BirdChicken	75.4	0.6	0.1	0.4	0.1	0.1
Car	3497.4	2.3	0.8	7.5	1.4	0.2
CBF	75.7	0.2	0.1	1.6	0.4	0.3
ChlorineConc	1265.5	9.4	5.2	102.5	20.9	1.2
Coffee	34.3	0.5	0.1	0.3	0.1	0.1
Computers	778.6	13.1	1.7	125.9	12.7	7.3
CricketX*	4489.1	23.2	16.2	94.1	61.7	12.4
CricketY*	4459.2	22.6	17.7	98.8	60.6	11.1
CricketZ*	4494.5	23.1	16.6	112.0	63.6	5.2
DiatomSize	726.3	0.4	0.1	1.0	0.3	0.2
DistPhalanxAge	998.6	3.4	1.1	18.6	0.9	1.1
DistPhalanxOut	549.3	4.8	2.0	39.9	2.3	1.2
DistPhalanxTW*	579.3	4.9	2.2	6.4	2.3	1.3
Earthquakes	1335.1	14.3	2.3	50.3	3.5	0.6
ECG200	44.0	0.7	0.1	0.8	0.2	0.1
ECG5000	80.5	11.4	3.7	5.3	15.0	41.7
ECGFiveDays*	15.1	0.2	0.1	0.7	0.2	0.2
FaceAll*	3022.9	24.7	36.0	92.5	60.5	1.2
FaceFour	660.2	0.5	0.5	0.8	0.2	0.1
FacesUCR*	827.6	6.3	5.5	42.6	22.4	0.3
FiftyWords*	21582.1	340.0	396.0	377.1	81.7	2.1
Fish*	3743.3	10.9	4.3	17.1	14.7	0.1

Table 5.3. Shapelet and classification run time comparison of PDRD-GAM, PDRD-RF and PDRD-DT. (cont.)

	Shapelet			Classification		
	GAM	RF	DT	GAM	RF	DT
FordA*	8987.3	1439.3	422.0	1927.1	532.8	33.2
GunPoint	30.9	0.9	0.1	0.6	0.3	0.2
Ham	329.8	5.4	0.8	8.7	1.6	0.2
Haptics*	2088.0	21.4	5.7	106.8	22.8	0.2
Herring	247.2	3.1	0.3	2.5	0.5	0.1
InlineSkate*	5883.4	20.3	6.6	583.2	347.6	0.3
ItalyPowerDemand	93.3	0.1	0.1	0.3	0.1	0.1
LargeKitchen*	2110.7	25.5	4.0	221.0	29.8	11.3
Lightning2*	509.4	2.6	0.3	4.5	1.0	1.3
Lightning7*	379.9	2.1	0.6	4.2	2.0	0.8
Mallat*	1149.2	5.7	1.7	262.3	128.3	0.3
Meat*	237.1	2.1	0.4	2.0	0.8	0.2
MedicalImages*	878.3	10.9	9.1	17.7	7.5	0.5
MidPhalanxAge	1412.4	3.5	1.3	18.1	1.2	1.0
MidPhalanxOutline	258.3	5.3	2.2	42.0	2.6	1.3
MidPhalanxTW*	789.8	5.4	2.7	6.5	3.1	1.2
MoteStrain	62.4	0.1	0.1	0.4	0.2	0.1
OliveOil	1680.4	2.7	0.3	1.5	0.2	0.1
OSULeaf*	1582.1	12.9	4.7	24.7	15.2	0.2
Phalanges	1907.4	23.2	16.5	565.6	22.6	5.3
Phoneme*	18414.1	221.4	190.2	3267.7	406.1	1.1
Plane*	303.6	1.8	0.8	1.1	1.0	0.1
ProxPhalanxAge	940.7	3.5	1.2	19.2	1.3	1.1
ProxPhalanxOut	254.6	5.6	2.4	38.8	2.5	0.6
ProxPhalanxTW*	440.7	5.3	2.6	6.1	2.0	1.1
Refr.Devices*	1935.1	27.7	4.0	119.6	49.0	22.9

Table 5.3. Shapelet and classification run time comparison of PDRD-GAM, PDRD-RF and PDRD-DT. (cont.)

