TIME SERIES ON RIEMANNIAN MANIFOLDS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

HAMZA ERGEZER

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

DECEMBER 2017

Approval of the thesis:

TIME SERIES ON RIEMANNIAN MANIFOLDS

submitted by **HAMZA ERGEZER** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Tolga Çiloğlu
Head of Department, **Electrical and Electronics Engineering** ________________

Prof. Dr. Kemal Leblebicioğlu
Supervisor, **Electrical and Electronics Eng. Dept., METU** ________________

Examining Committee Members:

Prof. Dr. A. Aydın Alatan
Electrical and Electronics Eng. Dept., METU ________________

Prof. Dr. Kemal Leblebicioğlu
Electrical and Electronics Eng. Dept., METU ________________

Prof. Dr. A. Enis Çetin
Electrical and Electronics Eng. Dept., Bilkent University ________________

Prof. Dr. Gözde Bozdağı Akar
Electrical and Electronics Eng. Dept., METU ________________

Assist. Prof. Dr. Aykut Erdem
Computer Engineering Department, Hacettepe University ________________

Date: ________________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: HAMZA ERGEZER

Signature :

ABSTRACT

TIME SERIES ON RIEMANNIAN MANIFOLDS

Ergezer, Hamza
Ph.D., Department of Electrical and Electronics Engineering
Supervisor : Prof. Dr. Kemal Leblebicioğlu

December 2017, 104 pages

In this thesis, feature covariance matrices are utilized to solve several problems related to time series. In the first part of the thesis, a novel representation is proposed to represent the time series using feature covariance matrices. By this representation, time series are carried onto Riemannian manifold space. The proposed representation is firstly applied to trajectories which are essentially 2D time series. Anomaly detection and activity perception problems in crowded visual scenes are studied by using the trajectories. The second utilization of the proposed representation is for classification of 1D time series. The feature covariance matrices of overlapping subsequences are extracted and fed into two well-known classifiers as the input. The last contribution of the thesis is a rank-based distance measure for high dimensional covariance matrices. The distance measure is utilized to solve skeletal action recognition problem. Unlike classical distance measures, the rank-based distance measure enables us to learn the manifold structure. For this reason, essentially, it can be asserted that the proposed approach is about manifold learning. Performances of the approaches proposed in this thesis have been compared to most of the state-of-the-art techniques on publicly available well-known datasets. For all of the studied problems, we achieve comparable or outperforming results compared to the state-of-the-art techniques.

Keywords: time series representation, feature covariance matrices, Riemannian manifolds, time series classification, trajectory clustering, anomaly detection, rank-based distance measure

ÖZ

RİEMANN MANİFOLDLARI ÜZERİNDE ZAMAN SERİLERİ

Ergezer, Hamza
Doktora, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Kemal Leblebicioğlu

Aralık 2017 , 104 sayfa

Bu tezde, zaman serileri ile ilgili problemleri çözmek için öznitelik kovaryans matrisleri kullanılmıştır. Tezin ilk bölümünde, zaman serilerinin öznitelik kovaryans matrisleri kullanılarak temsil edilmesi için yeni bir gösterim önerilmektedir. Bu gösterimle zaman serileri Riemann manifoldu uzayına taşınır. Önerilen gösterim, öncelikle iki boyutlu zaman serileri olan yörüngelere uygulanmıştır. Kalabalık görsel sahnelerde anomali tespiti ve aktivite algılama problemleri yörüngeler kullanılarak incelenmiştir. Önerilen gösterimin ikinci kullanımı bir boyutlu zaman serilerinin sınıflandırılmasında gerçekleşmiştir. Örtüşen alt dizilerin öznitelik kovaryans matrisleri çıkarılmış ve girdi olarak iki iyi bilinen sınıflandırıcıya beslenmiştir. Tezin son katkısı, yüksek boyutlu kovaryans matrisleri için sıralamaya dayalı bir mesafe ölçüsüdür. Mesafe ölçüsü, iskelet hareketi tanıma problemini çözmek için kullanılmıştır. Klasik mesafe ölçülerinden farklı olarak, sıralamaya dayalı mesafe ölçüsü, manifoldun yapısını öğrenmeyi sağlar. Bu nedenle, esas olarak, bir manifold öğrenme yaklaşımının da önerildiği iddia edilebilir. Bu tezde önerilen yaklaşımların performansları, yaygın olarak bilinen veri kümeleri üzerinde en yeni tekniklerle karşılaştırılmıştır. Çalışılan tüm problemler için, en yeni tekniklerle karşılaştırıldığında benzer veya daha iyi sonuçlar elde edilmiştir.

Anahtar Kelimeler: zaman serilerinin gösterimi, öznitelik kovaryans matrisleri, Riemann manifoldları, zaman serilerinin sınıflandırması, yörünge kümeleme, aykırılık tespiti, sıraya dayalı uzaklık ölçüsü

To my family

ACKNOWLEDGMENTS

I would like to thank the supervisor Professor Kemal Leblebicioğlu for his constant support, guidance, and patience. It was a great honor to work with him.

Members of my thesis committee, Professor Aydın Alatan and Professor Enis Çetin, always gave valuable feedbacks for the progress of this work. I am very thankful to them for their advice.

Firstly and deeply, I want to thank my parents and my beloved little brother for their emotional support. My elder brother, Halit Ergezer, deserves special thanks due to showing me that what I can accomplish during my whole life.

And there are a lot of people who were with me during the Ph.D. study. It is not possible to write down why each of them is important for me and for this work. Here, I can only express my gratitude to Erdem Akagündüz, Kubilay Pakin, Gökçen Aslan and Mustafa Ayazoğlu for their wonderful comments and suggestions to our works.

Last but not least, very special thanks to members of my little family. Neriman is always more than a wife for me. During the study, her support was very encouraging. And my daughter, Deniz, gave me the luck to complete this work.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGMENTS x

TABLE OF CONTENTS xi

LIST OF TABLES xv

LIST OF FIGURES xvii

LIST OF ALGORITHMS xx

CHAPTERS

- 1 INTRODUCTION 1
  - 1.1 Motivation 1
  - 1.2 Contributions 4
  - 1.3 Outline of the Thesis 6
- 2 LITERATURE REVIEW 7
  - 2.1 Literature Review on Time Series 7
    - 2.1.1 Time Series Representations 8
      - 2.1.1.1 Nondata Adaptive Methods 8

2.1.1.2 Data Adaptive Methods 9

2.1.1.3 Model-based Methods 9

2.1.2 Distance Measures on Time Series 10

2.1.2.1 Dynamic Time Warping (DTW) . . . 10

2.1.2.2 Modified Versions of DTW 11

2.1.2.3 Longest Common Subsequence (LCSS) 12

2.1.2.4 Edit Distance with Real Penalty 12

2.1.2.5 Time Warp Edit Distance (TWED) . . 13

2.2 Covariance Features 13

2.2.1 Riemannian Manifolds 14

2.2.2 Distance Calculation on Riemannian Manifolds . . 16

3 ACTIVITY PERCEPTION AND ANOMALY DETECTION WITH FEATURE COVARIANCE MATRICES 19

3.1 Introduction 19

3.2 Related Work 21

3.3 Trajectory Representation by Feature Covariance Matrices . . 22

3.4 Anomaly Detection on Trajectories 25

3.5 Activity Perception via Trajectories 28

3.6 Experiments 29

3.6.1 Results on Synthetic Dataset 29

3.6.2 Results on Real Datasets 31

3.7 Conclusion 34

4 TIME SERIES CLASSIFICATION WITH FEATURE COVARIANCE MATRICES 39

4.1 Introduction 39

4.2 Related Work 42

4.3 Representation of Time Series by Feature Covariance Matrices 44

4.4 Time Series Classification with Feature Covariance Matrices 49

4.4.1 Classification with 1-NN classifier 50

4.4.2 Classification with SVM classifier 51

4.5 Experiments 52

4.5.1 Comparative Results 56

4.5.2 Analysis on Pointwise Features 64

4.5.3 Analysis on Computational Complexity 70

4.6 Conclusion 71

5 SKELETAL ACTION RECOGNITION WITH RANK DISTANCE . 75

5.1 Introduction 75

5.2 Related Work 76

5.3 Representation of Skeletal Data with Feature Covariance Matrices 77

5.4 Action Recognition with Rank Distance Measure 78

5.5 Experiments 80

5.5.1 Comparison with Other Distance Measures 81

5.5.2 Action Recognition Results 83

5.5.3 Analysis on Number of Permutations 83

5.5.4 Analysis on Computational Complexity 85

5.6 Conclusion 87

6 CONCLUSION AND FUTURE WORK 89

6.1 Conclusion 89

6.2 Future Work 91

REFERENCES 93

CURRICULUM VITAE 103

LIST OF TABLES

TABLES

Table 2.1 Comparison of the distance measures against some challenges. . . . 13

Table 2.2 Summary of the distance measures for SPD matrices. 16

Table 3.1 Accuracies of anomaly detection methods for the synthetic dataset built in [26]. The proposed representation outperforms the state-of-the-art techniques with use of anomaly measures, nearest neighbors (NN) and sparse representation (SR). Sparse representation also gives better results compared to single use of nearest neighbor. 30

Table 4.1 Information about the datasets used in experiments 54

Table 4.2 Misclassification rates of the methods for a subset of the datasets in UCR repository. 57

Table 4.3 Error rates for remaining motion datasets. 64

Table 4.4 Analysis on global pointwise features. Average accuracies for seven combinations of pointwise features are listed. Abbreviations stand for LF: three Local Features together, CS: Cumulative Sum, DM: Difference between Mean. 65

Table 4.5 Analysis on local pointwise features. Average accuracies for seven combinations of pointwise features are listed. GF stands for global features. 65

Table 5.1 The comparison of the rank based distance measure with well-known distance measures on a synthetic data. 81

Table 5.2 Classification accuracies of different distance measures using the kNN classifier on MSR-Action3D and MSR-DailyActivity3D datasets. . . 82

Table 5.3 Comparison of the action recognition results with the techniques based on covariance matrix and other state-of-the-art approaches. 83

Table 5.4 Classification accuracies with respect to the number of permutations for the MSR-Action3D and MSR-DailyActivity3D datasets using kNN classifier. 85

Table 5.5 Computation times of distances between training and test sets for different distance measures on MSR-Action3D dataset. 87

LIST OF FIGURES

FIGURES

Figure 1.1 Some time series examples from the datasets used in the thesis. . . 2

Figure 2.1 Two time series to show the challenges for distance measures for time series. 11

Figure 2.2 Dynamic time warping for time series given in Figure 2.1. 12

Figure 2.3 Fundemantal concepts of Riemannian geometry: tangent space, exponential map, geodesic distance and logarithm map. 15

Figure 3.1 Representation of trajectories using feature covariance matrices. . 23

Figure 3.2 A scenario to explain the necessity of sparse anomaly detection algorithm. 26

Figure 3.3 Representative images from the datasets used in the experiments. The synthetic dataset, two scenes from UCSD dataset, Grand Central and MIT parking lot datasets are given in upper left, upper right, bottom right and bottom left, respectively. 30

Figure 3.4 A sample result for anomaly detection in the synthetic dataset. Ten samples are shown for each cluster of normal trajectories. Anomalous trajectories are indicated with bold magenta lines. 31

Figure 3.5 Clustering result for a set of trajectories in the synthetic dataset. . . 32

Figure 3.6 Similarity matrix of trajectories given in Figure 3.5. Five clusters can be observed together with the anomalies which lie in the last rows and columns. 32

Figure 3.7 Ten most anomalous trajectories in Train Station dataset. 33

Figure 3.8 Some examples of anomaly detection results in UCSD dataset. Results lied in the rows are from two different scenes in the dataset. 35

Figure 3.9 Some anomalous trajectories from MIT Parking Lot dataset. These anomalous trajectories might be result of problems in extraction stage. . . 36

Figure 3.10 Trajectory patterns in MIT Parking Lot dataset. 37

Figure 4.1 Representation of time series with feature covariance matrices is depicted in *yoga* dataset of UCR repository. First, a feature vector is defined for each point in the time series. Feature vectors defined for each point of time series form feature matrices for overlapping subsequences. For each subsequence, covariance matrices are calculated and the distance between subsequences in training and test sets are calculated using log-Euclidean distance metric. The distance between the time series is determined by averaging the distances between the subsequences. 47

Figure 4.2 Block diagram of the overall approach for time series classification with 1-NN classifier. 50

Figure 4.3 Time series classifcation with SVM classifer. The model for SVM is the concatentation of the vectorized feature covariance matrices. 53

Figure 4.4 Five samples from UCR time series dataset. Diversity in length and variability for the datasets can be observed. 54

Figure 4.5 Comparison of CovNN with other methods on 43 datasets based on the results presented in Table 4.2. CovNN is better than DTW, ST, TSBF and HOG1D+DTW-MDS for 34, 33, 34 and 22 of the 43 datasets, respectively. 60

Figure 4.6 Comparison of CovSVM with other methods on 43 datasets based on the results presented in Table 4.2. CovSVM is better than DTW, ST, TSBF and HOG1D+DTW-MDS for 35, 34, 35 and 25 of the 43 datasets, respectively. 61

Figure 4.7 Critical difference diagrams for the proposed approach with two classifiers and compared approaches. 62

Figure 4.8 Comparison of NN and SVM classifiers on 43 datasets based on the results presented in Table 4.2. 63

Figure 4.9 Accuracy values for different combinations of global pointwise features for 1-NN classifier. LF stands for local features, namely *value*, *derivative* and *time index*. The datasets are sorted according to accuracy values obtained in the case of six pointwise features. 66

Figure 4.10 Accuracy values for different combinations of global pointwise features for SVM classifier. LF stands for local features, namely *value*, *derivative* and *time index*. The datasets are sorted according to accuracy values obtained in the case of six pointwise features. 67

Figure 4.11 Accuracy values for different combinations of local pointwise features for NN classifier. GF stands for global features. The datasets are sorted according to accuracy values obtained in the case of six pointwise features. 68

Figure 4.12 Accuracy values for different combinations of local pointwise features for SVM classifier. GF stands for global features. The datasets are sorted according to accuracy values obtained in the case of six pointwise features. 69

Figure 4.13 Comparison of the computational times of CovNN, CovSVM and HOG1D+DTW-MDS. 72

Figure 5.1 An example of WTA hashing with 6-dimensional input vectors, K = 4, and θ = (1, 4, 2, 5, 0, 3). 78

Figure 5.2 General flow diagram of the skeletal action recognition approach. . 79

Figure 5.3 Analysis on how classification accuracies change with respect to the dimension of covariance matrices. 82

Figure 5.4 Confusion matrix for MSR-Action3D dataset. 84

Figure 5.5 Confusion matrix for MSR-DailyActivity3D dataset. 84

Figure 5.6 Analysis of the effect of number of permutations on the classification accuracy for MSR-Action3D dataset. 86

Figure 5.7 Analysis of the effect of number of permutations on the classification accuracy for MSR-DailyActivity3D dataset. 86

LIST OF ALGORITHMS

ALGORITHMS

Algorithm 1 WTA hashing of covariance matrices 79

CHAPTER 1

INTRODUCTION

We live in the big data era. Ninety percent of the data in the world today has been created in last two years alone [1]. Every year, together with the size of the data, the variation in the type of the data also increases. Another important fact is that the sharing of the data becomes easier, too. We share our photos and videos within seconds on our mobile phones with lots of other information. Albeit the time series is just a type of these huge data, the all of the technological developments in creating, storing and disseminating the digital data affects the time series in the same way.

