

# LEARNING DISCRIMINATIVE VIRTUAL SEQUENCES FOR TIME SERIES CLASSIFICATION

by

ABHILASH YESHWANT DORLE

(Under the Direction of Sheng Li)

## ABSTRACT

Temporal data are continuously collected in a wide range of domains. Time series classification, as an essential task in time series analysis, aims to assign a set of temporal sequences to different categories. Time series classification using distance metric learning such as the virtual sequence metric learning (VSML), have achieved remarkable performance, where virtual sequences attract samples from different classes to facilitate classification. However, the existing VSML methods simply employ fixed virtual sequences, which might not be optimal for the subsequent classification tasks. To address this issue, we propose a novel time series classification method, discriminative virtual sequence learning (DVSL). Following the framework of sequence metric learning, our DVSL method jointly learns a set of discriminative virtual sequences that help separate time series samples in a feature space and optimizes the temporal alignment by DTW measure. Extensive experiments on UCR datasets demonstrate the efficiency of DVSL, compared to several baselines.

INDEX WORDS: Sequence data, Time Series, Machine Learning

LEARNING DISCRIMINATIVE VIRTUAL SEQUENCES FOR TIME SERIES CLASSIFICATION

by

ABHILASH YESHWANT DORLE

B.Tech.,SRM University, 2016

A Dissertation Submitted to the Graduate Faculty  
of The University of Georgia in Partial Fulfillment

of the

Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2020

©2020

Abhilash Y. Dorle

All Rights Reserved

LEARNING DISCRIMINATIVE VIRTUAL SEQUENCES FOR TIME SERIES CLASSIFICATION

by

ABHILASH YESHWANT DORLE

Approved:

Major Professor: Sheng Li

Committee: In Kee Kim  
Le Guan

Electronic Version Approved:

Ron Walcott  
Interim Dean of the Graduate School  
The University of Georgia  
August 2020

*Dedicated to my mother, father and sister  
and my lovely friends Rahul Singh and Bhavana Bora*

# Acknowledgments

I would like to thank my advisor, Dr. Sheng Li for his help and guidance with this work. His expertise and insight helped me deal with problems which otherwise would have proved very challenging. His advice and motivation during our group meetings proved to be vital in helping me finish this work. I would also like to thank Dr. In Kee Kim whose expertise and knowledge in the domain of time series helped me streamline the project. His course about Cloud Computing and its association of Machine Learning helped me gain significant insight in my field. I would like to thank Dr. Le Guan for his valuable insights. His expertise in the security domain helped me in my work.

My colleagues at RISE lab- Heng-Shiou Sheiou, Seyedsaed Rezayidemne, Ronghan Zhu, Sumer Singh, Weifeng Wang and Brij Rokad also deserve a thanks for their help and company. Working with them and having fun conversations with them, both philosophical and other, made my time at UGA a pleasant experience.

My parents, sister, Mayur and my cousin, Prateek were constantly by my side throughout my journey. Their words of encouragement and belief in my abilities gave me the strength to push through some of the more stressful times in my research. I would also like to thank my friends Lahar Mishra, Akshay Agashe, Akshay Kokane and Santosh Bobade who supported me throughout my Masters degree.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Related Work</b>	<b>5</b>
<b>3</b>	<b>Our Method</b>	<b>8</b>
3.1	Definitions . . . . .	8
3.2	Virtual Sequence . . . . .	9
3.3	Virtual Sequence based Time Series Classification . . . . .	10
3.4	Shapelets as Virtual Sequences . . . . .	12
3.5	Learning Virtual Sequences . . . . .	12
3.6	Optimization . . . . .	17
<b>4</b>	<b>Experiments</b>	<b>19</b>
4.1	Datasets . . . . .	19
4.2	Experimental Settings . . . . .	20
4.3	Results and Analysis . . . . .	21
<b>5</b>	<b>Conclusions</b>	<b>26</b>

# List of Figures

3.1	(a)Classical. (b) Virtual Sequence approach. . . . .	9
3.2	(a) First Virtual Sequence. (b) Second Virtual Sequence. . . . .	11
3.3	Time Series from the UCR time series archive. Cylinder-Bell-Funnel is a simulated data set. X-axis represent time stamps and Y-axis represent values of the time series . . . . .	13
3.4	Shapelets extracted from CBF dataset. X-axis represent time stamps and Y-axis represent values of the time series . . . . .	13
3.5	Framework of the proposed DVSL method. The input contains time series samples and randomly generated virtual sequences (VS). During training, our method jointly learns discriminative VS that are well separated, aligns the training samples with the VS, and optimizes the ground metric for classification purpose. . . . .	15
4.1	Convergence Curve . . . . .	25
4.2	Parameter Sensitivity . . . . .	25

# List of Tables

3.1	Summary of notations . . . . .	11
4.1	Dataset description . . . . .	20
4.2	Hyperparameter range . . . . .	20
4.3	Train/Test split for the dataset . . . . .	21
4.4	Results for DVSL method . . . . .	22
4.5	Results for Shapelet as Virtual Sequence . . . . .	23
4.6	Results for varied Shapelet Length . . . . .	23
4.7	Times series classification results of the proposed DVSL method and baselines on UCR datasets. Dataset names have been made short. . . . .	24

# Chapter 1

## Introduction

The problem of Time Series Classification has attracted many researchers because of an inherent existence of time series data in several domains. The increase in the availability of such data has further motivated the investigation of this problem. Time series data is prevalent across many diverse domains such as medicine, finance and entertainment. To name a few examples, data readings from IoT devices could help monitor the status of electrical systems; body wearable devices can provide insightful information about human activities and behavior; surveillance cameras deployed in transit systems could help address public security concerns.

A rapid growth in the technology world has augmented the availability of time series data and has largely driven the research efforts on time series analysis in recent years. Some fundamental time series analysis tasks include classification, prediction, anomaly detection, clustering and visualization. Time series classification, as the primary task in this area, aims to separate temporal sequences into some predefined categories. With such large amount of data, it is very important for machine learning algorithms to perform classification efficiently while having a low time complexity.

Several algorithms have been proposed to tackle the problem of time series classification

since the 1990's. A great deal of improvement was observed when Dynamic Time Warping (DTW) was inculcated in these algorithms. Since then researchers have proposed numerous algorithms that involve DTW. An array of techniques, that do not implement DTW, have also produced exceptional results. These numerous algorithms can be roughly categorized as local feature methods, deep learning based methods and metric learning based methods.