	Shapelet			Classification		
	GAM	RF	DT	GAM	RF	DT
ScreenType*	1374.9	27.0	4.7	186.8	77.5	33.3
ShapeletSim	138.4	0.8	0.1	0.4	0.3	0.1
SmallKitchen*	1339.8	26.4	4.3	171.1	29.9	12.1
SonyRobot1	10.0	0.1	0.1	0.1	0.1	0.1
SonyRobot2	6.7	0.1	0.1	0.3	0.1	0.1
Strawberry*	931.7	16.2	5.9	33.6	10.3	0.5
SwedishLeaf*	2512.8	21.0	32.2	39.0	31.6	0.8
Symbols*	189.0	0.7	0.2	3.5	4.2	4.0
SyntheticControl*	490.7	3.2	1.4	3.1	2.4	1.6
ToeSegmentation1	66.1	0.7	0.1	1.4	0.5	0.3
ToeSegmentation2	59.2	0.7	0.1	1.3	0.4	0.2
Trace*	153.5	2.1	0.4	1.6	1.1	1.2
TwoLeadECG	9.4	0.1	0.1	0.4	0.2	0.1
TwoPatterns*	2155.2	22.0	19.3	190.2	84.4	2.9
UWaveAll*	13840.4	229.9	270.1	8173.9	4646.9	6.4
UWaveX*	8313.5	56.6	31.8	1374.5	768.6	413.8
UWaveY*	4022.5	56.2	31.1	1239.7	764.4	459.9
UWaveZ*	4104.1	57.6	35.3	1348.5	773.1	287.0
Wafer*	3579.1	20.1	6.7	234.7	29.6	28.2
Wine	138.2	1.0	0.1	0.6	0.2	0.1
WordSynonyms*	5450.4	44.2	39.9	126.5	28.7	0.5
Worms*	5572.0	16.3	5.9	41.5	17.0	0.1
WormsTwoClass	954.7	15.3	2.2	42.3	9.0	0.1
Yoga	994.7	12.5	2.8	304.1	57.0	0.9

Shapelet discovery time comparisons are presented on ChlorineConcentration dataset from Bagnall *et al.* [20]. ChlorineConcentration is a simulated dataset with 3 classes, 467 train cases and 166 length. In Figures 5.5, 5.6 and 5.7, the grid is produced by randomly 20%, 40%, 60%, 80%, 100% of train cases and head of 20%, 40%, 60%, 80%, 100% of time series length. Experiments are done to observe the computational time with changing number of observations and time series length. It is seen that shapelet discovery time increases non-linearly for GAM in Figure 5.5, and linearly for RF and DT in Figures 5.6, 5.7 as time series length and number of observations increase.

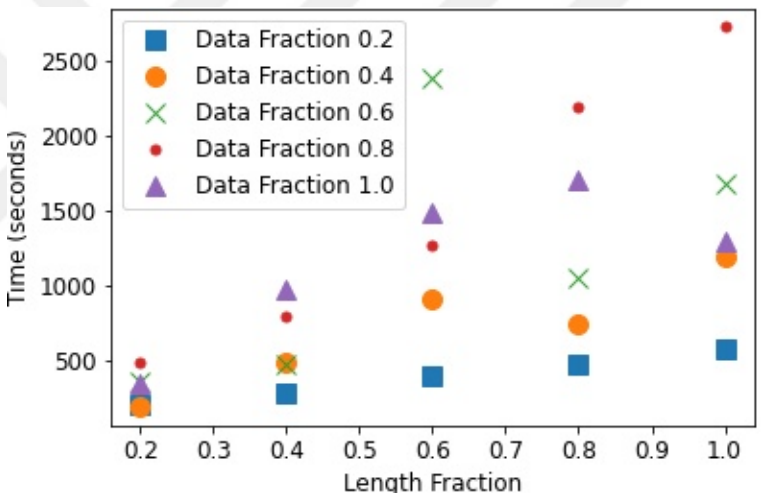


Figure 5.5. Shapelet discovery time by PDRD with GAM.