1.1 Motivation

Time series is a sequence of measurements. By even this short definition, one can comprehend how crucial is the need for time series analysis and how broad the application areas. People analyze the stock prices, customer usages, sensor readings and many other time series data everyday. Also, data coming from other types of sources such as videos, implicitly contain the time series data. When we think of the amount of data available in the digital era as mentioned above, the automatic analysis of the time series will be a more prominent problem. The examples of the time series focused on this study are shown in Figure 1.1. In the figure at the first row, a 1D time series is shown. We utilize such time series while studying time series classification in Chapter 4. The figure at the middle contains the trajectories of the three objects. We utilize trajectories in Chapter 3 to detect anomalies and to perceive dominant activities in similar scenarios. The figure at the bottom shows a motion

---

[1] http://www.iflscience.com/technology/how-much-data-does-the-world-generate-every-minute/

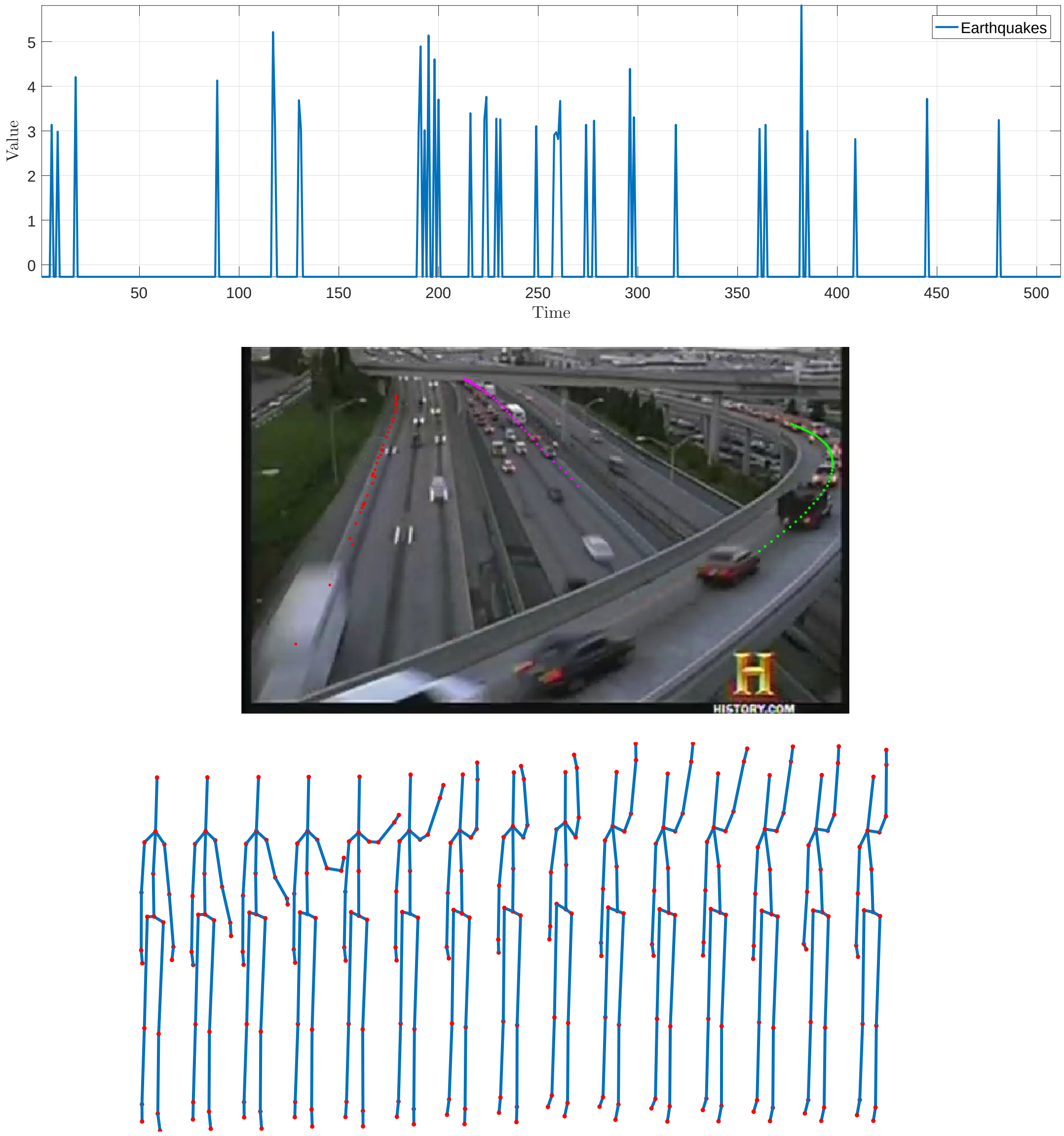


Figure 1.1: Some time series examples from the datasets used in the thesis.

sequence of a skeleton. For each skeleton joint which is shown with the red dots, a multidimensional time series is extracted. An action is defined as the combination of these multidimensional time series. These are only the some types of time series we work on this study. There are many other types of time series and many other problems that can be covered by automatic analysis of time series.

By time series analysis, people encapsulate some problems such as classification, forecasting, clustering, anomaly detection. For all of these problems, there are three major issues to be solved according to [35]: data representation, similarity measurement and an appropriate indexing method. In [15], Bengio et al. begin the paper with the argument that the success of machine learning algorithms generally depends on data representation. This is obviously a valid argument for time series. Although their paper focuses on the learning of the representations, the main message we should get from the paper is that we need good representations for the aforementioned tasks. Hence, a good representation is the most critical issue while dealing with the problems related to time series.

Feature covariance matrices or covariance features are utilized in this study to achieve a good representation for time series. Covariance features carry the characteristics of a good representation for time series as will be discussed in Chapter 2 in detail. As a quick summary, it carries the time series data into a lower dimension. It can capture the local and global shape characteristics. For computationally efficiency, it is shown that the covariance feature has great success compared to its competitor techniques. It also handles the unwanted cases such as noisy and missing data. By exploiting these powerful characteristics, we study some problems in trajectories, 1D time series data, and skeleton data.

The second issue is similarity measurement while dealing with the problems related to time series. After representing the time series data with feature covariance matrices, the data is carried onto Riemannian manifold space. In Riemannian manifold space, the Euclidean distance measure does not provide satisfactory results due to the nonlinearity of the space. On the other hand, there are well-known distance measures and divergence functions that utilize the Riemannian property of the manifolds. While dealing with trajectories and 1D time series, we utilize one of them. However, for skeletal action recognition problem, we have observed that these well-known distance measures may fail due to the high dimensionality of the covariance matrices. For this purpose, a hashing mechanism is exploited to measure the distances between the covariance matrices derived from skeletons.

At this point, the progress of thesis is given. The starting point of the thesis is trajectories which are basically 2D time series. When we investigate the activity perception and anomaly detection problems in crowd scenes, we figured out that detection and tracking of the objects separately are almost impossible. However, trajectories for some points of the objects or cumulative optical flows of the pixels can be still extracted. After these observations, trajectories are utilized as the feature of the objects. Trajectories are valuable features due to carrying both spatial and temporal information of the objects. At this point, feature covariance matrices are utilized to represent the trajectories. The main motivation behind the selection of the feature covariance matrices for trajectories as the feature is that they enable us to calculate the difference between the trajectories of different lengths. By using the feature covariance matrices, all of the trajectories are expressed as equal-sized covariance matrices. We try to solve the problems of anomaly detection and activity perception in crowded scenes through the use of trajectories. In 1D time series case, we apply the proposed representation for time series classification problem.

After applying the representation in 1D and 2D time series through the pointwise features, we investigate the problems which include multidimensional time series. At this point, we have observed that feature covariance matrices have already been applied to the skeletal action recognition problem. For this problem, the whole sequence of joint motions is represented with the feature covariance matrices. This makes the covariance matrices high-dimensional due to the number of joints and measurements at 3D coordinates. Therefore, the classical distance measures cannot model this high-dimensional manifold space. We propose a rank-based distance measure which is based on a hashing mechanism.

1.2 Contributions

The first contribution of the thesis is a novel representation for time series. The representation is based on feature covariance matrices. The representation is also novel in the sense that it comprises novel pointwise features. The most of the pointwise features in the investigated problems have been firstly utilized. Besides, it can be claimed that it offers a novel distance measuring mechanism for time series by putting a classical distance measure following the representation. The representation inherently addresses the several issues such as missing data, time series of different lengths. We achieve state-of-the-art results by using the representation for all problems in 1D and 2D time series.

Activity perception and anomaly detection problems are investigated using the proposed representation on trajectories. After representing the trajectories with feature covariance matrices, the distance between two trajectories is determined with the log-Euclidean distance measure. In anomaly detection part, a novel data-driven approach is proposed. The anomaly measure is a sparse combination of the weighted distances to the nearest neighbors. For activity perception, activities are treated as dominant motion patterns and therefore, clustering of trajectories is realized for this purpose. Spectral clustering is applied after a basic transformation of the distance matrix into the similarity matrix.

The exploitation of the feature covariance matrices for time series classification problem on 1D time series is another contribution of the study. For 1D time series, the proposed representation used in trajectories is modified. The modification of the representation is two-fold. The first modification is that the feature covariance matrices are determined for the subsequences. Secondly, the new local and global pointwise features are introduced. Our main purpose, as in 2D time series, is to evaluate the feature covariance representation for time series classification problem. Therefore, in classification stage, firstly, the 1-NN (nearest neighbor) classifier is used. SVM (support vector machine) classifier is also utilized for the problem. Experiments are conducted to test the performance of the method using UCR time series repository. UCR repository includes many types of time series such as motion, electrodiagram (ECG), image (or shape), sensor readings and synthetic data. The proposed method mostly outperforms the state-of-the-art methods on UCR repository.

The last contribution is a distance measure for high-dimensional symmetric positive definite (SPD) matrices. The distance measure is based on rank or ordinal metrics. Instead of projecting the points on Riemannian manifold space to other spaces, we directly measure the distances with the entries of covariance matrices by using a hashing mechanism. We apply the distance measure to the skeletal based action recognition problem which can be considered as a multidimensional time series classification problem. Due to the hashing approach, the rank-based distance measure is very fast compared to the classical distance measure used for SPD matrices. Our results on several datasets have proven the strength of the proposed distance measure.

Lastly, a list of the papers published during the study is given below:

- H. Ergezer, K. Leblebicioglu, “Time Series Classification Using Feature Covariance Matrices”, Knowledge and Information Systems, accepted, 2017.
- H. Ergezer, K. Leblebicioglu, “Anomaly Detection and Activity Perception Using Covariance Descriptor for Trajectories”, Workshop on Crowd Understanding, ECCV 2016.
- H. Ergezer, K. Leblebicioglu, “Time Series Classification Using Point-wise Features”, SIU 2017.
- H. Ergezer, K. Leblebicioglu, “Anomaly Detection in Trajectories”, SIU 2016.

1.3 Outline of the Thesis

A brief literature review about time series representation and the distance measures for time series are given in Chapter 2. Introductory information about Riemannian manifolds and distance calculation for SPD matrices are also given in this chapter. Starting from Chapter 3, we provide the solutions of the problems in trajectories, 1D time series and skeleton data, respectively. Chapter 3 introduces the application of the proposed representation on trajectories and solutions to the problems of activity perception and anomaly detection. The studies on time series classification for 1D time series are presented in Chapter 4. In chapter 5, a novel distance measure and its exploitation for skeletal action recognition problem are summarized. The thesis is summarized with some conclusions and possible future works in Chapter 6.

CHAPTER 2

LITERATURE REVIEW

The main goal of the chapter is to provide the background material for subsequent chapters. For this purpose, we give a literature review for representations and distance measures for time series. For other problems studied in each chapter, literature review of the problems are included in the relevant chapters.

The outline of the chapter is as follows. First, a brief literature review on representations and distance measures for time series is given in Section 2.1. In Section 2.2, the background material for feature covariance matrices is given by describing Riemannian manifolds and distance measures used for symmetric positive definite (SPD) matrices.

2.1 Literature Review on Time Series

Time series might appear in different problems from various disciplines. Therefore, in last two decades, the researchers propose several solutions for the problems [89] such as classification, clustering, anomaly detection and etc.

While dealing with the aforementioned problems for time series, there are two main approaches. These approaches basically differs from each other by how they measure the distances between two time series. The first approach is the measuring the distance directly on time series. The second approach is to represent the time series by bringing into the discriminative parts implicitly. Therefore, we give a background material for representations and distance measures for time series.

2.1.1 Time Series Representations

Time series are high-dimensional data for most of the applications. Working directly with the raw time series can be computationally expensive and memory inefficient in some cases. For such cases, we should need a representation or a descriptor for the data. An ideal representation should have the following characteristics:

- dimensionality reduction,
- capturing local and global shape characteristics,
- computationally inexpensive,
- implicit noise handling,
- low reconstruction error.

We follow the categorization given in [59]. In this taxonomy, the representation techniques for time series can be broadly divided into three categories: nondata adaptive, data adaptive and model-based.

2.1.1.1 Nondata Adaptive Methods

Nondata adaptive techniques do not include any data-specific parameters. It applies the same transformation to all time series. Nondata adaptive techniques can be broadly categorized into two categories: spectral-domain techniques and time-domain techniques. Spectral-domain techniques generally come from signal processing community. Discrete Fourier transform (DFT), discrete wavelet transform (DWT) and discrete cosine transform (DCT) are the most well-known examples of spectral decompositions or transformations. DFT transform the time series using the complex exponential functions as basis functions. DWT uses scaled and shifted versions of mother wavelet functions as basis functions. DCT utilizes only cosine function as a basis function while decomposing the time series.

Time-domain techniques do not transform the data any other domain. The Piecewise Aggregate Approximation (PAA) proposed in [56] represents the time series with the mean values of consecutive fixed-length segments. Lin et al. have proposed a multiscale extension of PAA in [57]. There are also some other works [6, 30] suggesting to extract the representations using amplitude levels of the time series.

2.1.1.2 Data Adaptive Methods

Data adaptive techniques transform the data using the parameters depending on the data. Every nondata adaptive representation can be modified to a data adaptive one by adding a data-dependent selection step. In general, the methods in this category are the modifications of the nondata adaptive methods. Similar to DFT or DCT, singular value decomposition (SVD) transform the data using basis functions. However, the basis functions in SVD case are eigenvectors of the data. Another analogy can be established between wavelets and shapelets. Similar to wavelets, shapelets are a small part of the time series. However, shapelets are extracted using the data and the main goal is to extract more representative parts of the data.

Another type of data adaptive techniques built a symbolic representation of the data. The most well-known symbolic representation is Symbolic Aggregate approXimation (SAX) [69]. SAX follows the basic idea of PAA by quantizing the amplitudes of time series. However, in SAX approach, the quantization levels are data-dependent and the quantization levels are optimized by considering the most frequent amplitude levels and alphabet size. The last approach we should mention is the piecewise approximation of the time series. In these approaches, the time series is represented by piecewise linear or polynomial approximations. The piecewise functions are determined according to the subsequence of the time series.

2.1.1.3 Model-based Methods

The model-based approaches assume that the time series are generated by an underlying model. The main aim is to find the parameters of the model. Therefore, the similarity between the time series is built on the similarity of the parameters. Moving average (MA), autoregressive moving average (ARMA), hidden Markov models (HMM) and Markov chains (MC) are exemplars of this type of approaches.

2.1.2 Distance Measures on Time Series

In this section, we provide an introductory overview of the distance measures for time series. These measures are applied directly to time series without utilizing any representation block. The first one is the shift in the time axis. This can be occurred due to the measurements from the same occasion can be occurred in different time instants. The second one is the noise on the measurements. A distance measure should not be susceptible to some noisy inputs and should inherently eliminate these distortions. Another challenge for a distance measure is the time series of different lengths. This problem is so usual for real-world applications. Uniform amplitude shift or bias could be another challenge for some applications. This problem is generally eliminated by zero-mean normalization. The last challenge is outliers. There could be some instances that do not fit the sequence of the data. Again, this problem can be handled by preprocessing the data. However, filtering the data can cause missing the discriminative information. A scenario that depicts the challenges mentioned above is given in Figure 2.1. The time series are from Adiac dataset of UCR repository [23] and belong to same class. The one given above, x_1, is more noisy, short in time and have a constant amplitude bias compared to time series given below. On the other hand, x_2 has two outliers.