1. **Local Feature based methods:** These methods try to extract local representative attributes from the time series sample and then perform classification based on these attributes.
2. **Deep Learning based methods:** These methods usually employ a deep learning architecture to learn a latent space, where the classification is efficient
3. **Metric Learning methods:** These methods try and bring samples of the same classes closer to each other while pushing apart samples from other classes, by learning a metric in a different dimension.

Several methods have been proposed that can be designated in one of the categories. Sometimes algorithms try and combine techniques from more than one category to perform classification. A novel method presented in [20], proposed Shapelets. Shapelets, defined as time series sub sequences that represent a class maximally, provided better interpretation than most of the algorithms. Such a method falls into the first category and utilizes local features to perform classification. Algorithms proposed in [11] and [9] utilize deep learning architectures to produce some interesting results. Metric learning has been one the most efficient and simplest methods that perform the classification task efficiently. These methods have also been implemented on sequence data, an example is data from video streams, and have produced some exceptional results. Most of the metric learning techniques look into methods like kNN, whose performance heavily depend upon the metric employed. A lot

of researchers are now trying to improve the efficiency and reduce the time complexity of the classification. This led to development of various algorithms that implemented different metric systems and improve the classification.

Inspired by *Regressive Virtual Sequence Metric Learning* in [18], this thesis proposes a novel method for classification of time series data. This work has the following contributions:

1. **Virtual sequence using Shapelets:** The proposed method generates virtual sequences by using the data values of shapelets, instead of employing combination of 0's and 1's. Since shapelets define a class maximally such virtual sequences separates the samples from each other distinctly.
2. **Novel loss function:** The new loss function implemented brings data samples closer to an associated virtual sequence, while pushing away samples from other classes. The loss function inculcates the novel idea of learning the virtual sequences. The function also takes into consideration alignment of the samples.
3. **Learning virtual sequences:** We propose an algorithm which learns the virtual sequences. Predefined virtual sequences may not ideal for any given tasks, which serves as motivation to learn virtual sequences. Such virtual sequences are learnt specific to given tasks and are discriminative in nature.
4. **Separating virtual sequences:** While we propose a new loss function, we incorporate a term that separates out the virtual sequences from each other, while advocating the metric learning principal.

This chapter introduced the problem of Time Series Classification and the contribution of this thesis. Rest of this work is organized as follows. Chapter 2 describes the related work. We develop a model, closely associated with virtual sequence learning, which is presented in this chapter. Chapter 3 describes our method in detail. Chapter 4 describes the details

of our experiments and provides some analysis on the results. We conclude this thesis in Chapter 5, where we highlight our contribution and discuss future work.

# Chapter 2

## Related Work

Time Series analysis finds its applications in several fields. For instance in the area of cloud computing, recently released cloud traces can be used for estimating the total resource usage. In [4, 17, 8] extensive study of such traces released by Google, Alibaba and Azure have been conducted. In the field of IoT security, time series analysis can be used to identify cyber-physical attacks, [14] shows a study in this aspect.

One of the critical task in Time series analysis is Time Series classification, a coveted topic among researchers. Algorithms for this purpose have been proposed since last two decades. Time series classification aims to classify time series samples into predefined categories. Methods proposed either use inherent characteristics of the time series samples or develop a similarity among the samples of same classes, which is considerably different for samples of other classes.

There has been a consistent improvement in the algorithms and techniques used for classification. The task of categorizing the data into predefined categories is a challenging one. In the case of Time Series classification, the challenge is to consider the sequential and ordered form of the data which also produces new avenues of research. A lot of research has been done in classification, forecasting and clustering of time series data. In this chapter,

we talk about different methods and research conducted in this field. We further point out how our work is different and the unique contributions we make.

As the availability of time series data increased, as did the research in the field. A lot of methods have been proposed for time series classification, including local feature based methods [20], deep learning based methods[11] and metric learning based methods[18].

Local feature based methods work with local features of the time series to perform classification. Certain representative time series sub sequences are extracted in [20], these sub sequences are called Shapelets. Once the extraction is complete, the sub sequences are compared to the training sample, by traversing the sample using sliding window technique. Several distance based methods have also been proposed to tackle the classification problem. Most of the subsequent work in this field has been focused on extracting Shapelets or learning Shapelets. [6, 10]. Researchers usually employ a similarity or dissimilarity measure to perform classification. Dynamic time Warping, DTW is one such measure that is widely used. Several variations of the DTW measure have been proposed. It has been combined with Support Vector Machines to obtain better results in [7]. DTW has been extensively applied for the task but has a higher time complexity. In [13] a new method has been proposed that outperforms DTW by one or two orders of magnitude with no loss in accuracy.

Methods that employ deep learning architecture have produced exceptional results. The latent space learnt, separates the samples from each other distinctively. In [11], the authors propose a framework that combines with distance learning methods and perform classification under data scarcity. The method projects the time series sample in a lower dimension by modeling evolving patterns. [12] implement a Long Short Term Memory (LSTM) network for the task of classification. Specifically they employ a Long Short Term Memory Fully Convolutional Network to achieve remarkable results.

Lately there have been significant improvements in the field of distance metric learning, which is elemental to problems in data mining. Metric learning aims to transfer the samples

into a new distance space where the samples of different classes are distinctly apart from each other. [1] provides a review on time series classification and how metric learning methods have been efficient. [3] proposes a model that considers samples from other classes to perform classification. [15] depicts the usage of Hamming distance for classification, demonstrating the efficiency of metric learning methods.

Most recently, a virtual sequence metric learning (VSML) framework for time series classification is proposed in [18]. This framework employs a set of virtual sequences that are inspired by the idea of virtual points [16]. [16] proposes a metric learning method that brings closer each data point to a predefined virtual point. It also depicts how the solution can be kernelized. This research has been extended in [18]. They propose a system of developing a virtual sequence for different number of classes. Their work proposes to learn a ground Mahalanobis metric that can incorporate a number of distance measures. The virtual sequence captures various temporal structures of the sequences via combination of identity and zero matrix. In this framework, virtual sequences are predefined in the sequential data space, and the objectives of metric learning and temporal alignment can be jointly optimized. In particular, this framework brings samples from each class to a class-specific virtual sequence, such that the time series samples from different classes can be separated.

Although this method has obtained quite promising results on time series classification [18], it still has some limitations. First, the performance of this method heavily relies on the quality of predefined virtual sequences. Since virtual sequence construction and time series classification are two isolated steps, such virtual sequences may not be optimal for the subsequent classification task. Second, the design of virtual sequences is very subjective, which may not fit various downstream applications in practice. To overcome this, we propose a novel method to learn virtual sequences to perform classification.