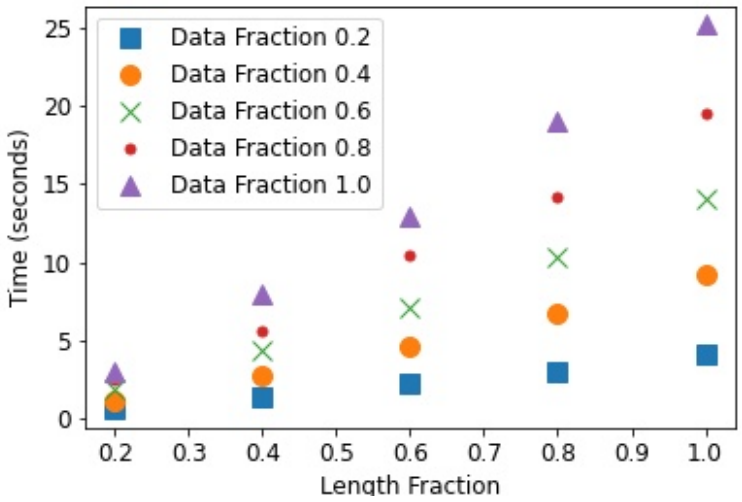


Figure 5.6. Shapelet discovery time by PDRD with RF.

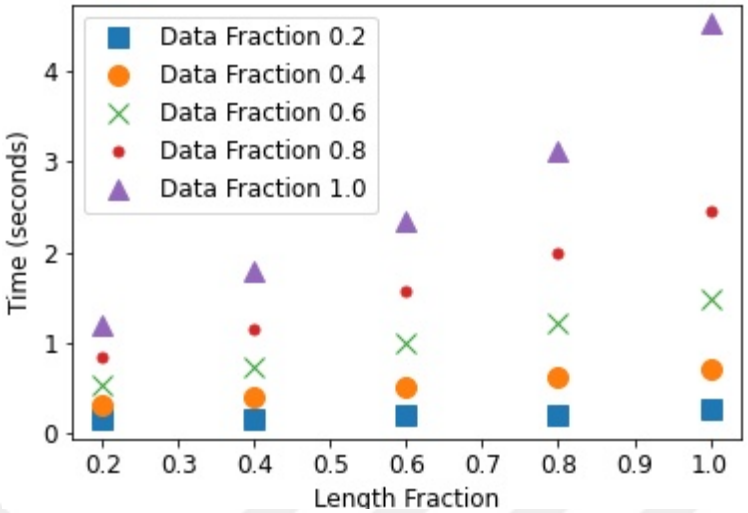


Figure 5.7. Shapelet discovery time by PDRD with DT.

5.1. Interpretation

Interpretability and shapelets of PDRD are illustrated over two popular datasets CBF and GunPoint. CBF is a synthetic dataset with three classes namely, “Cylinder” “Bell” and “Funnel”. Gun-Point is a motion capture time series dataset generated by mapping the motions of two actors with two classes namely, “Gun” and “NoGun”. For the Gun class, the actors “have their hands by their sides, draw a gun from a hip-mounted holster, point it at a target for approximately one second, and then return the gun to the holster and their hands to their sides” [28]. For NoGun class, motion is repeated without a gun.

Random forest classifier performs on CBF dataset by using PDRD with GAM, Random Forest and Decision Tree pipelines with classification accuracies 97% , 93% and 97% , respectively. In Figure 5.8, first 20 important features of random forest based on gini measure on CBF is seen. Y-axis represents the id of shapelets obtained by PDRD with GAM.