The simplest distance measure is the Euclidean distance for given two time series. However, the Euclidean distance and L_p norms cannot handle the challenges mentioned above. In order to handle these challenges, some elastic measures are proposed. We present these measures in the following subsections. First, DTW and its modified versions are presented. Then, we mention about LCSS, TWED and ERP which are essentially distance measures proposed for measuring the distances between strings.

2.1.2.1 Dynamic Time Warping (DTW)

Dynamic Time Warping (DTW) can be considered as the most well-known approach for calculating the distance between time series. It is mainly proposed to mitigate the distortions in the time axis. It determines the pairwise points between two sequences

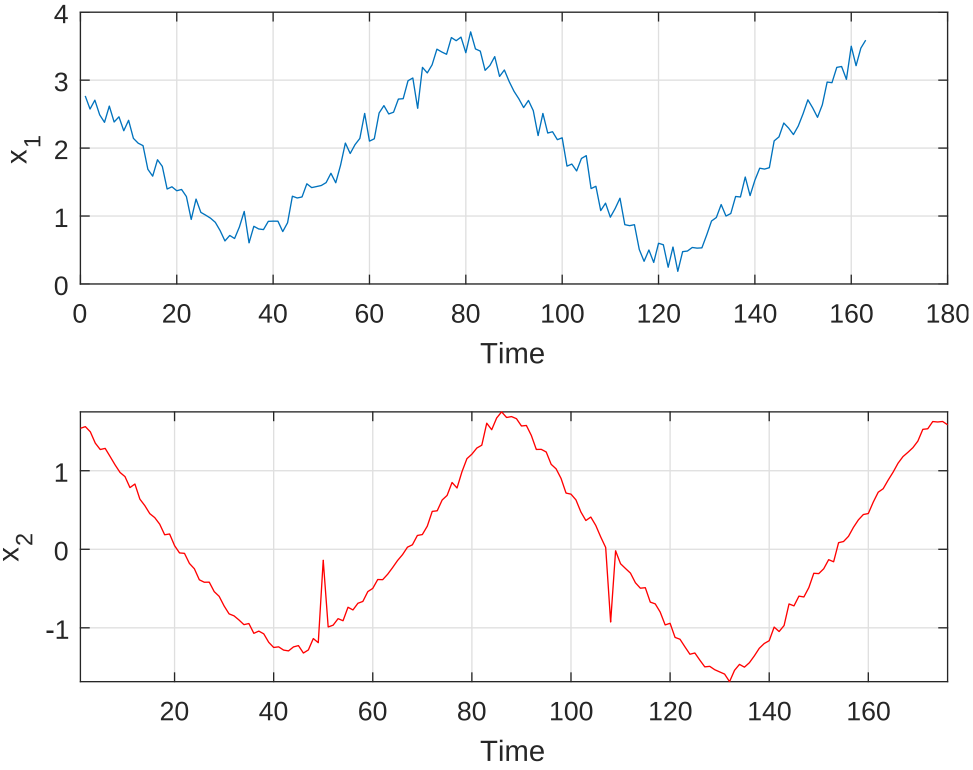


Figure 2.1: Two time series to show the challenges for distance measures for time series.

by dynamic programming. Since it is based on dynamic programming, it can be computationally expensive. Therefore, it should be a constraint on warping window to reduce computation time [88]. Lastly, we should note that DTW measures a distance-like quantity between two given sequences, however, it does not guarantee the triangle inequality.

2.1.2.2 Modified Versions of DTW

DTW is a successful approach for problems dealing with time series in most of the cases. However, it has modified versions that are proposed to solve some limitations of DTW. Derivative DTW [61] tries to solve singularity problem where a single point may map onto the large subsection of the second sequence. For this purpose, DTW approach is executed on average differences of the original sequences. In more detail, a point a_i is replaced by its average differences with its neighbors, a_{i-1} and a_{i+1}.

The other version of DTW is weighted DTW [52]. In this approach, it gives weights

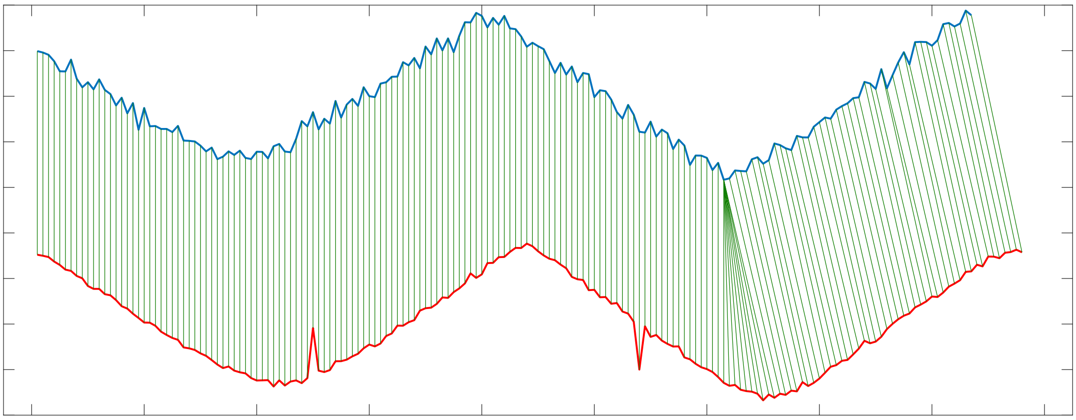


Figure 2.2: Dynamic time warping for time series given in Figure 2.1.

according to the warping distances between the corresponding points. When creating a distance matrix on which dynamic programming execute, a weighting function is applied to the distances. A logistic function is utilized to determine the weights together with a constraining the maximum warping distance.

2.1.2.3 Longest Common Subsequence (LCSS)

Starting from LCSS, the distance measures we will mention hereinafter are based on edit distance concept. The edit distance is introduced for string matching problem. The most basic form of the edit distance is LCSS. A threshold parameter is introduced to adapt the edit distance concept to the time series. The threshold parameter defines the maximum difference between the real-valued time series. The longest matched subsequence defines the similarity between the two time series. Therefore, the distance between two time series a and b of length m is given by Equation 2.1

$$d_{LCSS}(a,b) = 1 - \frac{LCSS(a,b)}{m} \tag{2.1}$$

2.1.2.4 Edit Distance with Real Penalty

One more step ahead from LCSS is edit distance on real sequences (EDR) [21]. It uses thresholding mechanism like LCSS but also penalizes the non-matching pairs. A modification of EDR is edit distance with real penalty (ERP) [22]. Since EDR posits

Table 2.1: Comparison of the distance measures against some challenges.

Distance measure	**Different Length**	**Time Shift**	**Additive Noise**	**Outliers**
L_p norms	✗	✗	✗	✗
DTW	✓	✓	✗	✗
Derivative DTW	✓	✓	✓	✓
Weighted DTW	✓	✓	✓	✗
LCSS	✗	✗	✓	✗
ERP	✗	✗	✓	✗
TWED	✓	✓	✓	✗

that the matching elements are equal, it does not satisfy the triangular inequality. In ERP, to satisfy the triangular inequality and achieve a distance metric, authors use real distances between the matching points. For non-matching elements, like in EDR, ERP takes a constant penalty term.

2.1.2.5 Time Warp Edit Distance (TWED)

As the name indicates, time warp edit distance [76] combines the characteristics of LCSS and DTW. As in LCSS, the distance is determined by the length of matching or common subsequences. Besides, the warping is allowed in time axis as in DTW. The warping is controlled and penalized by a parameter ν. The effect of ν is a multiplicative. Depending on the value of ν, TWED goes from unconstrained DTW to Euclidean (or another Lp-norm) distance. As in other elastic measures in previous subsections, TWED is implemented by using dynamic programming.

2.2 Covariance Features

One of the major contributions of the thesis is a novel representation for time series. The proposed representation is based on feature covariance matrices. By using feature covariance matrices, the time series is represented by a covariance matrix. Since covariance matrices can be considered as symmetric positive definite (SPD) matrices, they lie on Riemannian manifolds. In this section, we give a brief summary of

Riemannian manifolds to provide a background to the reader.

2.2.1 Riemannian Manifolds

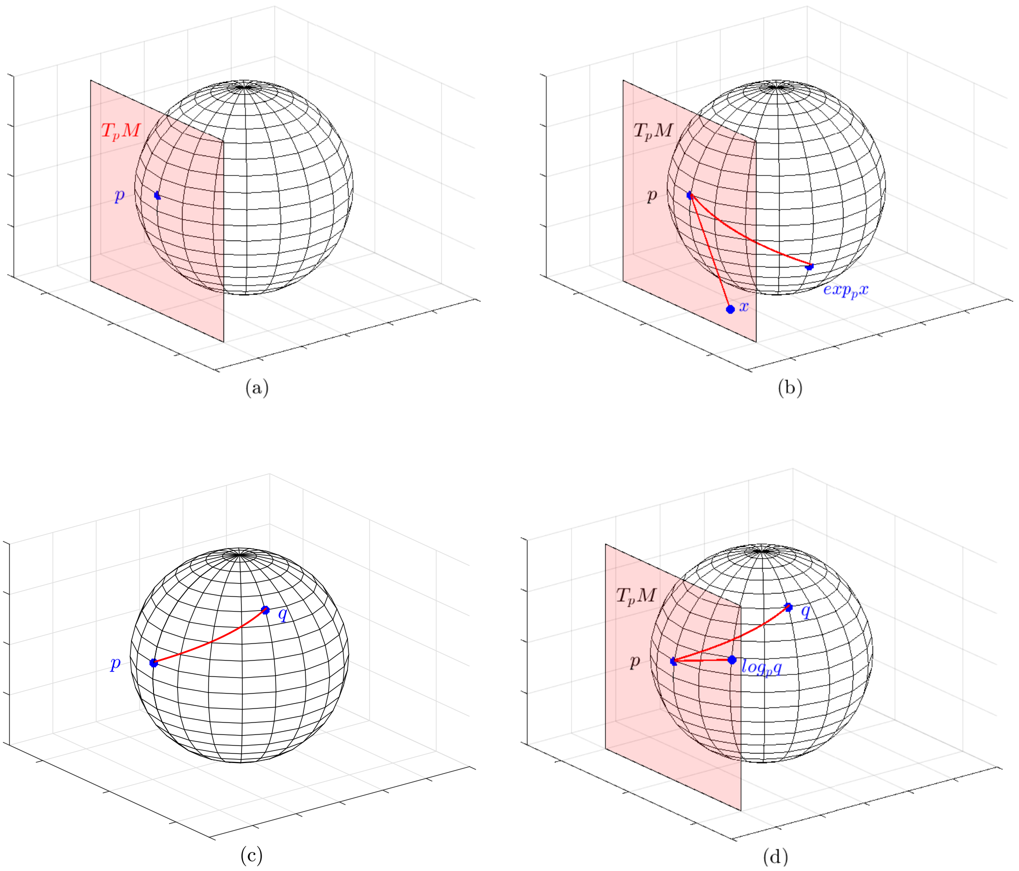
As a well known fact, covariance matrices are positive semidefinite matrices, i.e,

$$a^T C a \geq 0, \forall a \in R^n \tag{2.2}$$

where $C \in R^{nxn}$ is a covariance matrix and a is nonzero vector in R^n. Due to this well-known fact, covariance matrices have some properties and implicit constraints. The first property which is important for our study is that they are nonlinear. Fortunately, due to some other properties that they have, they lie on Riemannian manifolds. The advantage of lying on Riemannian manifolds is that Riemannian manifolds are locally linear and therefore they have some nice properties. For this reason, in this section, we give a brief information about the Riemannian manifolds to better understanding the theoretical background of the proposed representation.

Riemannian manifolds can be located on the third layer of a manifold hierarchy. In the first two layers, we have topological manifolds and smooth manifolds. Here, without diving into details of manifold geometry, our main goal is to give an intuitive background about manifolds. Topological manifolds are the most relaxed version of manifolds. They can be thought as the generalization of surfaces in R^n. The surfaces do not contain any weird parts or holes. In topological manifold layer, there is no difference between a cube and a sphere. In the layer of smooth manifolds, we leave cubes behind us. More formally, the structure of the manifolds in this layer is smooth. There is no essential difference between ellipses and spheres. Because we can not talk about the notions of angles and distances yet. When we move into the Riemannian manifold layer, we can talk about the notions of angles and distances. An ellipse and a sphere are different objects in this layer since the former one has a varying curvature while the latter has constant curvature.

Although we can talk about the distances and angles for Riemannian manifolds, they are still nonlinear. Therefore, the Euclidean distance measures do not work well for Riemannian manifolds. The notions of exponential and logarithmic maps are



(a) (b) (c) (d)

Figure 2.3: Fundemantal concepts of Riemannian geometry: tangent space, exponential map, geodesic distance and logarithm map.

utilized for this purpose. In Figure 2.3, the concepts of tangent space, exponential map, geodesic distance and logarithmic map are shown.

Tangent space and its vectors are used to represent the related points on the manifold. Exponential and logarithm maps are used to transfer the data between tangent spaces and the manifold or vice versa, respectively. In more detail, exponential one maps a tangent vector denoted by x in Figure 2.3 to a point on the manifold (Equation 2.3). On the other hand, logarithm map carries a point onto the tangent space of another point on the manifold (Equation 2.4).

$$\text{Exponential Map:} \quad exp_p : T_pM \rightarrow M \tag{2.3}$$

Table 2.2: Summary of the distance measures for SPD matrices.

Distance measure	**Formula**	**Distance Metric**
AIRM [85]	$\|\|log(C_1^{-1/2}C_2C_1^{-1/2})\|\|_F$	Yes
Log-Euclidean [5]	$\|\|log(C_1) - log(C_2)\|\|_F$	Yes
Stein [92]	$[log\ det(\frac{1}{2}C_1 + \frac{1}{2}C_2) - \frac{1}{2}log\ det(C_1C_2)]^{1/2}$	Yes
Cholesky	$\|\|chol(C_1) - chol(C_2))\|\|$	No
Kullback-Leibler	$tr(C_1^{-1}C_2 - I) - logdet(C_1^{-1}C_2)$	No

$$\text{Logarithm Map:} \quad log_p : M \to T_pM \tag{2.4}$$

By using tangent spaces of two points and the inner product of tangent vectors of these spaces, we can define the distances. The curve that defines this minimum distance is the geodesic and the length of the curve is the geodesic distance. As we will see in the next section, almost all of the distance measures use logarithm operation due to this reason.

2.2.2 Distance Calculation on Riemannian Manifolds

Feature covariance matrices are utilized in several image processing and computer vision problems. It has a proven success for these problems. For most of the problems, the most critical issue is to measure the similarity between the features represented by the feature covariance matrices. As emphasized in Section 2.2.1, a covariance matrix is an SPD matrix and lies on Riemannian manifolds. Since Riemannian manifolds are nonlinear, Euclidean distance or other Lp norms does not give satisfactory results. For these reasons, a proper distance measure for SPD matrices is one of the items that should be considered.

A summary of the classical distance measures for SPD matrices are given in Table 2.2. Affine invariant Riemannian metric or shortly Riemannian metric [85] is the most well-known distance metric. It can be considered as the geodesic distance since it consists of the operations exponential logarithm maps. The disadvantage of the AIRM distance is computationally expensive. To overcome the computation needs of AIRM distance, authors in [5] proposes Log-Euclidean distance measure. The log-

Euclidean distance measure is also used in this study and gives the best results for all problems. In order to achieve a faster distance measure, authors propose a distance metric which is based on Jensen-Bregman log-det divergence [25]. Sra converts the Jensen-Bregman divergence to a distance metric in [92] by taking the square root of it.