# Chapter 3

## Our Method

We develop a metric learning method to perform classification of time series. We model a method that implements virtual sequence metric learning, where by we learn the virtual sequences that aid in the classification process. We have also utilised the concept of shapelets to develop virtual sequences.

In this chapter we provide some background on time series, shapelets and virtual sequences. We begin by introducing some general concepts and then move to state some specific details that are relevant to our work. In later chapters, we will use the same concepts to showcase our model.

### 3.1 Definitions

**Definition 1. *Time Series.*** A time series  $X^n = [x_1, x_2, \dots, x_L]$  is an ordered sequence of data, where  $x_t$  is the value of the time series at time stamp  $t$ , and  $L$  is the length of the time series. The label of the time series  $X^n$  is denoted as  $y^n$ .

**Definition 2. *Shapelets.*** Shapelets are time series subsequences that maximally define a class

**Definition 3. Virtual Sequence [18].** A virtual sequence  $V^n = [v_1, \dots, v_m] \in R^{b' \times l^n}$  is defined as a function of  $X^n$  and  $y^n$ , i.e.,  $V^n = f(X^n, z^n)$ .

**Definition 4. Time series classification.** Time series classification is the task, whereby the algorithm assigns time series samples into predefined categories.

## 3.2 Virtual Sequence

Distance metric learning has been studied extensively for a long time. It aims to bring closer samples from same class and separate out samples from other classes. These methods learn metrics that improve the output of algorithms. A recent study [16] learns a Mahalanobis metric to perform classification. The algorithm proposed, instead of bringing samples of same class closer to each other while pushing samples from other classes away, brings samples closer to a predefined virtual point. The technique aims to move data points relative to the virtual points. This can be visualised in the Figure 3.1

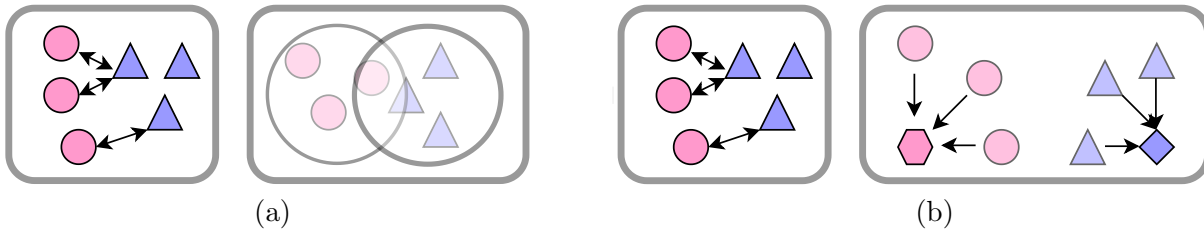


Figure 3.1: (a) Classical. (b) Virtual Sequence approach.

This was extended in [18]. The study aims to classify sequence data by unifying a number of distance measures with the help of Mahalanobis distance. The algorithm works with a predefined virtual sequence. It brings closer a sequence, of a particular class, closer to the virtual sequence of that particular class.

Our method is closely related to the virtual sequence metric learning method [18], which jointly learns the ground metric and aligns the time series samples with virtual sequences. However, the virtual sequences in [18] are predefined and also fixed during model training.

The length of the virtual sequence is set to 1 ensuring that the alignment between sequences is unique. As a result, the model performance highly depends on the construction of virtual sequences. Different from existing work, our method adaptively learns a set of discriminative virtual sequences for time series classification, which could be easily adapted to time series data in different domains.

### 3.3 Virtual Sequence based Time Series Classification

*Regressive Virtual Metric Learning (RVML)* [16] introduced the idea of virtual points. They are predefined points, where each training sample was brought close to these virtual points reducing the number of constraints in classification.

*Learning Distance for Sequences by Learning a Ground Metric*[18] extended this concept by introducing virtual sequences. Every sequence can be identified by a temporal structure. The virtual sequences can be thought of stages among these temporal structures. something like poses of an action. The method also takes into consideration the alignment of the sequences simultaneously.

A virtual sequence is generated by the combination of 0's and 1's. The combination involves an identity matrix for the class under consideration and zero matrix for the other. Virtual sequence for "GunPoint", a data set in the UCR archive with 2 classes and 3 temporal structure is depicted in Figure 3.2. Temporal structure is a hyperparameter

In this section we explore the model developed for time series classification by learning discriminative virtual sequences. Method proposed in [18] serves as a motivation for our model. [18] makes use of predefined virtual sequences for the purpose of classification. These predefined virtual may not be suitable for every task. Hence we devise a method to learn the virtual sequences that are specific to the given task.



Figure 3.2: (a) First Virtual Sequence. (b) Second Virtual Sequence.

Notation	Explanation
$T$	Matrix indicating correspondence relationship
$t_{ij}$	Element of matrix $T$ at the position (i,j)
$W$	Parameter matrix
$x$	Denotes the training sequence
$v$	Virtual sequence
$N$	Total number of training sequence
$L^n$	Length of nth sequence
$l^n$	Number of temporal structures
$\lambda$	Regularization parameter

Table 3.1: Summary of notations

The loss function proposed in [18] is given in Eq. 3.1. The notations used are summarized in Table 3.1

$$\sum_{n=1}^N \frac{1}{N} \langle T^n, D_I^n(W) \rangle + \lambda \|W\|_F^2 = \sum_{n=1}^N \sum_{i=1}^{L^n} \sum_{j=1}^{l^n} \frac{1}{N} t_{ij}^n \|W^T x_i^n - v_j^n\|_2^2 + \lambda \|W\|_F^2 \quad (3.1)$$

### 3.4 Shapelets as Virtual Sequences

Shapelets are discriminative sub sequences that aid in the classification task. We utilize such sub sequences to generate virtual sequences. The entire goal of the virtual sequence is to separate the training sequences efficiently. Virtual sequence for one class should be able to separate samples of that class from samples of other classes.

With this regard we propose generation of virtual sequences using Shapelets. Shapelets are time series subsequences that define a particular class maximally. Instead of implementing a combination of 1's and 0's in the virtual sequence, we use a combination of actual values of the shapelets and 0's. We select the hyperparameter  $m$  as the number of shapelets and the length of the sequence proportional to the length of the shapelets. While implementing such virtual sequences, we still consider the alignment of the sequences to get optimal results.