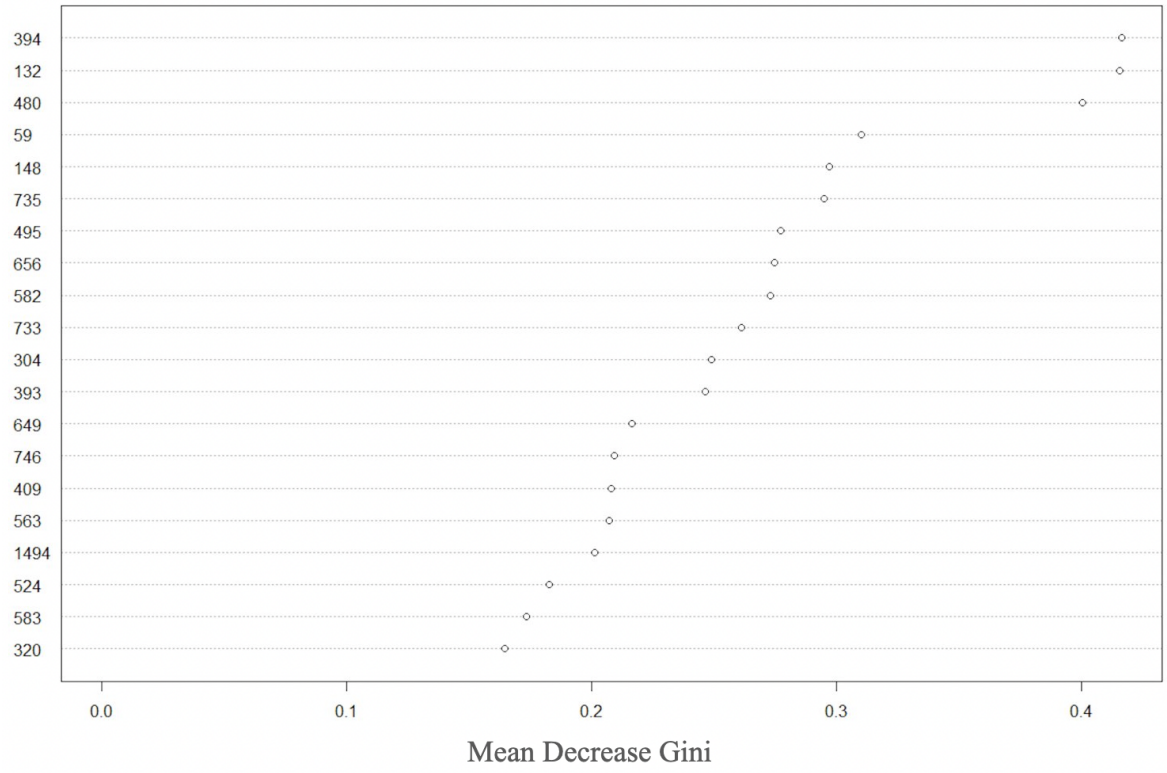
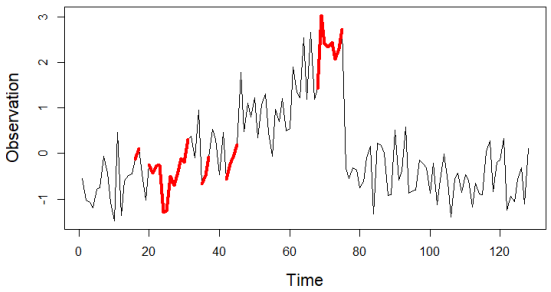
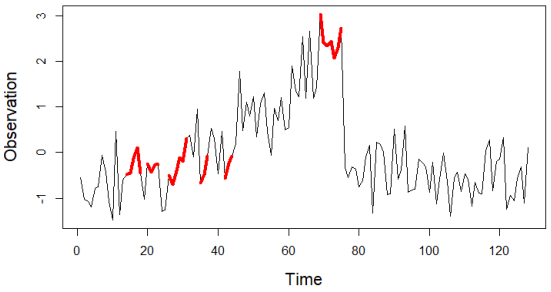


Figure 5.8. Variable importance of random forest based on Gini measure on CBF dataset.

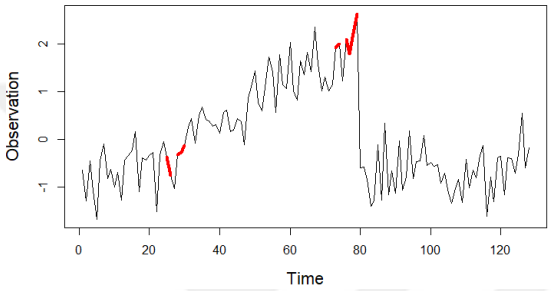
PDRD with GAM pipeline produces 767 shapelets for CBF dataset, so in the new feature representation includes 1534 (767*2) features with the location information. In Figure 5.8, there shapelet ids greater than 767 like id 1150 which means that the location of shapelet id 383 (1150-767). Figure 5.9 shows that first 4 important features of classification. In the new representation, features are obtained by calculating the distance between shapelets and time series and concatenating the location of the minimum distance. Therefore, one can call these features as discriminative 5 shapelets. These shapelets might be correlated, since they are extracted by correlated distances from many time series in the same class.



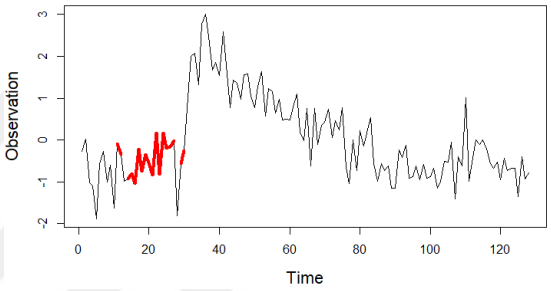
(a) Shapelet id 394 obtained by serie 12 with class bell.