The classical distance measures have shown great successes in many studies and they are used as in our study as mentioned in the following chapters. More recently, kernel methods are proposed to measure the similarity for SPD matrices [50, 51].

CHAPTER 3

ACTIVITY PERCEPTION AND ANOMALY DETECTION WITH FEATURE COVARIANCE MATRICES

In this chapter, we study the problems of anomaly detection and activity perception through the trajectories of objects in crowded scenes. For this purpose, we propose a novel representation for trajectories via covariance features. Representing trajectories via feature covariance matrices enables us to calculate the distance between the trajectories of different lengths. After setting this proposed representation and calculation of distances between trajectories, anomaly detection is achieved by sparse representations on nearest neighbors and activity perception is achieved by extracting the dominant motion patterns in the scene through the use of spectral clustering. Conducted experiments show that the proposed method yields results which outperform or are comparable with state of the art.

3.1 Introduction

Improvements in camera technology make the video surveillance systems easily accessible. For this reason, application areas of video surveillance systems are broad. Together with this progress, user expectations have induced new challenges to the field. The biggest challenge is that automated handling of some tasks became mandatory for surveillance systems. Activity perception and anomaly detection are among those important tasks for surveillance systems. Many approaches have been proposed in the literature for anomaly detection and activity perception in scenes. These approaches generally differ from each other with respect to the visual features they utilize. Despite some difficulties in the extraction stage, especially in crowded scenes, the trajectory is still one of the most useful features for an object of interest.

The trajectory is a 2D or 3D time series data depending on the application. It carries position information of the moving object with respect to time. Other valuable information such as velocity can also be derived from trajectory data. Therefore, trajectory data is crucial for several surveillance applications. In maritime surveillance, the trajectory of a vessel is the biggest clue about its behavior. A hijacked plane can be identified from its trajectory in aviation surveillance. For video surveillance, trajectories of the objects in the scene give information about motion patterns. Also, the trajectory of a high-speed car will be different from others and can be identified as an anomaly. As can be seen from the examples, trajectories are valuable features of moving objects to handle tasks such as anomaly detection and activity perception.

In this chapter, the proposed representation or descriptor is utilized for trajectories. A feature vector is defined for each point of the trajectory and a feature matrix is obtained by concatenating these vectors. The proposed descriptor is the covariance of the feature matrix. By representing trajectories via feature covariance matrices, essentially, a novel distance measure is introduced for trajectories. This measure is capable of calculating the distance between the trajectories of different lengths. Since covariance matrices lie on Riemannian manifolds, a distance metric which is capable of measuring geodesic distance is utilized while calculating the distance between the trajectories. Another contribution in this chapter is the achievement of anomaly detection by sparse representations on nearest neighbors. The proposed anomaly detection approach based on sparse representation optimizes the number and weights of the nearest neighbors while setting up an anomaly measure. Activity perception is achieved by spectral clustering of trajectories. Distances determined through the covariance matrices are transformed to similarities to build a similarity graph. Activity perception is then treated to extract the dominant motion patterns in the scene through the use of spectral clustering.

Organization of the chapter is as follows. A brief literature review of the activity perception and anomaly detection in visual scenes is given in Section 3.2. In Section 3.3, the proposed representation for trajectories is introduced. Anomaly detection approach based on the sparse representation of nearest neighbors is described in Section

3.4. Activity perception through clustering of trajectories using spectral clustering algorithm is presented in Section 3.5. Experimental results on both synthetic and real datasets are given in Section 3.6.

3.2 Related Work

Feature covariance matrices are firstly proposed and used as descriptors in [100]. The covariance descriptor basically enables to determine the distance between two instances by representing the instances by their features and their covariance matrix of the feature matrices. After it is proposed in [100] for object detection and classification, covariance descriptor is exploited to solve several computer vision problems such as visual tracking [87], action recognition [40, 16, 102], and saliency detection [33]. In all of these works, covariance descriptor is utilized as region descriptor. Some optical flow components are included in the feature vector, however, in none of them, covariance descriptor is used to describe a 2D time series.

The trajectory is a spatiotemporal feature for a moving object and carries information about its journey in the scene. Hence, it is important to get information about the activities and it is used for activity perception in previous works [12, 105, 80]. While analyzing trajectories, the critical point is the selection of proper distance measure. Several distance measures [12, 60, 45, 17, 26, 7] for trajectories have been proposed so far. Two excellent review papers [79, 113] compare different distance measures for trajectories.

Anomaly detection and activity perception are two important problems for surveillance systems. In recent years, there are many successful works that handle these problems for realistic scenarios. For anomaly detection, in [75], authors use a mixture of temporal and spatial models to detect the anomalies and in [68], they extend the models to multiple scales to detect anomalies at different spatial and temporal scales. A Gaussian Mixture Model (GMM) based probabilistic model is fit to particle trajectories which are extracted by particle advection in [108]. Trajectories that do not fit this model are labeled as anomalies. Aside from computer vision community, there are other works focusing on anomaly detection on trajectories. Laxhammar et.

al. [65] apply their anomaly detector called conformal anomaly detector to the trajectories. In [26], a 1-class Support Vector Machine (SVM) is utilized to detect the anomalous trajectories. The most interesting part of the study is the introduction of a faster solution for SVM training in the presence of outliers. An outlier detection method which is based on the concept of discords is introduced in [58]. Discord for an instance is another instance that has the maximum Euclidean distance to its nearest neighbor.

Nonparametric Bayesian models are widely used for activity perception in recent years. Starting from the pioneering work [106], there are significant works [44, 63] in this path. In [106], nonparametric Bayesian models are adapted to activity perception in visual scenes by modeling the motions in the scene as visual words, short video clips as documents and activities as topics. Follow-up works [44, 63] adapt Markov models to learn the temporal dependencies between activities.

There is a recent approach [94] that considers the trajectories on Riemannian manifolds. The method is based on a representation called transported square-root vector field (TSRVF) and L2 norm on the space of TSRVFs. Authors have also applied their methods to visual speech recognition problem in [95]. In this method, trajectories are mapped into a tangent space by parametrization via its TSVRF. TSVRF formulation includes the derivative and square root of the derivative of the parametrized version of the trajectory. To conclude, the method has a similar idea with our method; however, in our method feature covariance matrices are exploited to map the trajectories to Riemannian manifolds.

3.3 Trajectory Representation by Feature Covariance Matrices

Trajectories can be considered as time series of 2D coordinates. For a visual scene, there might be lots of trajectories of different lengths. In order to analyze these trajectories, first, a similarity or a distance function should be defined. In this study, we propose to describe the trajectories with covariance matrices of their features. By doing so, all trajectories are transferred to space of Riemannian manifolds and similarities between them are calculated in this set.

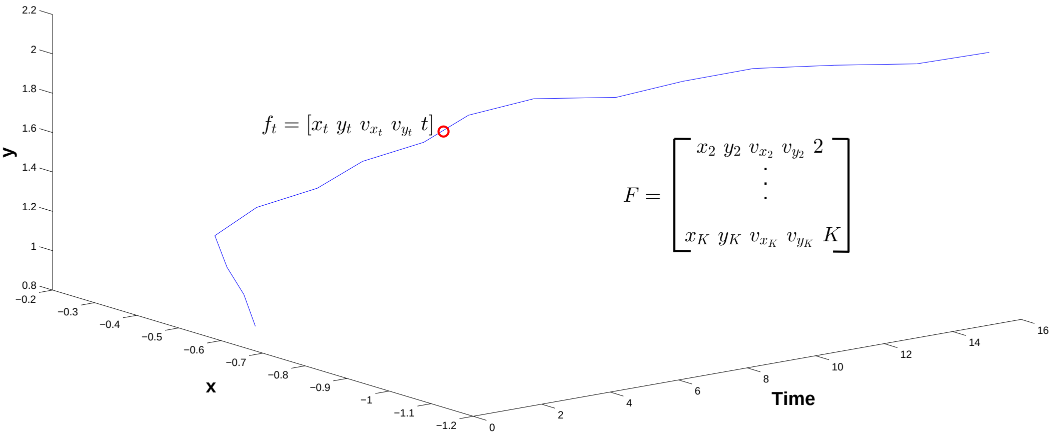


Figure 3.1: Representation of trajectories using feature covariance matrices.

A 2D trajectory can be defined as a sequential concatenation of K points or more formally as a Kx2 matrix, $[x_1\ y_1,...,x_K\ y_K]$, as shown in Figure 3.1. In this figure, a trajectory from synthetic dataset [26] is shown as a sample. The lengths of the trajectories are 16 in the dataset. A feature vector is formed for all points in the trajectory except the first point. For such a case, F matrix will be 15x5 including feature vectors of all trajectories and the resulting covariance matrix will be 5x5.

In order to represent a trajectory with feature covariance matrices, first, a point of a trajectory can also be defined by its features

$$f = [x\ y\ v_x\ v_y\ t] \tag{3.1}$$

where x and y define the position, v_x and v_y are velocities in x and y directions respectively and t is the time index. During experiments, several features including cumulative sum, acceleration etc. have been examined to increase the performance. However, the best performance values are obtained with feature set defined in Equation 3.1. For the whole trajectory, feature matrix can be defined similarly as

$$F = \begin{bmatrix} x_2 \; y_2 \; v_{x_2} \; v_{y_2} \; t_2 \\ . \\ . \\ . \\ x_K \; y_K \; v_{x_K} \; v_{y_K} \; t_K \end{bmatrix} \tag{3.2}$$

Feature covariance matrix is determined as

$$C = \frac{1}{K} \sum_{k=1}^{K} (F_k - \mu)(F_k - \mu)^T \tag{3.3}$$

where μ is the mean vector of all instances in matrix F. At this point, a small multiple of the identity matrix is added to covariance matrices. This regularization is performed to ensure the positive definiteness of the covariance matrix. Positive definiteness is important for the distance metric which involves a logarithm operation.

It should be noted that for all trajectories of different lengths, we end up with a 5x5 covariance matrix. This enables us to determine the similarity between the trajectories of different lengths. After covariance representation, trajectories are carried onto Riemannian manifolds. The critical point from now on is to calculate the distances between the trajectories on Riemannian manifolds.

A distance measure that approximate the geodesic distance between two points on Riemannian manifolds must be used. For this purpose, as previous works [100, 87, 40] that utilize covariance matrices suggested, Euclidean distance metrics must be avoided. We use the log-Euclidean metric which was first proposed in [5] between covariance matrices. Compared to other distance metrics [37] and divergence functions [20], the best performance is achieved by using log-Euclidean metric in this study. Log-Euclidean metric is, in principle, based on matrix logarithms. The determination of the log-Euclidean metric starts with the eigenvalue decomposition of covariance matrices.

$$C = VQV^T \tag{3.4}$$

After this eigenvalue decomposition, matrix logarithm is obtained as

$$log(C) \triangleq V\widetilde{Q}V^T \tag{3.5}$$

where $\widetilde{Q}$ is a diagonal matrix obtained from Q by replacing Q's diagonal entries by their logarithms. The distance between covariance matrices is calculated via Frobenius norm of the distance matrix logarithms.

$$\rho(C_1, C_2) = \|log(C_1) - log(C_2)\|_F \tag{3.6}$$

Now, we can calculate all the distances between trajectories via feature covariance matrices. In subsequent sections, anomaly detection and activity perception problems will be based on these distances. Anomaly detection is carried out by a novel approach based on a sparse representation of nearest neighbors. Activity perception is achieved by forming a similarity matrix from pairwise distances and utilizing this in spectral clustering.

3.4 Anomaly Detection on Trajectories

In this work, trajectories are utilized as the feature of objects in the scene. Therefore, in order to detect the anomalous motions of the objects, anomalous trajectories are determined within the set of all trajectories. An anomalous trajectory can be described as a sample that does not fit motion patterns in the scene. Based on this definition, the nearest neighbor approach can be considered as the simplest solution for anomaly detection. The distance to the nearest neighbor can be a good measure for some cases while deciding anomalies. However, depending on the structure of the data and amount of anomalous observations, distance to the nearest neighbor might not be a good alternative. We propose a method which considers the distances to a set of nearest neighbors and tries to optimize the weights and number of nearest neighbors. A scenario is depicted in Figure 3.2 to explain the necessity of the algorithm. For some anomalies, the nearest neighbor or a weighted sum of nearest neighbors might not be a good anomaly measure. Anomalies are shown inside the red dashed ellipse. For anomalies in the orange circle, the distance to third nearest neighbor should be included in the anomaly measure.

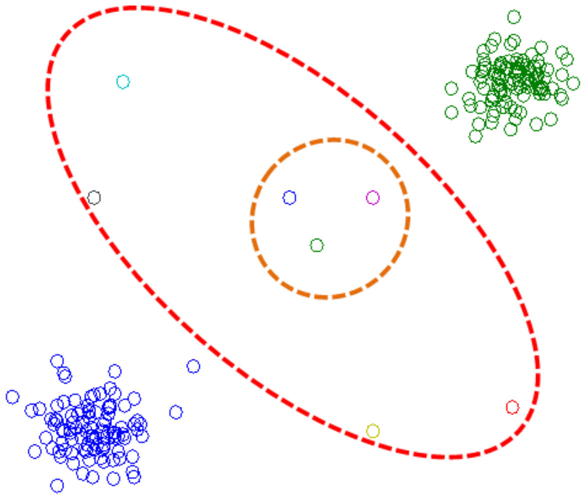


Figure 3.2: A scenario to explain the necessity of sparse anomaly detection algorithm.

After the representation of trajectories via covariance matrices and calculation of distances between trajectories, anomaly detection is carried out by using a measure comprising distances to nearest neighbors. For this purpose, we select nearest neighbors through a sparse representation. In this approach, an anomaly measure is calculated via the weighted sum of distances to nearest neighbors for each sample. Our goal is to optimize the number of the neighbors and their weights while deciding if an instance is anomaly or not.

At this point, the remedy can be considered as dictionary learning on trajectories. Dictionary learning approaches search the sparsest representation of basis vectors that makes the reconstruction error minimum. There are also some previous works [93, 24] that propose approaches for dictionary learning on Riemannian manifolds. However, in these works, the main goal is to retrieve a more accurate form of nearest neighbor. In our case, we try to optimize the weights and number of nearest neighbors for anomaly detection. Therefore, we follow a different strategy for this purpose after the determining the distances using log-Euclidean measure.

In sparse anomaly detection approach, the data is assumed to be offline and available to be divided into uniform parts. In particular, we exploit some part of the data for training and derive optimal weights of nearest neighbors from this subset. A same number of data samples are taken into the testing process. Anomaly measure is composed of distances to K nearest neighbors for each sample in the training set of M

samples.

$$A_i = w_1 d_1 + ... + w_K d_K \tag{3.7}$$

where d_i,..., d_K are distances to K nearest neighbors. Equation 3.7 can be written in a matrix form

$$A_i = \begin{bmatrix} w_1 \ ... \ w_K \end{bmatrix} \begin{bmatrix} d_1 \\ \vdots \\ d_K \end{bmatrix} = WD_i \tag{3.8}$$

and finally when all instances are considered

$$A = WD \tag{3.9}$$

where A is a 1xM vector consisting of anomaly measures for all instances, W is a 1xK vector consisting of weights of K nearest neighbors and D is the matrix that contains distances to the K nearest distances for all samples.