We look at two methods for the extraction of shapelets. First, as mentioned in [19] we cluster the time series [22] from the same class to select candidate shapelets. These candidate shapelets are then compared to the samples by employing euclidean distance. A sliding window technique is then used to select the shapelets. We also use the method mentioned in [6], where we learn the shapelets. The algorithm exploits logistic regression model. It looks at binary variable for learning process. The loss function learns the weights and shapelet values simultaneously to obtain convergence of the model. Shapelets and the Time series for the dataset "CBF" can be visualised in Figure 3.4 and 3.3 respectively.

### 3.5 Learning Virtual Sequences

Predefined virtual sequences are not established for a given task. We propose a novel time series classification method based on Discriminative Virtual Sequences Learning (DVSL). Let  $\mathbf{X} = \{X^n, y^n\}_{n=1}^N$  denote a set of  $N$  time series samples. We assume that the time series belong to  $C$  different classes. In particular, a time series sample can be represented

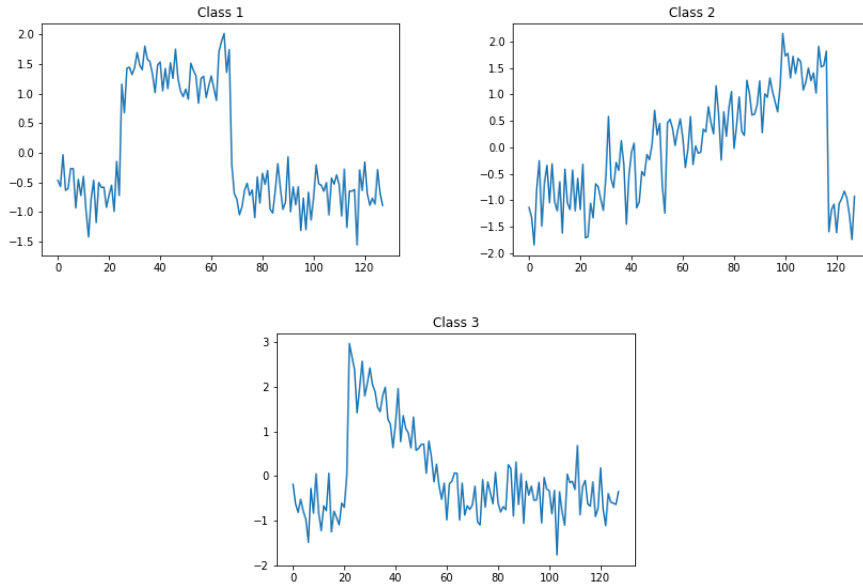


Figure 3.3: Time Series from the UCR time series archive. Cylinder-Bell-Funnel is a simulated data set. X-axis represent time stamps and Y-axis represent values of the time series

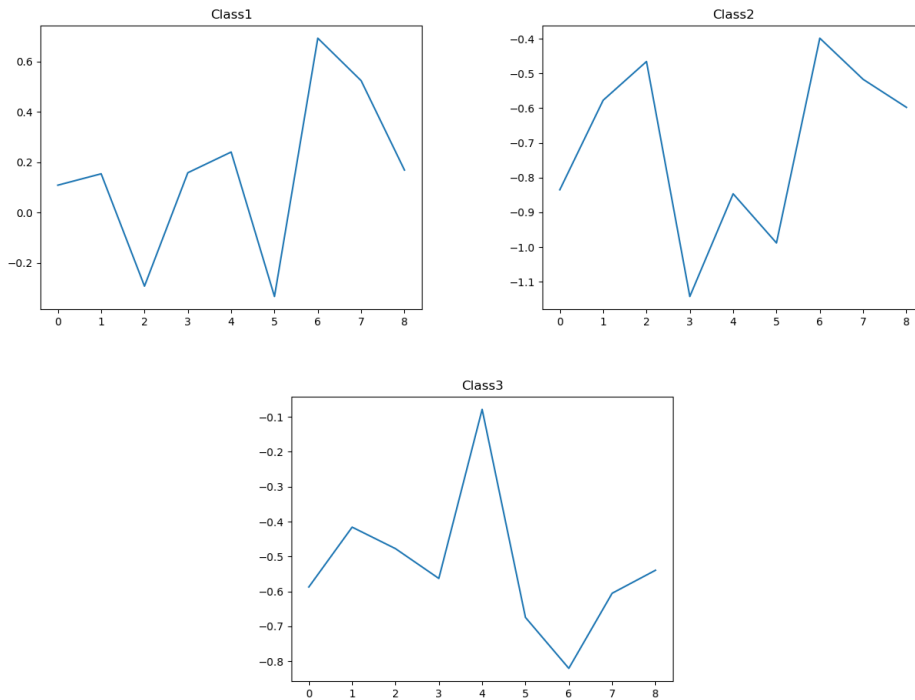


Figure 3.4: Shapelets extracted from CBF dataset. X-axis represent time stamps and Y-axis represent values of the time series

as  $X^n = [x_1^n, \dots, x_{L_n}^n]$ , where  $L_n$  is the length of  $X^n$ . A regression virtual sequence metric learning (RVSML) framework is proposed in [18], which jointly optimizes a ground metric  $D(W)$  parameterized by  $W$  and a temporal alignment matrix  $T$ . The objective function of RVSML is formulated as:

$$\min_{W, T} \Phi(X, D(W), T, V^*) + \Omega(T), \quad (3.2)$$

where  $\Phi(\cdot)$  denotes metric learning,  $\Omega(\cdot)$  denotes the regularization term on  $T$ , and  $V^*$  is a set of predefined virtual sequences.

The major limitation of RVSML is that the virtual sequences are fixed during metric learning, which motivates us to explore adaptive and discriminative virtual sequences for time series classification. By considering virtual sequences  $V$  as variables, we formulate our method as:

$$\min_{W, T, V} \Phi(X, D(W), T, V) + \Theta(V) + \Omega(T), \quad (3.3)$$

where  $W$  is the parameter matrix and  $\Theta(V)$  is a regularization term.