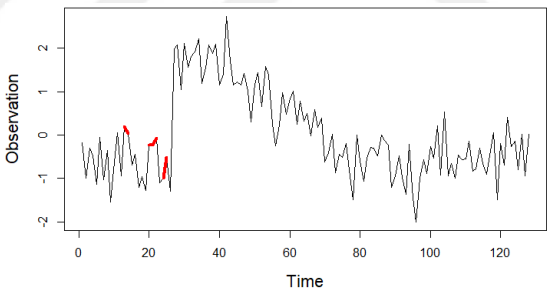
(b) Shapelet id 132 obtained by serie 12 with class bell.



(c) Shapelet id 480 obtained by serie 11 with class bell.



(d) Shapelet id 59 obtained by serie 29 with class funnel.



(e) Shapelet id 148 obtained by serie 28 with class funnel.

Figure 5.9. First 5 important shapelets of PDRD for CBF dataset.

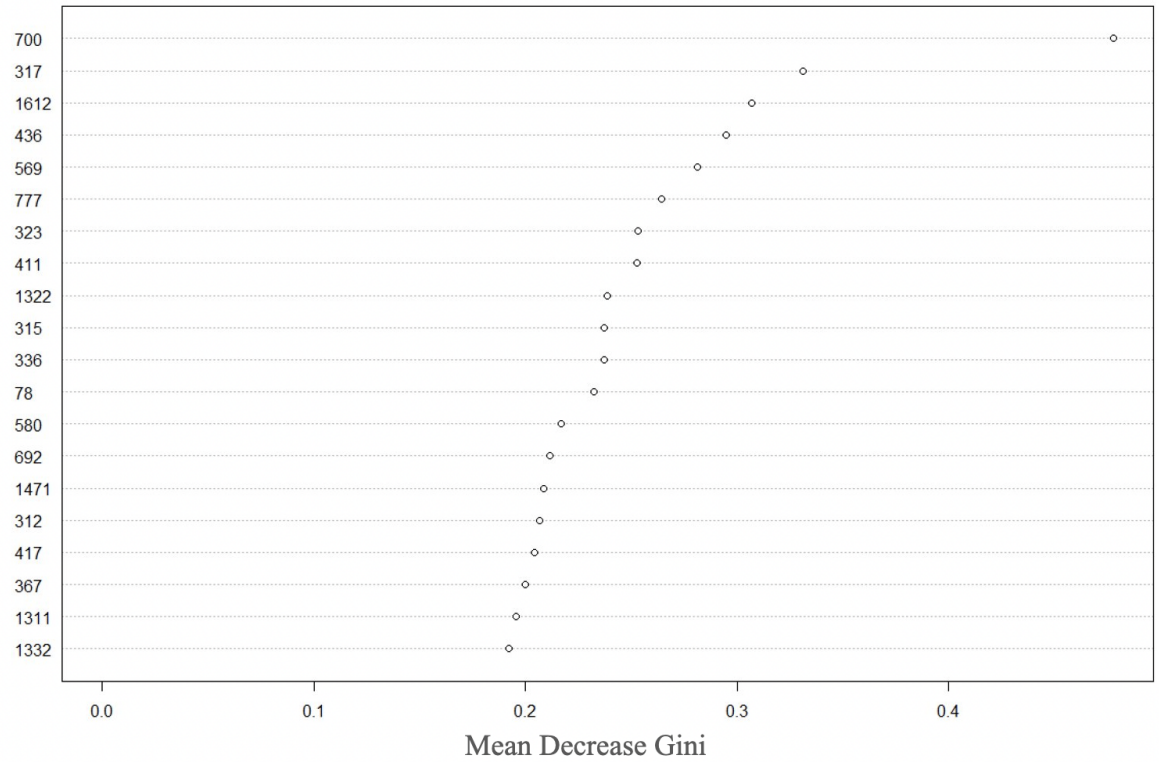
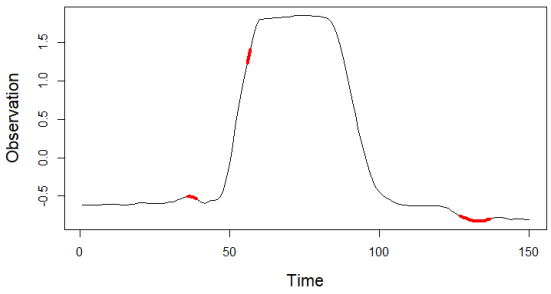


Figure 5.10. Variable importance of random forest based on Gini measure on GunPoint dataset.