At this point, we use the anomaly labels of the training samples. Our weigths should be such that it supress the non-anomalous samples whereas they boosts anomaly measure for the anomalous samples. Besides, since there might be several combinations of weighted neighbors for each instance, a minimum number of neighbors should be used. Therefore, combining with previous observations, the optimization problem can be summarized as

$$W = \underset{W}{\operatorname{argmin}} \{\lambda |W|_0 + |L_{tr} - W * D|_2\} \tag{3.10}$$

where L_{tr} is the label vector in the training set. Since L0 norm is a nonconvex function, the L1 norm is the first alternative to L0 norm. However, with L1 norm, we still does not guarantee the positive weights in our problem. For this purpose, we put a constraint for non-negative weigths. Then the final optimization becomes a constrained optimization as in Equation 3.11.

$$
\begin{aligned}
&W = \underset{W}{\operatorname{argmin}} \{\lambda |W|_1 + |L_{tr} - W * D|_2\} \\
&\text{subject to } \; w_i \geq 0
\end{aligned}
\tag{3.11}
$$

In our experiments, we show that anomaly detection with sparse representation gives better results than the single use of nearest neighbors or equally weighted of them.

3.5 Activity Perception via Trajectories

Activity perception is the second problem for which the proposed representation is exploited. An activity can be considered as a set of similar trajectories. Clustering is the direct solution for the identification of these sets or activities. Therefore, activity perception is handled with clustering of trajectories.

Describing trajectories through the utilization of feature covariance matrices enables us to construct a similarity matrix between trajectories of different lengths. This similarity matrix can be used to build an undirected graph which allows extracting the motion patterns in the scene. Spectral clustering methods are popular since they are capable of handling non-convex patterns in the data. As in [84], the similarity matrix is built using the distances derived with feature covariance matrices

$$s_{ij} = e^{-d_{ij}^2/2\sigma^2} \tag{3.12}$$

where d_{ij} is the distance between the trajectories i and j. Spectral clustering is achieved by the clustering of eigenvectors of a matrix called Laplacian. In its un-normalized formulation, Laplacian is the difference between the degree matrix and the similarity matrix.

$$L = D - S \tag{3.13}$$

where D is a diagonal matrix which contains the sum of each row of similarity matrix (or column depending on its symmetry). Laplacian matrix is normalized as in [84] to handle the clusters of different sizes.

$$L = I - D^{-1/2} S D^{-1/2} \tag{3.14}$$

where L is the normalized Laplacian and D is degree matrix. Clusters are determined by applying the k-means algorithm on eigenvectors of normalized Laplacian.

3.6 Experiments

During experiments, a synthetic dataset and three real datasets are exploited. Synthetic dataset firstly introduced in [58] is used. The real datasets are UCSD anomaly detection [75, 68], MIT Parking Lot [105] and Train Station (Grand Central) [116]. Anomaly detection on trajectories is carried out on both synthetic and three real datasets to evaluate the performance of the proposed representation. On the other hand, for activity perception, we use only the synthetic dataset and MIT Parking Lot dataset. It is better to mention about two practical details before experimental results. First, the regularization parameter mentioned after Equation 4.3 is selected as 0.005 in all experiments. Secondly, in all real datasets, a size threshold is applied to eliminate small tracks.

3.6.1 Results on Synthetic Dataset

The synthetic dataset generated in [26] is firstly exploited for anomaly detection. The performance of the proposed approach is compared to previous works given in [26, 65, 58]. The synthetic dataset includes 1000 subsets and in each subset, there are 260 trajectories. In each subset, last 10 trajectories are anomalous. Comparative results are given in Table 3.1 for the dataset and a sample result is shown in Figure 3.4. As can be seen in Table 3.1, the proposed representation has outperformed the state-of-the-art techniques just by utilizing the distance to the nearest neighbor only.

The synthetic dataset is also exploited while evaluating the performance of sparse anomaly detection. Sparse anomaly detection is implemented through running of Monte Carlo simulations in the synthetic dataset. In each run, we select 100 sets for training from the whole dataset including 1000 sets. The remainder of the dataset is used for testing. Sparse representation or the weights of the nearest neighbors are applied to the testing set. As shown in Table 3.1, the best results are obtained with the combination of proposed trajectory representation and sparse anomaly detector.

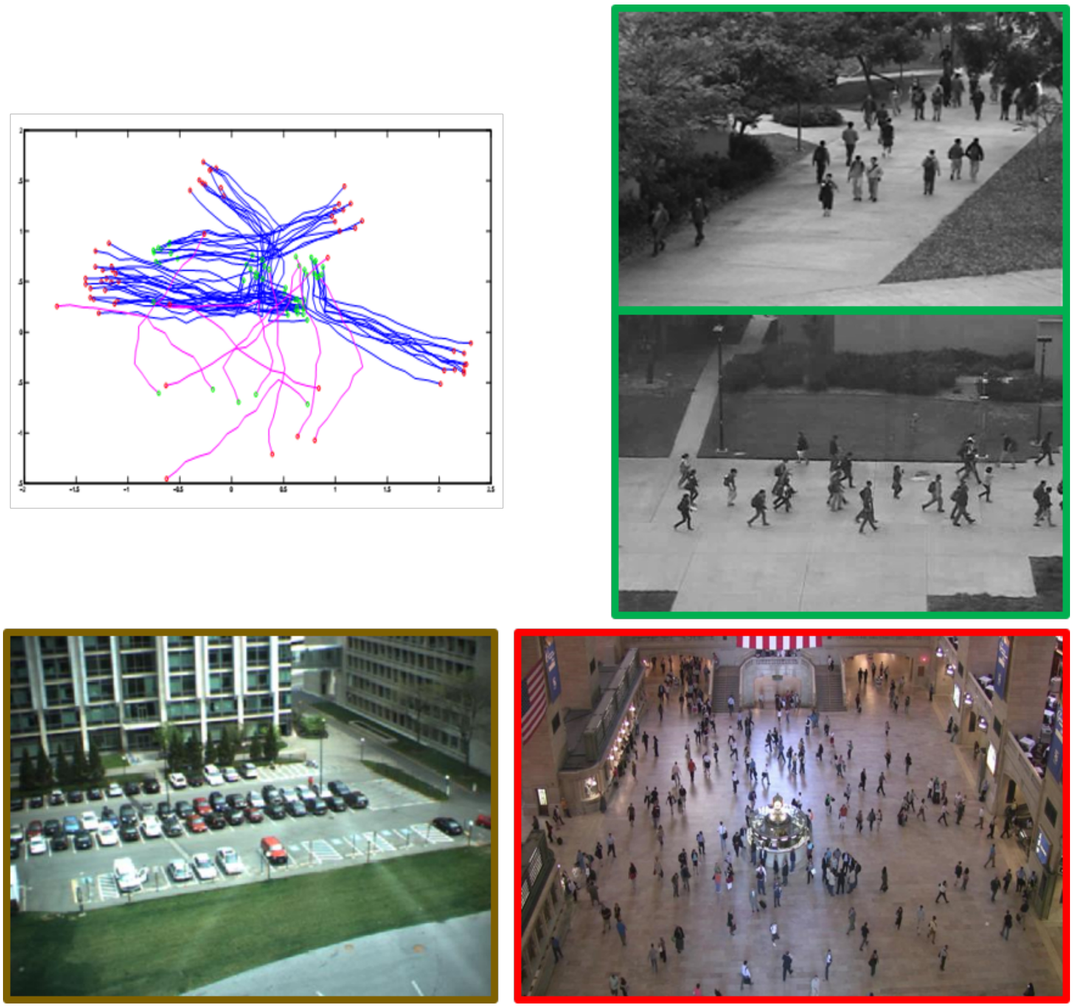


Figure 3.3: Representative images from the datasets used in the experiments. The synthetic dataset, two scenes from UCSD dataset, Grand Central and MIT parking lot datasets are given in upper left, upper right, bottom right and bottom left, respectively.

Table 3.1: Accuracies of anomaly detection methods for the synthetic dataset built in [26]. The proposed representation outperforms the state-of-the-art techniques with use of anomaly measures, nearest neighbors (NN) and sparse representation (SR). Sparse representation also gives better results compared to single use of nearest neighbor.

Method	**Accuracy**
1-Class SVM [26]	0.9630
Conformal Anomaly Detector [65]	0.9709
Discords [58]	0.9706
Proposed representation w/ NN	0.9805
Proposed representation w/ SR	0.9827

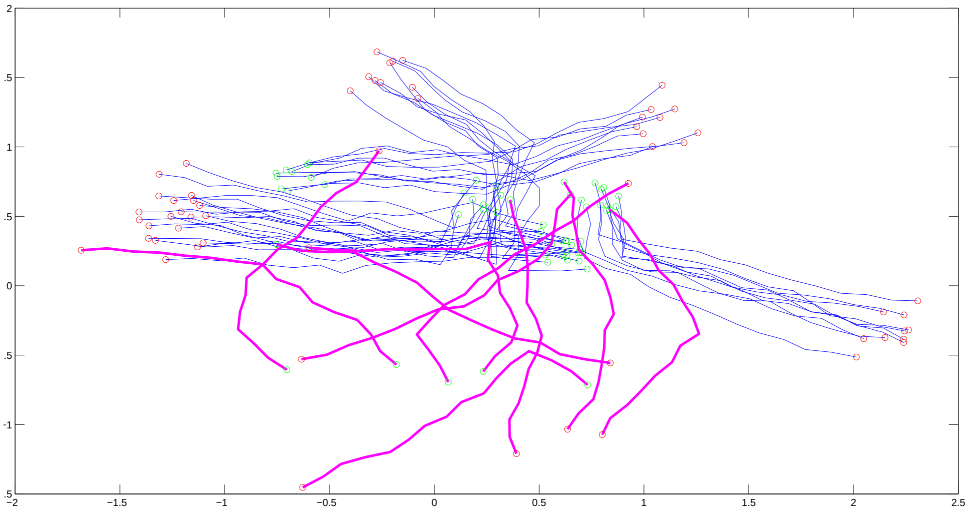


Figure 3.4: A sample result for anomaly detection in the synthetic dataset. Ten samples are shown for each cluster of normal trajectories. Anomalous trajectories are indicated with bold magenta lines.

The synthetic dataset is also utilized for the activity perception part. The previously mentioned 250 non-anomalous trajectories in the dataset belong to five equal size clusters. Similarity matrix shown in Figure 3.6 is used to obtain the clustering result shown in Figure 3.5. Similarity matrix shown in Figure 3.6 also gives an idea about the usefulness of the representation. Correct clustering rate is 0.9055 for the whole dataset which contains 1000 subsets of 260 trajectories.

3.6.2 Results on Real Datasets

Before going into details, it is better to give the reasoning of missing of quantitative results. There is no ground truth data for anomalies or activities in MIT Parking Lot [105] and Train Station (Grand Central) [116] datasets. In UCSD case [75, 68], anomaly ground data are frame based and not appropriate for our approach. Therefore, quantitative results cannot be produced for these datasets.

In UCSD dataset, there are sequences of two scenes. For these scenes, training and test sequences are also provided. Anomalies are motions of non-pedestrian objects such as cars, skaters and bicyclists. A critical issue for UCSD anomaly detection dataset is the extraction of trajectories. For this purpose, KLT tracker used in [115] is exploited to extract the trajectories. After extraction of trajectories for both training

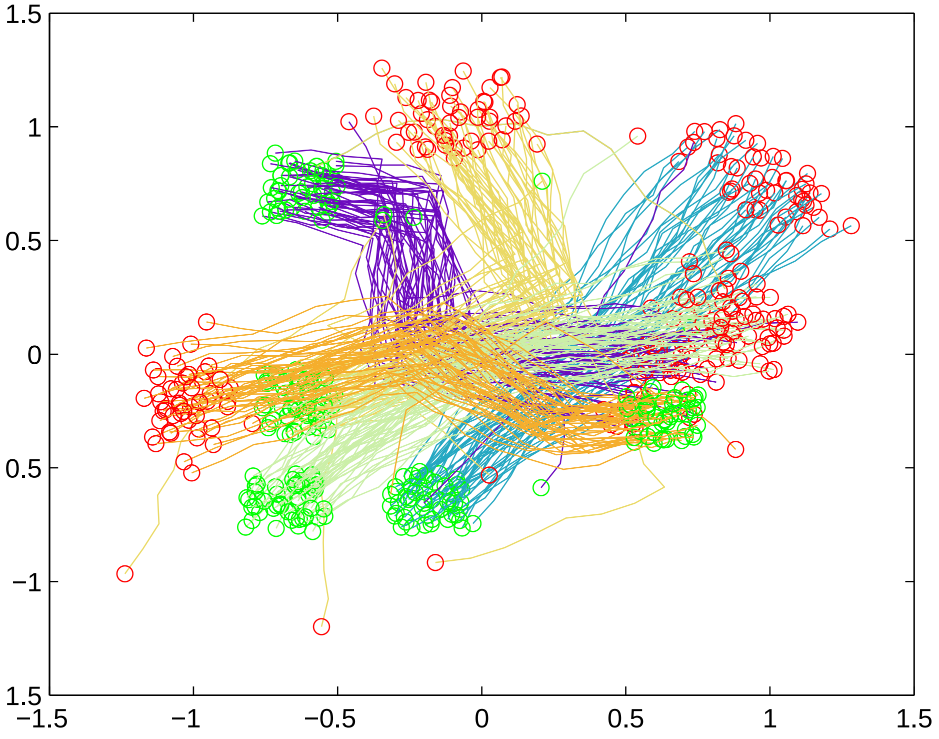


Figure 3.5: Clustering result for a set of trajectories in the synthetic dataset.

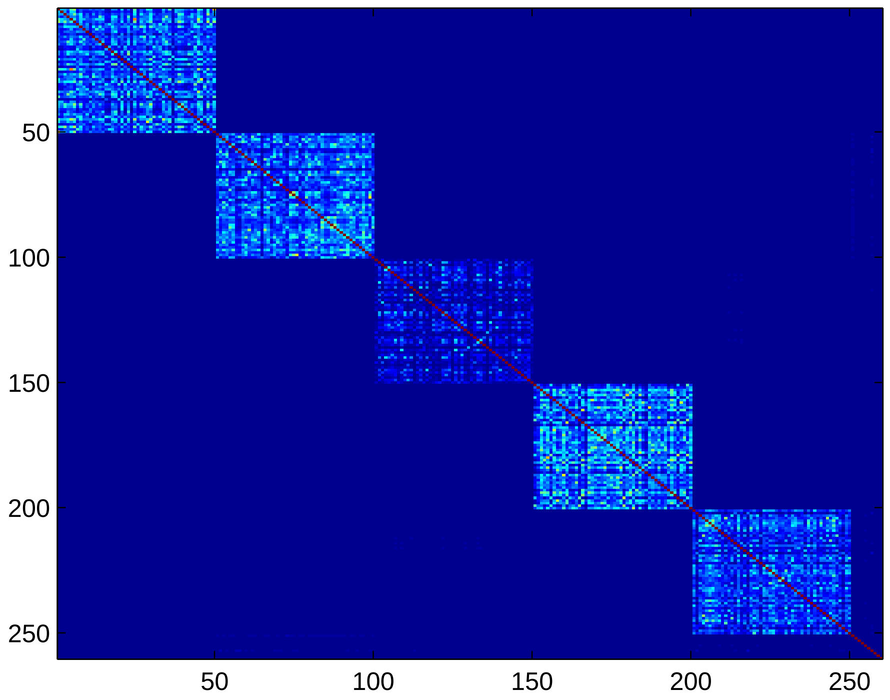


Figure 3.6: Similarity matrix of trajectories given in Figure 3.5. Five clusters can be observed together with the anomalies which lie in the last rows and columns.