### Distance Metric Learning

The first term of Eq. 3.3 aims to learn a ground metric  $D(W)$  that optimizes the distance between the virtual sequence and the training time series samples, given the alignment matrix  $T$ . By using the virtual sequences, DVSL brings closer time series samples from the same class to a specific virtual sequence, and meanwhile push away samples from different classes. Specifically,  $\Phi(X, W, T, V)$  is formulated as:

$$\begin{aligned} \Phi(X, D(W), T, V) &= \frac{1}{N} \sum_{n=1}^N \langle T^n, D_T^n(W) \rangle + \lambda \|W\|_F^2 \\ &= \frac{1}{N} \sum_{n=1}^N \sum_{i=1}^{L_n} \sum_{j=1}^{l_n} t_{ij}^n \|W^T x_i^n - v_j^n\|_2^2 + \lambda \|W\|_F^2 \end{aligned} \quad (3.4)$$

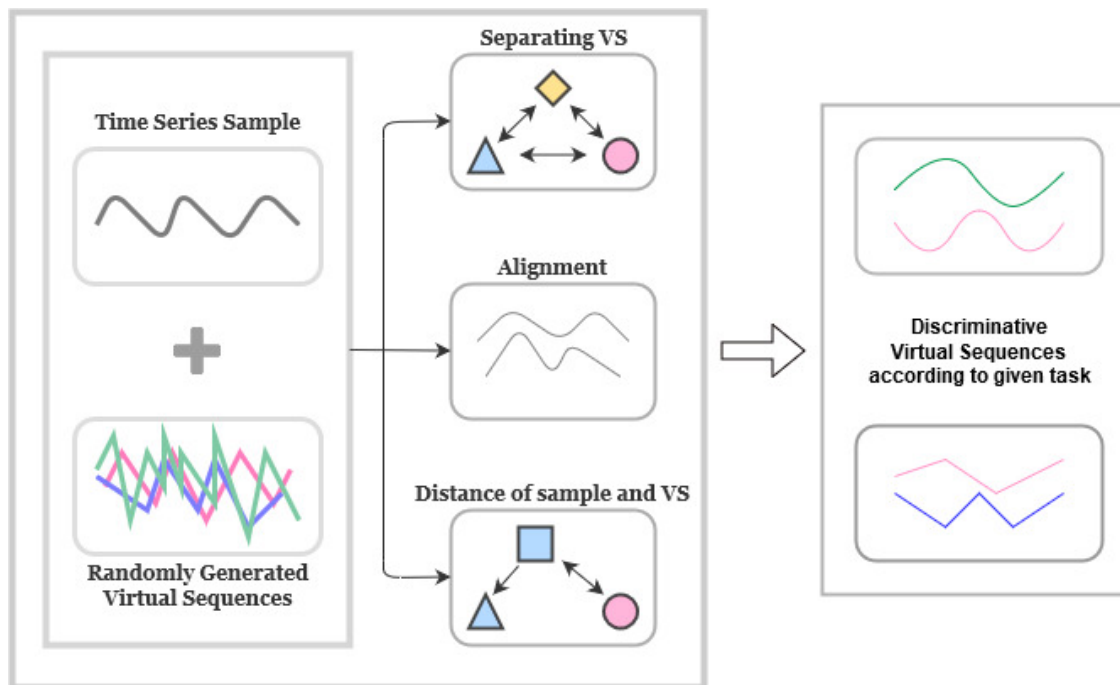


Figure 3.5: Framework of the proposed DVSL method. The input contains time series samples and randomly generated virtual sequences (VS). During training, our method jointly learns discriminative VS that are well separated, aligns the training samples with the VS, and optimizes the ground metric for classification purpose.

where  $\lambda$  is a trade-off parameter, and  $l_n$  is the length of virtual sequence.  $t_{ij}^n$  is an element in the alignment matrix  $T$ , which captures how closely the training sample and the virtual sequence align.

## Virtual Sequence Learning

As discussed above, the virtual sequences used in RVSML [18] are fixed, which may not be optimal for the subsequent time series classification task. Instead, our DVSL method aims to learn a set of discriminative virtual sequences that can directly benefit the classification task. In particular, the discriminative virtual sequences shall be well separated in the data space. To this end, we design the second term in Eq. 3.3 as follows:

$$\Theta(V) = - \sum_{n=1}^N \sum_{c=1, c \neq n}^N \sum_{j=1}^{l^n} \sum_{i=1}^{l^c} \left\| V_j^n - V_i^c \right\|_2^2 \quad (3.5)$$

where  $V_j$  and  $V_i$  denote virtual sequences. With this term, we push the virtual sequences far away from each other by maximizing their pairwise distances in the sequential data space. To align with the minimization problem in Eq. 3.3, a negative sign is added to Eq. 3.5.

## Objective Function

Combining the distance metric learning, virtual sequence learning and the regularization term, the overall objective function of our DVSL method can be viewed in Eq. 3.6. The framework of the model is displayed in Figure 3.5.

$$\begin{aligned} \min_{W, T, V} \mathcal{L} = & \sum_{n=1}^N \sum_{i=1}^{L^n} \sum_{j=1}^{l^n} \frac{1}{N} t_{ij}^n \left\| W^T x_i^n - v_j^n \right\|_2^2 + \lambda \left\| W \right\|_F^2 \\ & - \sum_{n=1}^N \sum_{c=1, c \neq n}^N \sum_{j=1}^{l^n} \sum_{i=1}^{l^c} \left\| V_j^n - V_i^c \right\|_2^2 + \Omega(T). \end{aligned} \quad (3.6)$$

## 3.6 Optimization

Although the objective function in Eq. 3.6 is not jointly convex with respect to all the variables  $W$ ,  $V$  and  $T$ , it is convex to each variable separately when the others are fixed. Thus, we alternatively update these variables. In particular, for the subproblem w.r.t.  $T$ , we implement it with DTW that is solved by dynamic programming. For  $W$  and  $V$ , we employ a gradient descent approach by initializing the variables  $W^{(0)}$  and  $V^{(0)}$  with random values and then updating them with the following rules:

$$W^{(t+1)} = W^{(t)} - \gamma \frac{\partial \mathcal{L}}{\partial W} \quad (3.7)$$

and

$$V_j^{t+1} = V_j^{(t)} - \gamma \frac{\partial \mathcal{L}}{\partial V_j} \quad (3.8)$$

where  $\gamma$  is a learning rate. The detailed derivatives with respect to  $W$  and  $V$  are mentioned in Eq. 3.9 and 3.10 respectively.

$$\frac{\partial \mathcal{L}}{\partial W} = A^{-1} \sum_{n=1}^N \sum_{i=1}^{L_n} \sum_{j=1}^{l_n} t_{ij}^n x_i^n v_j^{nT}, \quad (3.9)$$

where  $A = \sum_{n=1}^N \sum_{i=1}^{L_n} \sum_{j=1}^{l_n} t_{ij}^n x_i^n x_i^{nT} + \lambda NI$ .

$$\frac{\partial \mathcal{L}}{\partial V_j} = \left( \sum_{n=1}^N \sum_{i=1}^{L_n} t_{ij}^n W^T x_i^n - \sum_{n=1}^N \sum_{c=1, c \neq n}^N \sum_{i=1}^{l_n} V_i^c N \right) / C, \quad (3.10)$$

where  $C = \sum_{n=1}^N \sum_{i=1}^{L_n} t_{ij} - N \sum_{n=1}^N \sum_{c=1, c \neq n}^N \sum_{i=1}^{l_n} 1$ .