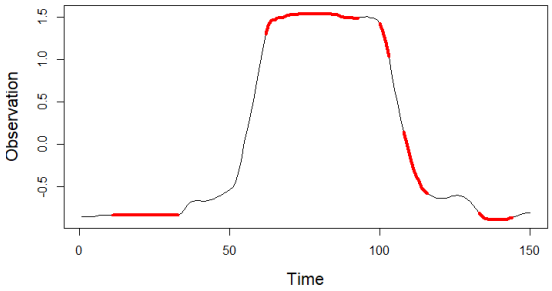
Random forest classifier performs on GunPoint dataset by using PDRD with GAM, Random Forest and Decision Tree pipelines with classification accuracies 95%, 93% and 96%, respectively. In Figure 5.10, first 20 important features of random forest based on Gini measure on GunPoint is seen. Y-axis represents the id of shapelets obtained by PDRD with GAM. PDRD with GAM pipeline produces 806 shapelets for GunPoint dataset, so in the new feature representation includes 1612 (806*2) features with the location information. In Figure 5.10, there shapelet ids greater than 806 like id 1471 which means that the location of shapelet id 665 (1471-806).

Both in CBF shapelets, Figure 5.9, and GunPoint shapelets, Figure 5.11, missing values on shapelets are seen. Missing values also support the logical relationship extracted shapelets from series and classes. It provides computational advantage since sliding these shapelets has early break impact instead of using shorter shapelets. One may also use these shapelets by removing the missing parts and extracting multiple shapelets from one PDRD shapelet. For example in Figure 5.11(a), darker region with

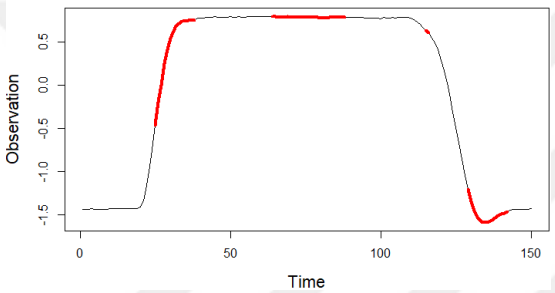
observation between -0.5 and 1.0 and time between 45-50 can be used as one shapelet.



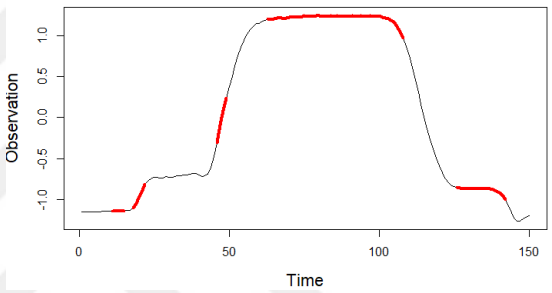
(a) Shapelet id 700 obtained by serie 47 with class gun.



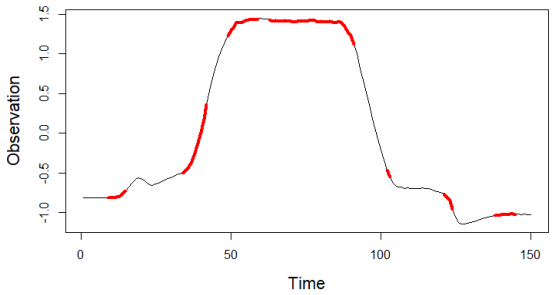
(b) Shapelet id 317 obtained by serie 21 with class gun.



(c) Shapelet id 806 obtained by serie 50 with class nogun.



(d) Shapelet id 436 obtained by serie 42 with class gun.



(e) Shapelet id 569 obtained by serie 11 with class gun.

Figure 5.11. First 5 important shapelets of PDRD for GunPoint dataset.

6. CONCLUSION

In this thesis, flexible local feature extraction framework for time series and shapelet-based time series classification pipeline are proposed. Detecting, describing and matching local features have attracted interest because of reducing the dimension of data by constructing a new representation. Also, shapelet-based classification methods are popular because of their proven success in time series classification.