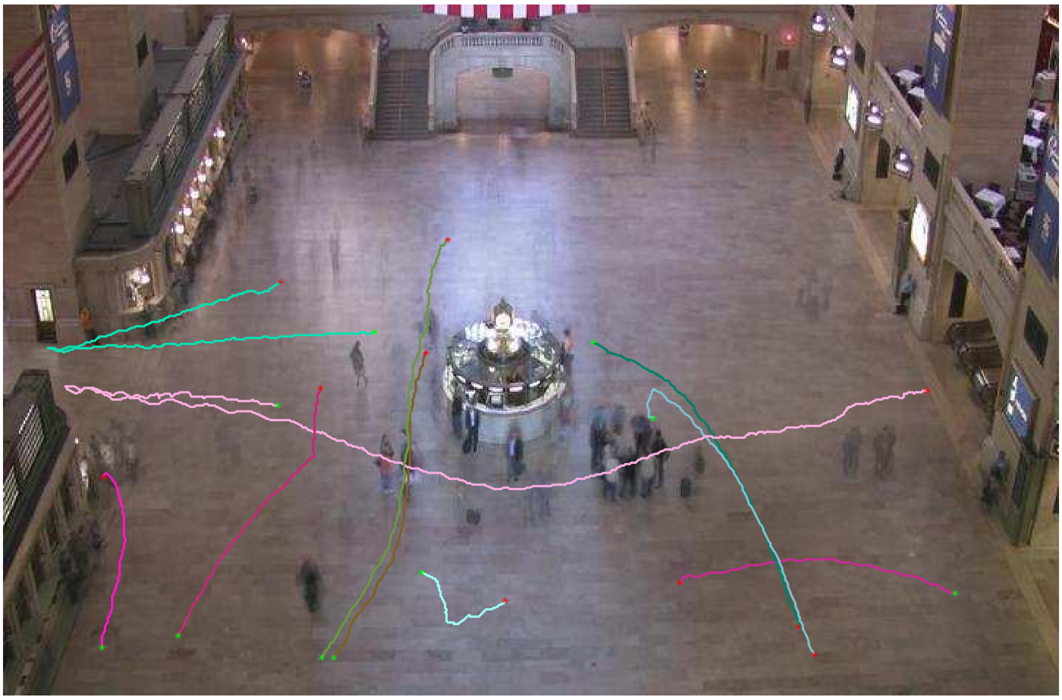


Figure 3.7: Ten most anomalous trajectories in Train Station dataset.

and test sequences, a covariance matrix for each trajectory is calculated. The covariance matrix of each trajectory in the test sequences is compared with all covariance matrices of the trajectories in the training set. Training dataset is not sufficiently large to calculate a sparse representation for this dataset. Therefore, a predefined anomaly measure which is the combination of three nearest neighbors are used in experiments. For some test sequences from two scenes of the dataset, the most anomalous trajectories are shown in Figure 3.8. In Figure Figure 3.8, images in the first two rows indicate the most anomalous trajectories in the folders of Test014, Test019, Test022, Test024 of UCSDped1 scene and the ones in the third and fourth rows are for Test003, Test005, Test006, Test009 of UCSDped2 scene. Starting points are shown with green star and end points with red star, respectively.

Train Station (Grand Central) [116] dataset is also used to detect anomalous trajectories. The dataset provides directly the extracted trajectories. Anomaly detection is again based on distance to three nearest neighbors. In Figure 3.7, ten most anomalous trajectories are shown on the background image provided with the dataset. It should be noticed that starting and ending points for anomalous trajectories are not the normal entrance and exit points.

The next dataset utilized is MIT Parking Lot dataset [105]. This dataset comprises trajectories captured from a parking lot containing trajectories of cars and people. There are certain motion patterns in the scene and the dataset is exploited to detect these motion patterns or activities. In this study, the dataset is used for both of anomaly detection and activity perception. Anomaly detection is again based on distance to three nearest neighbors. Anomaly detection results are shown in Figure 3.9. Some sharp trajectories which are caused by an error in extraction stage are labeled as an anomaly. Activity perception is carried out by a forming a similarity matrix and applying spectral clustering. In the dataset, there are 40453 trajectories and spectral clustering might not be computationally manageable when applied to the whole dataset. However, aforementioned size limit makes the spectral clustering feasible. The activity perception results are given in Figure 3.10. The number of clusters is set to eight to achieve these results in the final k-means step of spectral clustering. Obviously, some clusters contain more than one meaningful motion pattern. It is observed that these motion patterns are extracted when the number of clusters is set to a bigger number. A potential improvement lies in this part of the study. A clustering algorithm without specifying the number of clusters and still works on similarity matrices will be a good alternative to spectral clustering.

3.7 Conclusion

In this study, we propose a novel approach by describing trajectories with feature covariance matrices. We study the problems of anomaly detection and clustering for trajectories in this context. Feature covariance matrices enable us to measure the similarity between trajectories of different lengths. Also, conducted experiments show that covariance descriptor for trajectories yields satisfactory results compared to the state of the art.

We have also introduced a sparse anomaly detector to decide the number and the weights of the nearest neighbors that should be used. This sparse representation can be applied to other similar problems. The only requirement is to have a training dataset for which annotated anomaly data is given.

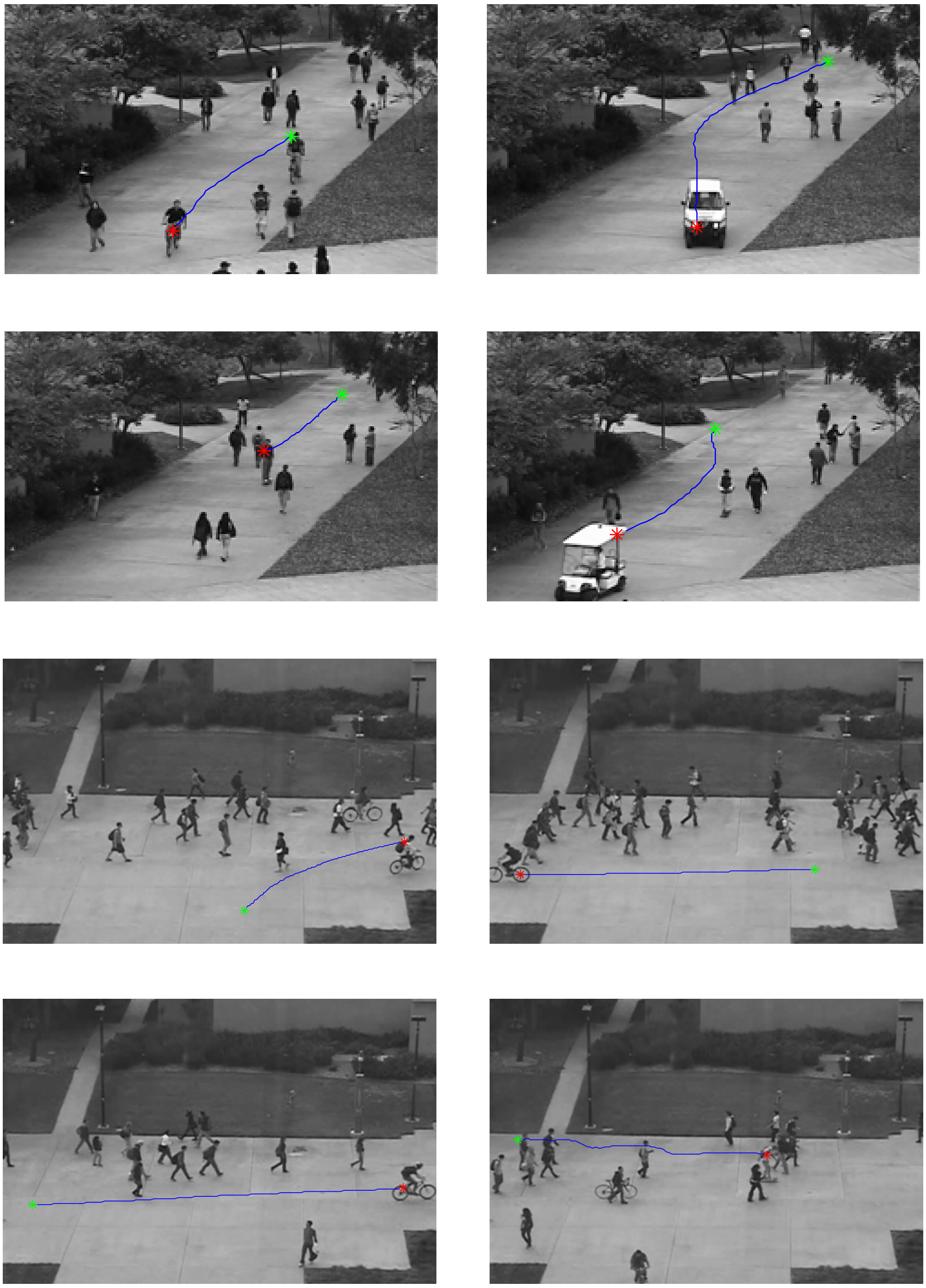


Figure 3.8: Some examples of anomaly detection results in UCSD dataset. Results lied in the rows are from two different scenes in the dataset.

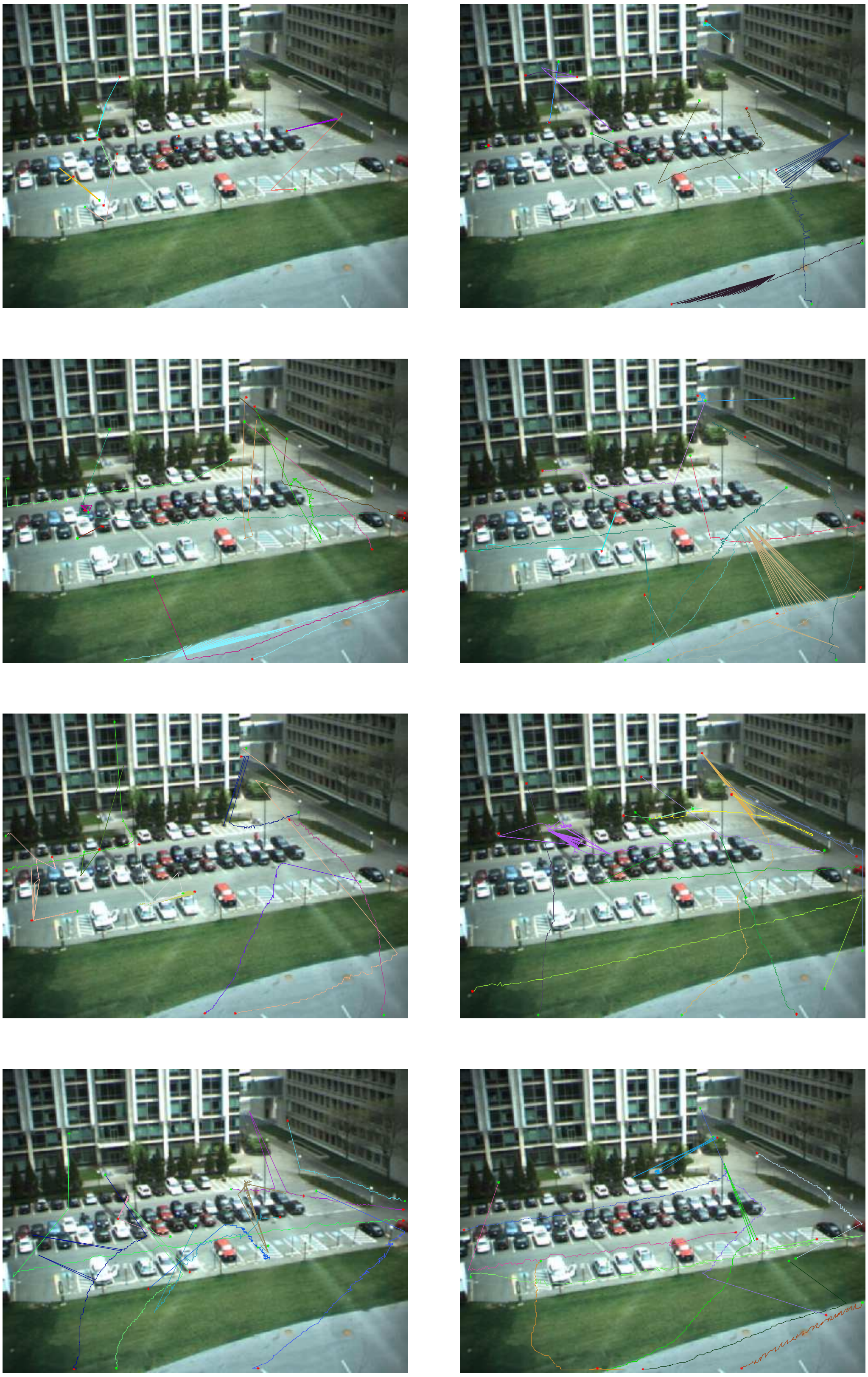


Figure 3.9: Some anomalous trajectories from MIT Parking Lot dataset. These anomalous trajectories might be result of problems in extraction stage.

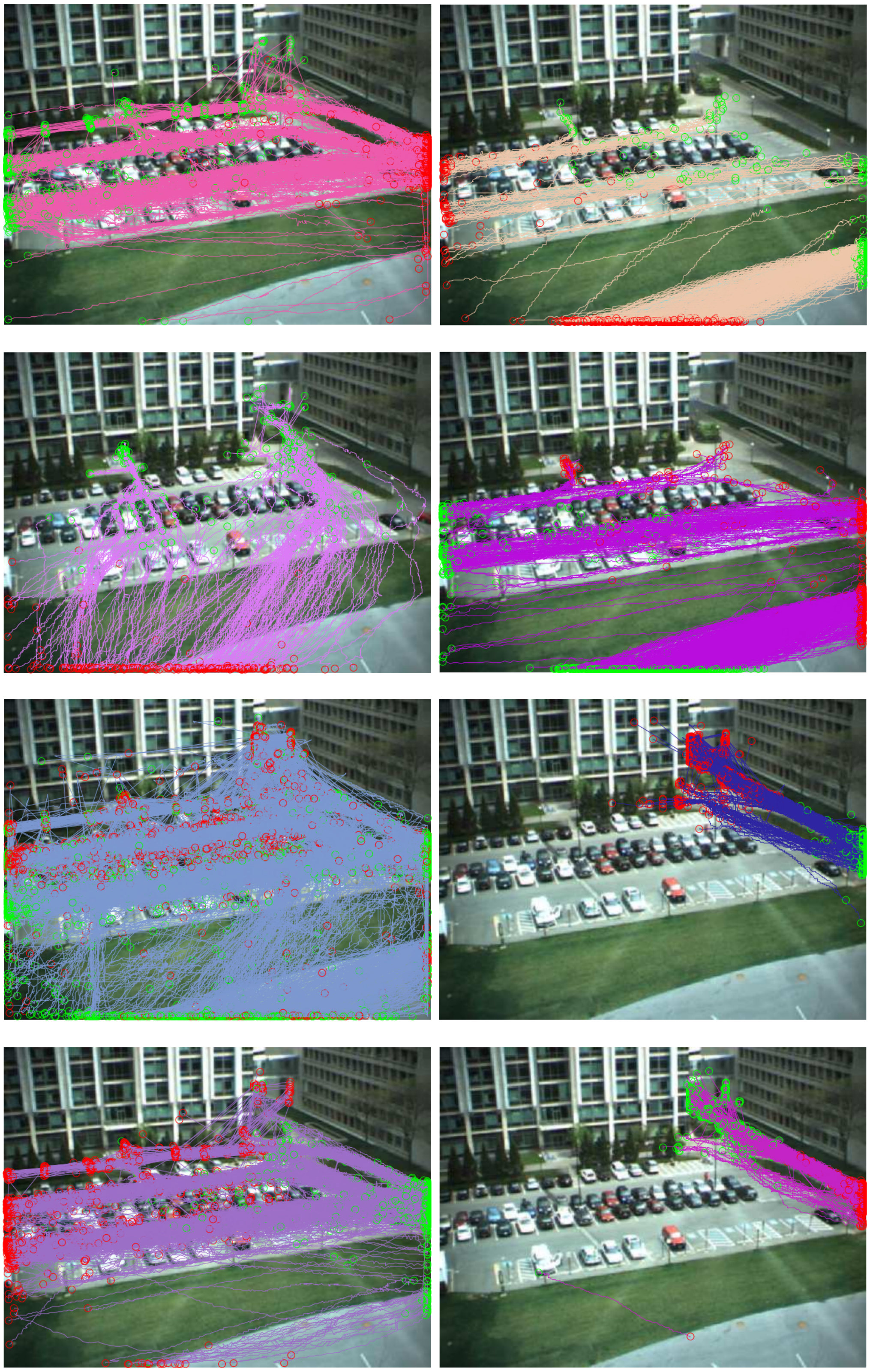


Figure 3.10: Trajectory patterns in MIT Parking Lot dataset.