After optimizing the virtual sequences  $V$  and the ground metric  $W$ , we employ the one

---

**Algorithm 1:** Solving the Optimization problem

---

**Result:** Discriminative Virtual sequence and transformation  $W$   
Initialize virtual sequence and the matrix randomly;  
**Input:** Time Series samples  $X_1, X_2, \dots, X_n$ , parameters  $\lambda, \gamma, m$ ;  
**while**  $W$  has not converged **do**  
    **for**  $i:=1,2,\dots,n$  **do**  
        Update  $T$  by 3.4, keeping  $W$  and  $V$  constant;  
        Update  $W$  by 3.4, keeping  $V$  and  $T$  constant;  
        Update  $V$  by 3.4, keeping  $W$  and  $T$  constant;  
    **end**  
**end**

---

nearest neighbor (1-NN) classifier with DTW distance measure for time series classification. The steps for the same can be represented in Algorithm 1. The updating procedures are repeated until convergence or a maximum number of iterations.

The algorithm produces discriminative sub sequences for a particular task. Such a metric learning approach produces exceptional results and reduces the constraints at the same time. We have tested the model on various data sets, showing good convergence of the model.

# Chapter 4

## Experiments

In this section, we discuss the details of our experiments and display the results. We also describe the various datasets used and parameter settings.

### 4.1 Datasets

We evaluate our method thoroughly on the UCR time series archive [5]. The repository contains datasets from various real word domains. Every data set has a different source from which it is recorded and has its own individual characteristics.

The repository contains vast spectrum of datasets from various sources like sensor data, image data and spectrograph data. For instance, the data set *Beef* consists of data points from beef spectrographs. In general, food spectrographs are used to classify food types. The data set consists of four classes on which classification is performed. Similarly, the dataset Chinatown, reports pedestrian activities in various cities. This helps in future planning of the city. These are sensor readings that are converted to time series data. A summary of few datasets from the UCR archive [5] can be in found in Table 4.1. The table shows the variability in the datasets.

Dataset	Length of time series	Number of classes	Type of Dataset
Arrowhead	251	3	Image
Beef	470	5	Spectrograph
Car	577	4	Sensor
ChlConcent	166	3	Sensor
Coffee	286	2	Spectrograph
DiaSizeRed	345	4	Image
FaceFour	350	4	Image
FacesUCR	131	14	Image
GunPoint	150	2	Motion
Wine	234	2	Spectrograph

Table 4.1: Dataset description

Hyperparameter	Range
Regularization parameter ( $\lambda$ )	$10^{-4}$ - $10^{-1}$
Learning rate ( $\gamma$ )	$10^{-4}$ - $10^{-2}$
Temporal structures ( $m$ )	2 - 8

Table 4.2: Hyperparameter range

## 4.2 Experimental Settings

There are three hyper parameters in our method: the regularization parameter  $\lambda$ , the learning rate  $\gamma$  and the number of temporal structures  $m$ . According to definition in [18], virtual sequences are a combination of 0's and 1's. They are generated using identity and zero matrices. The length of which is determined by the parameter  $m$

To tune the hyper parameters, we use a validation set and do a grid search. The range of values for the hyperparameters is tabulated in Table 4.2. We also report the train/test split used in the implementation of experiments. These splits are standard splits in the UCR time series archive [5].

We perform our experiments on a system with Intel i5-8250 processor with 8 gigabytes memory. The experiments were implemented on MATLAB version R2020a running on Linux

Dataset	Size of Training set	Size of Test set
Arrowhead	36	175
Beef	30	30
Car	60	60
ChlConcent	467	3840
Coffee	28	28
DiaSizeRed	16	306
FaceFour	24	88
FacesUCR	200	2050
GunPoint	50	150
Wine	57	54

Table 4.3: Train/Test split for the dataset

Ubuntu 18.04.4 LTS.

We evaluate the performance of proposed DVSL method instantiated by DTW distance. We learn the ground matrix and the virtual sequences, after which a 1-NN classifier in combination with DTW distance measure is applied. We report accuracy as performance measure. The influence of hyperparameters is mentioned in further sections.

In the next section, we describe our results.

### 4.3 Results and Analysis

We perform experiments on several datasets from the UCR archive and demonstrate the effectiveness of our method. Table 4.4 summarizes the results of our DVSL method and baseline. The results demonstrate that our method outperforms baseline in most of the cases and produces comparable results in other cases. Similar to RVSML, our method can be extended to learn nonlinear representations with deep neural networks, which will be explored in our future work.

Results for virtual sequences generated by shapelets are summarized in Table 4.5. We

Table 4.4: Results for DVSL method

Data set	RVSML	DVSL
ArrowHead	<b>74.86</b>	72
Beef	83.33	<b>90</b>
Car	80	<b>83.50</b>
CBF	<b>85.33</b>	76.16
ChinaTown	88.69	<b>90.46</b>
ChlorineConcentration	60.833	<b>65.92</b>
Coffee	<b>100</b>	<b>100</b>
Computers	<b>56.4</b>	53.08
DiatomSizeReduction	95.09	<b>96.08</b>
ECG200	<b>85</b>	83.5
ECG5000	92.04	<b>92.53</b>
ECGFiveDays	95	<b>97.36</b>
FaceAll	<b>85.44</b>	83.50
FaceFour	<b>87.5</b>	86.25
FacesUCR	78.03	74.30
GunPoint	86	<b>88</b>
Ham	<b>81.9</b>	63.9
Haptics	41.88	<b>42.11</b>
Herring	65.60	<b>65.63</b>
InsectWingbeatSound	<b>58.83</b>	58.19
ItalyPowerDemand	<b>97.18</b>	95.71
Meat	90.00	<b>98.83</b>
MoteStrain	86.90	<b>87.37</b>
OliveOil	73.30	<b>84.67</b>
Plane	<b>100</b>	98.76
TwoLeadECG	<b>91.83</b>	91.6
Wine	59.25	<b>65</b>

also varied the length of the shapelets, results of which are tabulated in Table 4.6.