There are varied methods for extracting local features in text, image and time series domains. Although the methods are specific for the type of dataset, since the main purpose of them is detecting discriminative regions to find a suitable representation, it is possible to connect analogy between them. For example, in image domain the problem is in 2D and the solutions depend on the image quality such as low resolution and distortions. Scale Invariant Feature Transform (SIFT) [3], Speeded-Up Robust Features (SURF) [4] and Local Self-Similarity (LSS) [5] are well-known methods to detect local features on images. The idea behind these methods are matching features and measuring the similarities which is the same as in time series domain. For time series, detecting discriminative regions problem is in 1D and the solutions depend on the time series data quality such as distortions and missing values. These 1D discriminative regions are called as shapelets in time series domain.

Shapelets are used to define a new feature representation by calculating the distance between time series and shapelets with sliding window idea. Feature quality directly affects the classification accuracy. Since the construction of feature representation depends on the shapelet quality, finding the most discriminative shapelets and extracting them efficiently is the main of the shapelet-based time series classification problem. Current state-of-the-art methods require either high computation or give inaccurate results. For example, Learned Shapelets (LS) [18] aims to optimize the logistic cost function to find the best shapelet space. Generalized Random Shapelet Forest (GRSF) [17] prunes the shapelet space by using forest of decision trees. LS and GRSF contain too many parameters and optimizing them computationally heavy pro-

cesses. Shapelet Transform (ST) [16] efficiently prunes the shapelet space with many shapelet candidates and interaction of these candidates. ST is also computationally inefficient, to decrease the computational complexity Fast Shapelet (FS) [19] is proposed. FS randomly takes samples from the shapelet space for classification, it reduces the computational complexity but classification performance is not competitive with the state-of-the-art methods.

In this thesis, in order to detect and interpret the discriminative regions by using time-observation space, Probabilistic Discriminative Region Descriptor (PDRD) is proposed. Also, the proposed descriptor is used for shapelet-based time series classification purpose. Extracting discriminative regions only depends on a model which can produce probabilistic outputs such as decision tree, random forest and GAM. Once class probability estimations are obtained, visualization and interpretation of the time-observation space are very easy and logical. For the shapelet extraction, class probability estimations for each time series are filtered by a given threshold. Then, new feature representation is calculated by using Euclidean distance between shapelets and time series with sliding window approach. Adding location information is also optional as a parameter, but the experiments show that adding it gives more precise results. After new feature representation is obtained, any classifier can be used for the classification. In this thesis, random forest is preferred for the classification step because of its proven success in classification problems. PDRD with decision tree, random forest and GAM achieve the state-of-the-art classification accuracy with a very low computational requirements.

REFERENCES

1. Kakizawa, Y., R. H. Shumway and M. Taniguchi, "Discrimination and Clustering for Multivariate Time Series", *Journal of the American Statistical Association*, Vol. 93, No. 441, pp. 328–340, 1998.

2. Khadra, L., A. Al-Fahoum and S. Binajjaj, "A Quantitative Analysis Approach for Cardiac Arrhythmia Classification Using Higher Order Spectral Techniques", *IEEE Transactions on Bio-Medical Engineering*, Vol. 52, No. 1, pp. 1840–1845, 2005.

3. Lowe, D. G., "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91–110, 2004.

4. Bay, H., A. Ess, T. Tuytelaars and L. Van Gool, "Speeded-Up Robust Features (SURF)", *Computer Vision and Image Understanding*, Vol. 110, No. 3, pp. 346–359, 2008.

5. Shechtman, E. and M. Irani, "Matching Local Self-Similarities across Images and Videos", *IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, USA, 2007.

6. Beggel, L., B. X. Kausler, M. Schiegg, M. Pfeiffer and B. Bischl, "Time Series Anomaly Detection Based on Shapelet Learning", *Computational Statistics*, Vol. 34, No. 3, pp. 945–976, 2019.

7. Keogh, E., L. Wei, X. Xi, S.-H. Lee and M. Vlachos, "LB-Keogh Supports Exact Indexing of Shapes under Rotation Invariance with Arbitrary Representations and Distance Measures", *Proceedings of the 32nd International Conference on Very Large Data Bases*, Seoul, Korea, 2006.

8. Keogh, E. and C. A. Ratanamahatana, "Exact Indexing of Dynamic Time Warp-

ing", *Knowledge and Information Systems*, Vol. 7, No. 3, pp. 358–386, 2005.

9. Lin, J., E. Keogh, L. Wei and S. Lonardi, "Experiencing SAX: A Novel Symbolic Representation of Time Series", *Data Mining and Knowledge Discovery*, Vol. 15, No. 1, pp. 107–144, 2007.