The whole study has been conducted with the assumption that crowd density allows extracting trajectories for each object. In dense crowd scenarios, other approaches such as particle advection used in [108] might be more feasible to obtain trajectories.

There is a possible improvement in activity perception part of this study. Instead of standard spectral clustering approach for which the number of clusters must be given, another clustering approach on distance matrix calculated with the representation can be used.

A representation is proposed for time series of 2D data in this study. A possible extension of this work is the shape classification problem. Although there is no time information and invariance on rotation and scale could be problematic, feature covariance matrices can be used to describe 2D shapes.

CHAPTER 4

TIME SERIES CLASSIFICATION WITH FEATURE COVARIANCE MATRICES

In this chapter, a novel approach utilizing feature covariance matrices is proposed for time series classification. In order to adapt the feature covariance matrices for time series classification, a feature vector is defined for each point in a time series. The feature vector comprises local and global information such as value, derivative, rank, deviation from the mean, time index of the point and cumulative sum up to the point. Instead of representing the whole time series with a single covariance matrix, time series is divided into overlapping subsequences. Extracted feature vectors for the time instances are concatenated to construct feature matrices for the overlapping subsequences. Covariance of the feature matrices are used to describe the subsequences. After the determination of feature covariance matrices for both training and test samples, distances are calculated by using log-Euclidean distance measure. Our main purpose in this work is to introduce and evaluate the feature covariance representation for time series classification problem. Therefore, in classification stage, first, 1-NN classifier is used. SVM classifier is also utilized for the problem. Conducted experiments on UCR time series dataset show that the proposed method yields results which mostly outperform well-known methods such as DTW, shapelets and other state-of-the-art techniques.

4.1 Introduction

Time series analysis has received great interest over the past decades from several disciplines including biology, medicine, economics, etc. In tandem with the increasing availability of digital data, it is highly possible that this problem will gain much more attention. Time series classification is one of the most important problems in time series analysis [89, 35]. There are two main issues in time series classification as in other time series analysis problems: representation of time series and similarity between time series [89, 35].

Time series classification can be treated as a supervised learning problem. From this perspective, aforementioned common issues related to time series analysis correspond to a feature extraction step and an appropriate classification approach. The feature extraction step is needed to extract the discriminative part of the time series as in problems in computer vision [77] and speech recognition [8]. The time series is then represented with the extracted features for further tasks such as indexing, clustering and classification. In addition, by representing the data in a feature space, dimensionality reduction can also be achieved for most of the cases.

In the last two decades, several feature extraction methods [77, 73, 64, 100, 13, 27] have been proposed for detection and classification problems in images and videos. Although some visual features are utilized for the problems in time series domain [98, 96, 86], feature extraction methods have not become popular yet when time series classification is considered. There are recent studies [114, 10] applying well-known features in computer vision to the time series classification problem; however, some important techniques remain untouched for the time series domain. Covariance descriptor, which is the utilized technique in this study, is one of them that can be efficiently applied to time series classification problem.

Previously, covariance descriptor has been used to describe regions for images [100, 99, 87, 33] and video blocks in capturing different types of actions [40]. Covariance descriptor is exploited as a high-level feature in all these applications. It captures the pairwise correlations of the basic image or video features. Similarly, for representing the time series, it captures the pairwise correlations between the pointwise features. Covariance descriptor presents a compact representation due to its symmetry. It provides a low-dimensional and fixed-size covariance matrix which is independent of the length of time series. Also, it is robust to noisy inputs since it includes an inherent averaging in covariance computation.

In this chapter, we propose a novel approach for time series classification problem. In previous chapter, feature covariance matrices were used to represent trajectories which are basically 2D time series. It was shown that representing trajectories via feature covariance matrices outperforms the state-of-the-art methods for anomaly detection and activity perception problems. We utilize the same representation for time series classification problem by adapting it to subsequences and defining novel point-wise features. First, feature covariance matrices are determined for overlapping subsequences of both test and training samples. Overlapping subsequences are selected in order to cover all discriminative portions of the time series. While building covariance descriptor, a feature vector is defined for each point in the subsequence. Feature vectors are then combined to determine the feature matrix. Covariance of the feature matrix is utilized to represent the subsequence. In classification stage, two well-known classifiers are utilized. For 1-NN classifier, the distance between training samples and a test sample is determined by using Log-Euclidean distance metric. For SVM classifier, the upper (or lower) triangular part of the covariance matrices of the subsequences are concatenated. Comparative experiments using 43 different datasets are conducted. We achieve mostly outperforming results compared to well-known methods such as DTW, shapelet transform and state-of-the-art methods [114, 14]. The proposed approach is also very efficient in computation time compared to other methods. The computation time efficiency is due to the fact that overall method is simply based on a feature covariance representation.

The main contribution of the chapter is our novel time series classification approach which is based on covariance descriptor. To the best of our knowledge, this is the first method where each point of the time series is used for representation and feature extraction. Another important contribution is that a novel distance measure is proposed for time series. The distance obtained by covariance descriptor can be applied to time series of different lengths. Lastly, our results are mostly outperforming the well-known and state-of-the-art methods.

The remainder of this chapter is organized as follows. A summary of previous works on time series classification is provided in the next section. Section 4.3 presents the proposed representation based on covariance descriptor and how it is applied to time series classification. How we utilize two classifiers is summarized in Section 4.4.

Conducted experiments are given in Section 4.5. Conclusions and possible future work are provided in Section 4.6.

4.2 Related Work

There are several previous works that address the time series classification problem. A review of representations and distance measures for time series is provided in [107]. In our perspective, time series classification is a supervised learning problem. As other supervised learning problems, time series classification is conceived as a two-fold problem in this work: representation and classification. However, there are many studies handling time series classification problem from different perspectives. Studies are divided into three categories according to how they approach the problem. The first category includes the distance-based approaches [60, 61, 52]. In the second category, approaches seek a better representation for time series [110, 43, 81, 69, 14, 114, 10]. Approaches in the last category mostly focus on the classification part of the problem and apply various classification methods like random forests [29], ensembling [9, 70, 39].

The most direct solution for time series classification problem is measuring the Euclidean distance between time instances and applying nearest neighbor classifier. However, this solution is susceptible to time distortions such as shifting, stretching and contracting. Dynamic time warping (DTW)[60] is used to mitigate these distortions. DTW measures the similarity between time series by searching the optimum map over the points in a time series. In [61], authors propose a modification of original DTW methodology to handle the singularities by utilizing first-order differences. On the other hand, a weighted version of DTW is used to give different weights to warping distances in [52]. In a very recent work [55], DTW distance is utilized as a feature for time series classification. Lastly, Wang et al. compare the several distance measures including DTW and its modified versions and report their time series classification performances in [107].

Representation-based approaches are divided into three subcategories: shapelet transform, bag-of-words and feature-based approaches. Shapelet is a discriminative subsequence of time series. The main idea behind the approach is to find a subsequence which has a smaller distance to one class of time series compared to other ones. The first shapelet-based approach was proposed in [110] to reduce computation complexity of preceding methods and get more insights about data. In this pioneering work of Ye and Keogh [110], the discovery of the shapelets is embedded into a decision tree classifier. Mueen et al. [81] propose logical shapelets to reduce the computation time of shapelet discovery. They discover the conjunction or disjunction of shapelets instead of discovering a single shapelet. However, the method is still based on an enumerative search. The approach proposed in [43] finds top k shapelets in a single run and utilizes the transformed data onto these top k shapelets while classifying the time series. In a more recent study [9], the data after shapelet transformation are one of the blocks fed into an ensemble classifier.

Another representation-based approach for time series classification is based on bag-of-words or bag-of-features models. Bag-of-words models are used in several tasks including the examples of image retrieval [111], object detection [78] and music classification [38]. An approach called symbolic aggregate approximation (SAX) [69] that attacks several time series data mining problems follows a similar path for time series classification. In this approach, words are generated using the symbols for fixed-size windows and time series is represented by these words. Authors use different distance measures for different time series data mining problems. In [14], some features such as mean, deviation and slope of the fitted regression line are put in a bag to represent a randomly selected subsequence. After extracting features for subsequences, random forest classifier is used to generate the codebooks for subsequences and classify the time series.

Feature-based approaches generally consist of two steps: extraction of local or global features and a classifier. The classifier is trained using the extracted features from a training set. Some recent works adapt the proven descriptors in computer vision to the time series [114, 10, 97]. Histogram of gradient (HOG) descriptor [27] is applied to 1D time series and is utilized in time series classification in [114]. In [114], there are also descriptors which are used to represent the subsequences. The fused descriptor is the input of a Fisher vector encoding followed by a linear kernel SVM. Scale-invariant feature transform (SIFT) [73] is another popular descriptor used in several computer vision problems. It is applied to time series classification with a bag-of-features approach in [10]. A more recent work [55] uses the DTW distance measure as a feature. A time series is represented in terms of its DTW distances from each of the training examples. Our approach can also be categorized as a feature-based approach. It learns covariance features of subsequences from a training set and inputs these covariance matrices to a classifier to decide the class of test samples.

For the classification part of the problems, most of the well-known classifiers such as SVM [114], random forests [54], NN [88] are used for time series classification problem. In [4], the authors present a benchmark of approaches for time series forecasting. Since regression and classification are very similar problems, the approaches and their results mentioned in [4] provide a vision for time series classification problem. On the other side, there are some recent works [9, 70, 39] that utilize the ensembling strategy. Thirty-five classifiers constructed in time, frequency and shapelet transformations are combined in [9]. Classifiers are combined according to their training set cross-validation accuracy. In [39], authors exploit the training set to achieve the most informative features from thousands of interpretable features. The most informative features for each class in a time series are found using greedy feature selection with a linear classifier. Another ensemble method proposed in [70] combines the basic distance measures including two aforementioned variants of DTW and edit distance-based measures. As expected, all three methods that exploit the ensemble strategy achieve significantly better results compared to other methods.

4.3 Representation of Time Series by Feature Covariance Matrices

In our previous work [34], feature covariance matrices were used to represent trajectories in image sequences, which are basically 2D time series. It was shown that representing trajectories via feature covariance matrices provided state-of-the-art results for anomaly detection and activity perception problems. Similarly, in this work, feature covariance matrices are used to represent 1D time series. A set of analogies is built up between computer vision and time series domains, such that the concept of time series-subsequence-point triplet of a time series signal is inherited from the concept of an image-region-pixel triplet in computer vision. More clearly, covariance descriptor is used to represent an image region [99] or a video block [40] in computer vision problems. In a similar fashion, in this study, it is utilized to represent the subsequences of a time series as shown in Figure 4.1. Covariance descriptor is built by determining the covariance of a feature matrix comprising feature vectors defined for each point in the subsequence. Analogous to the selection of regions in computer vision problems [100, 99], overlapping subsequences are selected so that discriminative portions of the data are not missed. A parallel analogy is pursued in the selection of pointwise features. Pointwise features that are used to represent time series are analogous to pixel features such as pixel value, pixel coordinates, optical flow, etc.

Covariance descriptor is based on the covariance of feature matrices. Covariance matrices are semi-positive definite matrices. As other semi-positive definite matrices, covariance matrices lie on Riemannian manifold space. Therefore, before getting into the details of the proposed approach, we provide a brief summary of Riemannian manifolds. Riemannian manifolds are located in the third layer of a manifold hierarchy. Topological manifolds are locally Euclidean spaces which are located in the first layer of the hierarchy. In the second layer, differentiable (or smooth) manifolds are located. Differentiable manifolds are topological manifolds, for which some calculus operations such as the derivative can be defined. Riemannian manifolds are differentiable manifolds, for which distance metrics and angles can be defined.

The critical point is the calculation of distances between covariance matrices while using them for a classification problem. Before explaining the distance measure, a more illustrative definition of the Riemannian manifold can be given as follows: A Riemannian manifold is a differentiable manifold equipped with the inner product on tangent space at each point. The distance measure is based on exponential and logarithmic maps between two points on Riemannian manifolds. It should be noted that these definitions are introductory basics about manifold geometry. More information on Riemannian manifolds can be found in [66] and in the second chapter of [36].

Time series can be considered as a sequential concatenation of several points. Each point has its own features such as value, slope, distance to mean, time index, etc. These can be used for the classification problem since they carry important information about the series. Moreover, the change of these pointwise features with respect to each other carries another important information about time series. The main idea behind the use of covariance matrix as a descriptor is that it captures these pairwise correlations between the pointwise features.

Time series can be defined as a sequential combination of M points or formally as a vector of length M ($[x_1,\ldots,x_M]$). Feature candidates can be merged in a feature vector for a point in time series. Let the number of features, $\{s_i\}$, defined for a point be K. The feature vector for $N\mathrm{th}$ point of the subsequence can be shown as

$$f_N = [s_{N1}, \ldots, s_{NK}] \tag{4.1}$$

When feature vectors are combined for all points, we end up with a feature matrix F,

$$F = \begin{bmatrix} s_{11} & \ldots & s_{1K} \\ & . & \\ & . & \\ & . & \\ s_{M1} & \ldots & s_{MK} \end{bmatrix} \tag{4.2}$$

The covariance of the feature matrix is calculated as

$$C = \frac{1}{M-1} \sum_{i=1}^{M-1} (F_i - \mu)(F_i - \mu)^T \tag{4.3}$$

where μ is the mean vector of feature vectors $\{f_1, \ldots, f_M\}$.