Compared with RVSML, our method has more variables for optimization. However, the computational efficiency of our method is comparable to that of RVSML, owing to the efficient calculations of gradients. For instance, on the SonyAIBORobotSurface1 dataset, the training time (in seconds) of RVSML and our method are 0.55s and 0.70s, respectively.

Dataset	RVSML(%)	Shapelet Accuracy(%)
ArrowHead	<b>74.86</b>	74.80
Beef	<b>83.33</b>	<b>83.33</b>
CBF	<b>85.33</b>	84.80
GunPoint	<b>86.00</b>	<b>86.00</b>
TwoLeadECG	<b>91.83</b>	79.10

Table 4.5: Results for Shapelet as Virtual Sequence

Dataset	length=9	length=19	length=29
ArrowHead	<b>74.80</b>	74.29	73.00
Beef	<b>83.33</b>	<b>83.33</b>	<b>83.33</b>
CBF	84.11	<b>84.80</b>	<b>84.80</b>
GunPoint	80.67	85.33	<b>86.00</b>
TwoLeadECG	79.00	<b>79.10</b>	75.00

Table 4.6: Results for varied Shapelet Length

In addition, we analyze the convergence property of our model as well as the parameter sensitivity. Figure 4.1 shows that our method converges quickly within 50 iterations. Figure 4.2 displays the parameter sensitivity to  $m$ . Our method obtains relatively stable results when  $m$  varies from 2 to 8.

**Baselines.** We also compare DVSL with the following baselines.

1. **Euclidean Distance (ED):** It’s a standard metric for time series comparison.
2. **Dynamic Time Warping (DTW):** As mentioned in [2], DTW follows a dynamic programming approach for classification.
3. **Regressive Virtual Sequence Metric Learning (RVSML)** [18]: This method learns distances for sequences by finding a ground metric.
4. **Locally Slope-based Dynamic Time Warping for Time Series Classification (LSDTW)** [21]. It implements a weighted DTW technique by looking at regional information and pairing locally similar shapes.

Dataset	ED	DTW	LSDTW	RVSML	DVSL
ArrowHead	<b>80.00</b>	70.29	73.14	74.86	72.00
Beef	66.67	63.33	83.33	83.33	<b>90.00</b>
Car	73.33	73.33	<b>86.67</b>	80.00	83.50
ChlConcent	65.00	64.84	72.11	60.83	<b>77.43</b>
Coffee	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>
ECG200	<b>88.00</b>	77.00	85.00	85.00	83.50
ECGFiveDays	79.67	76.77	90.01	95.00	<b>97.36</b>
Herring	51.56	53.13	46.88	65.60	<b>65.63</b>
InsectWingb	56.16	35.51	50.91	<b>58.83</b>	58.19
Meat	93.33	93.33	80.00	90.00	<b>98.83</b>
MPhaOLAge	51.95	50.00	56.49	48.70	<b>58.18</b>
OliveOil	<b>86.67</b>	83.33	83.33	73.30	84.67
SonyAIBR	69.55	72.55	78.20	<b>83.86</b>	76.16
TwoLeadECG	74.71	90.52	<b>93.15</b>	91.83	91.60
Wine	61.11	57.41	57.41	59.25	<b>65.00</b>

Table 4.7: Times series classification results of the proposed DVSL method and baselines on UCR datasets. Dataset names have been made short.

We tabulate the results in Table 4.7 and demonstrate the effectiveness of our method. We outperform the baselines in most of the cases or achieve comparable results otherwise.

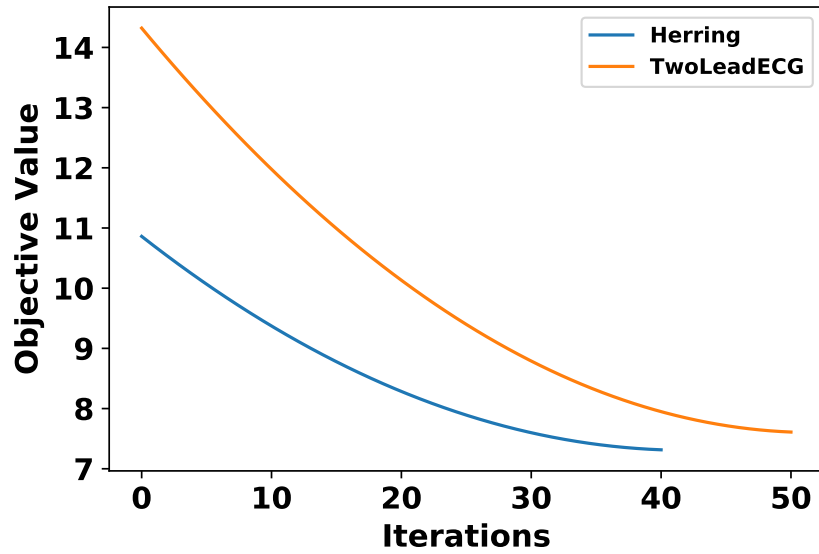


Figure 4.1: Convergence Curve

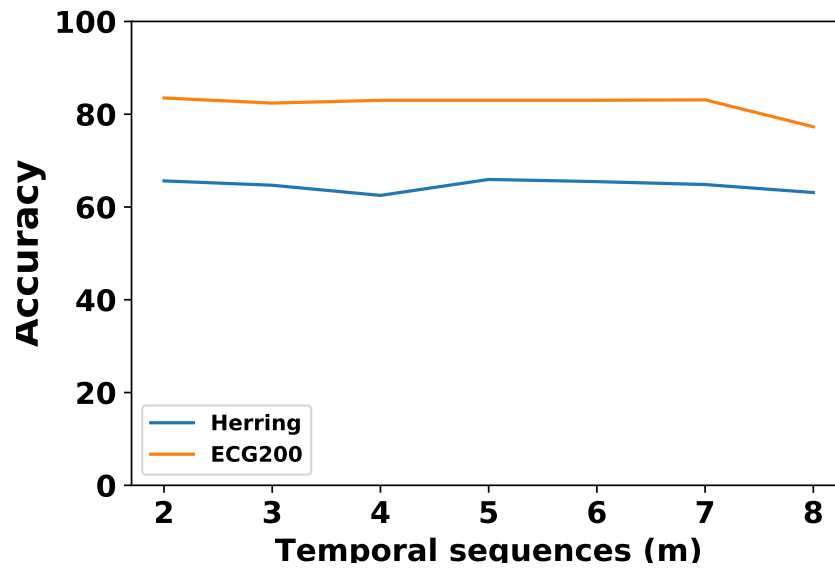


Figure 4.2: Parameter Sensitivity

# Chapter 5

## Conclusions

Time series classification has been proven instrumental in numerous domains and is considered an important task. This thesis proposes two methods for the generation of virtual sequences, one with shapelets and another with learning the virtual sequences. Shapelets produces results equivalent to the existing methods. We learn time series shapelets and substitute them along with a combination of 0's to generate a virtual point. In the second method we randomly initialise the virtual sequences and learn them simultaneously with parameter matrix.