10. Lin, J., R. Khade and Y. Li, "Rotation-Invariant Similarity in Time Series Using Bag-of-Patterns Representation", *Journal of Intelligent Information Systems*, Vol. 39, No. 2, pp. 287–315, 2012.

11. Baydogan, M. G. and G. Runger, "Learning a Symbolic Representation for Multivariate Time Series Classification", *Data Mining and Knowledge Discovery*, Vol. 29, No. 2, pp. 400–422, 2015.

12. Edali, M., M. G. Baydogan and G. Yucel, "Classification of Generic System Dynamics Model Outputs via Supervised Time Series Pattern Discovery", *Turkish Journal of Electrical Engineering; Computer Sciences*, Vol. 1, No. 1, p. 832–846, 2019.

13. Baydogan, M. G., G. Runger and E. Tuv, "A Bag-of-Features Framework to Classify Time Series", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 35, No. 11, pp. 2796–2802, 2013.

14. Liang, Z. and H. Wang, "Efficient Class-Specific Shapelets Learning for Interpretable Time Series Classification", *Information Sciences*, Vol. 570, No. 1, pp. 428–450, 2021.

15. Ye, L. and E. Keogh, "Time Series Shapelets: A New Primitive for Data Mining", *The 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Paris, France, 2009.

16. Hills, J., J. Lines, E. Baranauskas, J. Mapp and A. Bagnall, "Classification of Time Series by Shapelet Transformation", *Kluwer Academic Publishers*, Vol. 28, No. 1,

pp. 851–881, 2013.

17. Karlsson, I., P. Papapetrou and H. Bostrom, "Generalized Random Shapelet Forests", *Kluwer Academic Publishers*, Vol. 30, No. 5, p. 1053–1085, 2016.

18. Grabocka, J., N. Schilling, M. Wistuba and L. Schmidt-Thieme, "Learning Time-Series Shapelets", *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2014.

19. Rakthanmanon, T. and E. Keogh, "Fast Shapelets: A Scalable Algorithm for Discovering Time Series Shapelets", *Proceedings of the SIAM International Conference on Data Mining (SDM)*, Austin, TX, USA, 2013.

20. Bagnall, A., J. Lines, A. Bostrom, J. Large and E. Keogh, "The Great Time Series Classification Bake Off: A Review and Experimental Evaluation of Recent Algorithmic Advances", *Data Mining and Knowledge Discovery*, Vol. 31, No. 3, pp. 606–660, 2017.

21. Han, H., X. Guo and H. Yu, "Variable selection using Mean Decrease Accuracy and Mean Decrease Gini based on Random Forest", *7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, China, 2016.

22. Hastie, T. and R. Tibshirani, "Generalized Additive Models", *Statistical Science*, Vol. 1, No. 3, pp. 297–310, 1986.

23. Wood, S., *Generalized Additive Models: An Introduction With R*, Chapman and Hall/CRC, New York, NY, USA, 2006.

24. Niennattrakul, V. and C. A. Ratanamahatana, "On Clustering Multimedia Time Series Data Using K-Means and Dynamic Time Warping", *International Conference on Multimedia and Ubiquitous Engineering (MUE'07)*, Seoul, South Korea, 2007.

25. Bostrom, A. and A. Bagnall, "Binary Shapelet Transform for Multiclass Time Series Classification", *Big Data Analytics and Knowledge Discovery*, Cham, Germany, 2015.

26. Vicker, W., "Time Series Classification", `http://timeseriesclassification.com`.

27. Süngü, P. and M. Baydoğan, "PDRD Implementation in R", 2022, `https://github.com/psungu/PDRD`, accessed on August 13, 2022.

28. Ye, L. and E. Keogh, "Time Series Shapelets: A Novel Technique That Allows Accurate, Interpretable and Fast Classification", *Data Mining and Knowledge Discovery*, Vol. 22, No. 1, pp. 149–182, 2011.

29. Chang, K.-W., B. Deka, W.-M. W. Hwu and D. Roth, "Efficient Pattern-Based Time Series Classification on GPU", *IEEE 12th International Conference on Data Mining*, Brussels, Belgium, 2012.

30. Liang, Z. and H. Wang, "Efficient Class-Specific Shapelets Learning for Interpretable Time Series Classification", *Information Sciences*, Vol. 570, No. 1, pp. 428–450, 2021.

31. Marteau, P., "Time Warp Edit Distance", *CoRR*, Vol. abs/0802.3522, No. 1, 2008.