Virtual sequences generated using shapelets contain the values of shapelets in the virtual sequences. Since the goal of virtual sequence is to better separate out the samples from each other, values of shapelets are employed. The number of columns in the sequence is proportion to the length of the shapelets and the number of rows is equal to the number of shapelets learnt for that length.

In the second method we learn the virtual sequences based on discriminative virtual sequence learning. We learn the parameter matrix and the virtual sequences simultaneously. The end product consists of virtual sequences learnt for that specific task. The goal of DVSL is to bring closer time series samples from the same class to a specific virtual sequence

and meanwhile push away samples from other classes. Different from existing work, DVSL adaptively learns a set of discriminative virtual sequences. A new objective function is formulated and a gradient descent algorithm is designed for optimization. Experiment on several UCR time series data set demonstrates the effectiveness of our method.

In the future, we plan to extend the idea of triplet loss to our method. We realise that the current loss function can be considered as a special case of the triplet loss model. We also intend to employ certain deep learning models to learn a latent space that can better perform the classification.

# References

- [1] Amaia Abanda, Usue Mori, and Jose A Lozano. A review on distance based time series classification. *Data Mining and Knowledge Discovery*, 33(2):378–412, 2019.
- [2] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, USA:, 1994.
- [3] Huanhuan Chen, Fengzhen Tang, Peter Tino, Anthony G Cohn, and Xin Yao. Model metric co-learning for time series classification. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.
- [4] Eli Cortez, Anand Bonde, Alexandre Muzio, Mark Russinovich, Marcus Fontoura, and Ricardo Bianchini. Resource central: Understanding and predicting workloads for improved resource management in large cloud platforms. In *Proceedings of the 26th Symposium on Operating Systems Principles*, pages 153–167, 2017.
- [5] Hoang Anh Dau, Eamonn Keogh, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, Yanping, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, Gustavo Batista, and Hexagon-ML. The ucr time series classification archive, October 2018. [https://www.cs.ucr.edu/~eamonn/time\\_series\\_data\\_2018/](https://www.cs.ucr.edu/~eamonn/time_series_data_2018/).

- [6] Josif Grabocka, Nicolas Schilling, Martin Wistuba, and Lars Schmidt-Thieme. Learning time-series shapelets. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 392–401, 2014.
- [7] Steinn Gudmundsson, Thomas Philip Runarsson, and Sven Sigurdsson. Support vector machines and dynamic time warping for time series. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 2772–2776. IEEE, 2008.
- [8] Jing Guo, Zihao Chang, Sa Wang, Haiyang Ding, Yihui Feng, Liang Mao, and Yungang Bao. Who limits the resource efficiency of my datacenter: an analysis of alibaba datacenter traces. In *Proceedings of the International Symposium on Quality of Service*, pages 1–10, 2019.
- [9] Shuichi Hashida and Keiichi Tamura. Multi-channel mhlh: Lstm-fcn using macd-histogram with multi-channel input for time series classification. In *2019 IEEE 11th International Workshop on Computational Intelligence and Applications (IWCIA)*, pages 67–72. IEEE, 2019.
- [10] Qing He, Fuzhen Zhuang, Tianfeng Shang, Zhongzhi Shi, et al. Fast time series classification based on infrequent shapelets. In *2012 11th international conference on machine learning and applications*, volume 1, pages 215–219. IEEE, 2012.
- [11] Chao Huang, Xian Wu, Xuchao Zhang, Suwen Lin, and Nitesh V Chawla. Deep prototypical networks for imbalanced time series classification under data scarcity. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2141–2144, 2019.
- [12] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Shun Chen. Lstm fully convolutional networks for time series classification. *IEEE access*, 6:1662–1669, 2017.

- [13] Eamonn J Keogh and Michael J Pazzani. Scaling up dynamic time warping for datamining applications. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 285–289, 2000.
- [14] Fangyu Li, Yang Shi, Aditya Shinde, Jin Ye, and Wenzhan Song. Enhanced cyber-physical security in internet of things through energy auditing. *IEEE Internet of Things Journal*, 6(3):5224–5231, 2019.
- [15] Mohammad Norouzi, David J Fleet, and Russ R Salakhutdinov. Hamming distance metric learning. In *Advances in neural information processing systems*, pages 1061–1069, 2012.
- [16] Michaël Perrot and Amaury Habrard. Regressive virtual metric learning. In *Advances in neural information processing systems*, pages 1810–1818, 2015.
- [17] Charles Reiss, Alexey Tumanov, Gregory R Ganger, Randy H Katz, and Michael A Kozuch. Heterogeneity and dynamicity of clouds at scale: Google trace analysis. In *Proceedings of the Third ACM Symposium on Cloud Computing*, pages 1–13, 2012.
- [18] Bing Su and Ying Wu. Learning distance for sequences by learning a ground metric. In *International Conference on Machine Learning*, pages 6015–6025, 2019.
- [19] Liuyi Yao, Yaliang Li, Yezheng Li, Hengtong Zhang, Mengdi Huai, Jing Gao, and Aidong Zhang. Dtec: Distance transformation based early time series classification. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pages 486–494. SIAM, 2019.
- [20] Lexiang Ye and Eamonn Keogh. Time series shapelets: a new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 947–956, 2009.

- [21] Jidong Yuan, Qianhong Lin, Wei Zhang, and Zhihai Wang. Locally slope-based dynamic time warping for time series classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1713–1722, 2019.
- [22] Jesin Zakaria, Abdullah Mueen, and Eamonn Keogh. Clustering time series using unsupervised-shapelets. In *2012 IEEE 12th International Conference on Data Mining*, pages 785–794. IEEE, 2012.