This general representation is adapted for time series classification problem by dividing the time series into L overlapping subsequences each of them has a length M. A point in a subsequence is defined by its features, namely its value, derivative, cumulative sum, the difference between mean, rank and time index. The features carry both local and global information about the point. Value, derivative and time index features are local features since they depend only on the point itself and in certain cases on its adjacent point. On the other hand, cumulative sum, the difference between mean and rank depend on almost all points in the subsequence, and hence, they can be called as global features. The feature vector with these pointwise features is formed as

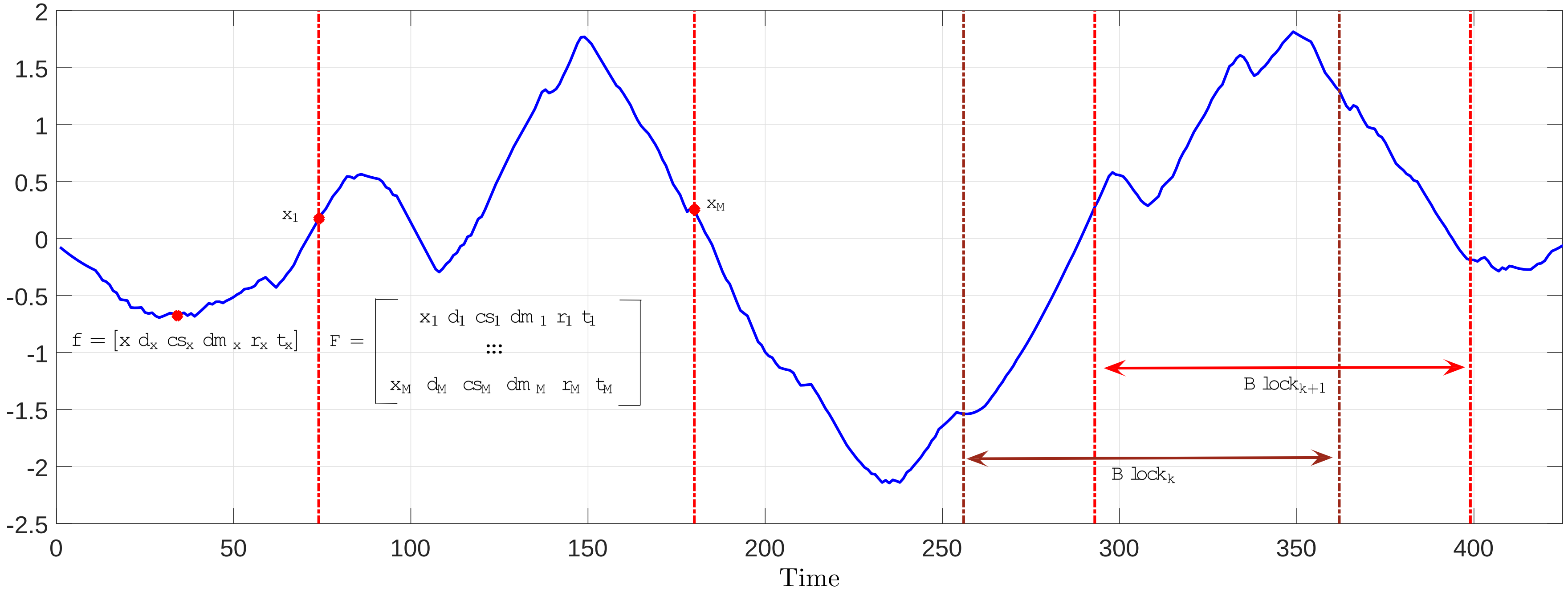


Figure 4.1: Representation of time series with feature covariance matrices is depicted in *yoga* dataset of UCR repository. First, a feature vector is defined for each point in the time series. Feature vectors defined for each point of time series form feature matrices for overlapping subsequences. For each subsequence, covariance matrices are calculated and the distance between subsequences in training and test sets are calculated using log-Euclidean distance metric. The distance between the time series is determined by averaging the distances between the subsequences.

$$f = [x_T \; d_T \; cs_T \; dm_T \; r_T \; T] \tag{4.4}$$

where x_T is the value, d_T is the derivative at a specific point at time T

$$d_T = x_T - x_{T-1} \tag{4.5}$$

The third feature in the feature vector is cumulative sum up to a point and is given by Eq. (4.6) at point T

$$cs_T = \sum_{t=1}^{T} x(t) \tag{4.6}$$

The next pointwise feature is the difference between the mean of the subsequence and value at point T. This pointwise feature indicates the distinctiveness of the point in the subsequence, and it is calculated as

$$dm_T = x(T) - \frac{1}{M}\sum_{t=1}^{M} x(t) \tag{4.7}$$

where M is the length of subsequence. The feature of rank, r_T, is the $N\text{th}$ biggest value of the subsequence. We have also inserted time directly to the feature vector to show us how other features change in time. It should be noted that the time feature is normalized within the subsequence. For each subsequence, time index starts from 2, due to the calculation of the derivative, and goes up to M. Throughout the experiments, the effects of several other features including the second derivative, the difference between the maximum etc., are evaluated as well. However, the best classification performance is obtained with the proposed feature set. Besides, a detailed analysis of the effects of selected pointwise features is given in Section 4.5. After determination of all pointwise features for a whole subsequence, feature matrix is defined by

$$F = \begin{bmatrix} x_2 \;\; d_2 \;\; cs_2 \;\; dm_2 \;\; r_2 \;\; 2 \\ . \\ . \\ . \\ x_M \; d_M \; cs_M \; dm_M \; r_M \; M \end{bmatrix} \tag{4.8}$$

Covariance of the feature matrix defined in Eq. 4.8 is utilized to represent each subsequence of time series. By doing so, time series are carried onto the Riemannian manifold space. Each subsequence of the time series corresponds to a point in the manifold. The covariance of the feature matrix is calculated as in Eq. 4.3. At this point, a small multiple of the identity matrix is added to covariance matrices for all subsequences in test and training sets. This regularization is performed to ensure the positive definiteness of the covariance matrix. Positive definiteness is crucial for the distance metric which involves a logarithm operation.

The covariance descriptor brings some advantages for time series classification problem. The covariance matrix enables us to combine multiple feature vectors. The dimension of the covariance matrix depends only on the dimension of the feature vector, f, given in Eq. 4.4. The covariance matrix C is a $d \times d$ matrix when f is d-dimensional. Also, due to its symmetry, C has only $(d^2 + d)/2$ independent values. As stated before, dimension reduction is one of the main goals of feature extraction. When compared to the dimension of time series or even the dimension of the subsequences, C lies in a lower dimensional space. Last but not least, the novel part of the representation with covariance matrix is that it maps the time series into a fixed-size vector space, which is independent of the series' length.

4.4 Time Series Classification with Feature Covariance Matrices

After representing 1D time series with feature covariance matrices, two well-known classification techniques are utilized. First, the 1-NN classifier is utilized by considering it one of the simplest classification methods. The main goal of the utilization of the 1-NN classifier is to evaluate the effectiveness of the proposed representation and isolate the performance of the overall method from the performance of the classifier.

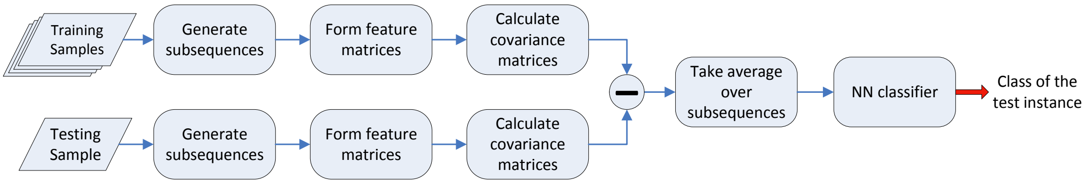


Figure 4.2: Block diagram of the overall approach for time series classification with 1-NN classifier.

After determining the satisfactory results with the 1-NN classifier, SVM classifier is used as a more complex classification technique. In this section, we give the details about how classifiers are applied to the problem.

4.4.1 Classification with 1-NN classifier

For 1-NN classifier, the most crucial step is the calculation of the distances between the covariance matrices of training samples and a test sample. As stated before, a distance measure that approximate the geodesic distance between two points on Riemannian manifolds must be used. For this purpose, similarly to [100, 87, 40] that utilize covariance matrices, Euclidean distance metrics must be avoided. We use the log-Euclidean metric which was firstly proposed in [5]. Compared to other distance metrics [37] and divergence functions [20], log-Euclidean metric gives the best performance in our experiments.

Log-Euclidean distance metric is based on matrix logarithm operation. Matrix logarithm operation maps covariance matrices from a Riemannian manifold to Euclidean space. The determination of the log-Euclidean metric starts with the eigenvalue decomposition of the covariance matrix.

$$C = VDV^T \tag{4.9}$$

After this eigenvalue decomposition, matrix logarithm is obtained as

$$\log(C) \triangleq V\widetilde{D}V^T \tag{4.10}$$

where $\widetilde{D}$ is a diagonal matrix obtained from D by replacing its diagonal entries by their logarithms. The distance between two covariance matrices is calculated via Frobenius norm of the distance between matrix logarithms.

$$\rho(C_{ik}, C_{jk}) = \|\log(C_{ik}) - \log(C_{jk})\|_F \tag{4.11}$$

where i and j indicate train and test instances of $k\text{th}$ subsequence of time series.

Now, we have all the distances between subsequences via their feature covariance matrices. The distance between the whole time series is determined by averaging the distances between subsequences.

$$\rho(T_i, T_j) = \frac{1}{L}\sum_{k=1}^{L} \rho(C_{ik}, C_{jk}) \tag{4.12}$$

where L is the number of subsequences. Here, the average value of the distances between the subsequences is used as the distance between two time series. Also, there are other options like taking minimum or maximum of the distances as the final distance. However, averaging the distances hinders the domination of a single subsequence.

After obtaining the final distance between time series, 1-NN classification is applied to determine the class of a test sample. In other words, the class of a sample in the test set is assigned to the class of its nearest neighbor in the training set using the distance measure explained above.

4.4.2 Classification with SVM classifier

SVM classifier is generally the first solution that comes to mind for a classification problem. In our case, after determining the covariance matrices of the subsequences, the critical question is how we build the model. There are some recent studies [11, 90] that aim kernel learning over the manifolds. However, there is a more direct way. Utilizing symmetric property of covariance matrices, lower (or upper) diagonal parts of the covariance matrices are fed into the SVM classifier as in [11]. In a more formal definition, for a time series dataset for which the number of subsequences is L and the dimension of the feature vector is d, the dimension of the input vector fed into the SVM classifier is $L * \frac{d(d+1)}{2}$. Again, as in [11], the non-diagonal elements are multiplied with $\sqrt{2}$ to keep the equality of norms.

SVM is originally a binary classification technique. There are some approaches while utilizing SVM in a multi-class classification problem. One-versus-all (or one-against all) is one of these approaches. In this work, we utilize one-versus-all approach for time series classification. In this approach, for each class, a classifier is trained. For the $i\mathrm{th}$ classifier, let the positive examples be all the points in class i, and let the negative examples be all the points not in class i. In order to make a decision for an unseen sample x, we take the corresponding classifier which reports the highest confidence score

$$y = \underset{i=1,\ldots,N}{\arg\max} f_i(x) \tag{4.13}$$

where $f_i(x)$ is the$i\mathrm{th}$ classifier and N is the number of classifiers or classes.

4.5 Experiments

In this section, we evaluate the performance of the proposed approach by using the datasets in the UCR repository [23, 1]. The UCR repository consists of a diverse set of 85 datasets which grouped into six different types. This diversity in the type of UCR datasets leads to high variability in series length and character as depicted in Figure 4.4 and summarized in Table 4.1. Due to this variability, the number of subsequences, overlap ratio between the subsequences and the regularization constant have been obtained for each dataset by cross-validation. For this purpose, for both classifier, the original training set is split into equally sized validation and training sets. This procedure is performed 100 times by randomly selecting validation and training sets from the original training set. The optimum parameters are based on the average classification errors acquired in these simulations. All the other parameters are kept fixed for training and test sets in a dataset. Lastly, for SVM classifier, LIBSVM [2] is utilized during the experiments.

In our experiments, we have utilized 43 of these 85 datasets for which the clas-

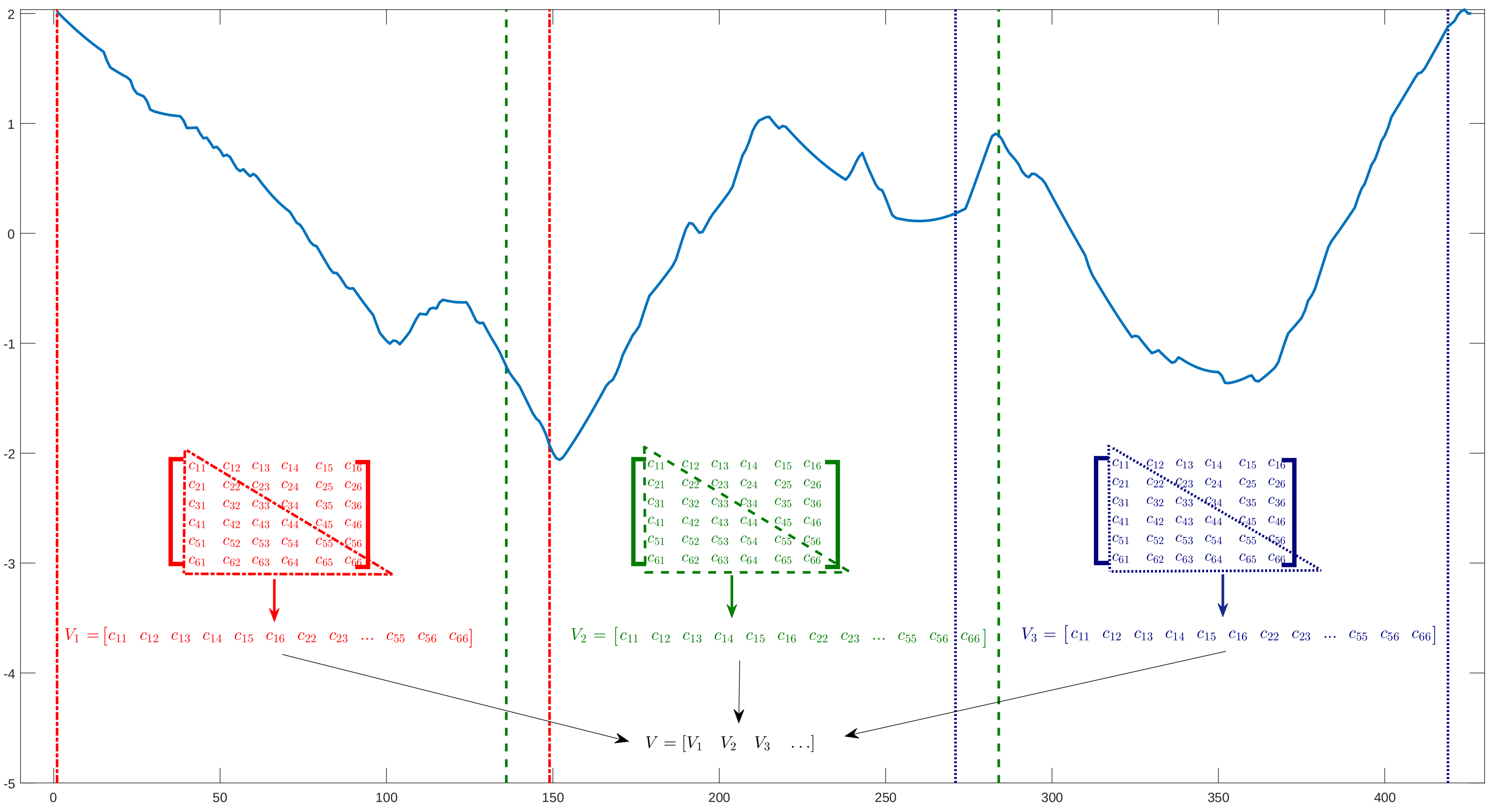


Figure 4.3: Time series classifcation with SVM classifer. The model for SVM is the concatentation of the vectorized feature covariance matrices.

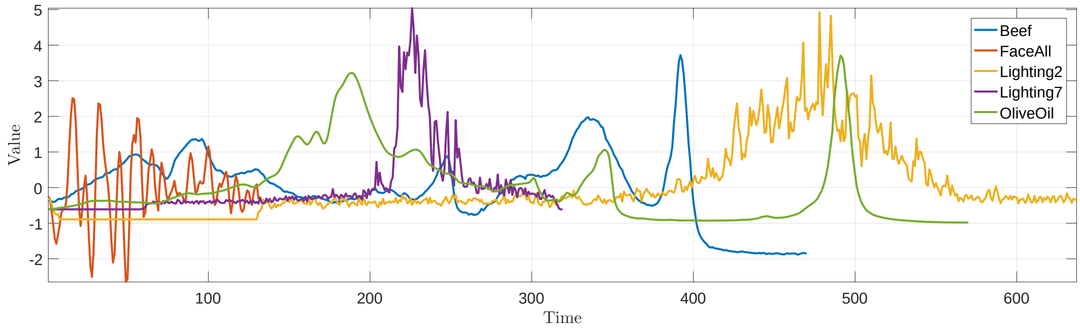


Figure 4.4: Five samples from UCR time series dataset. Diversity in length and variability for the datasets can be observed.

sification results of all the compared methods have been previously published. In the following subsection, we first compare the performance of the proposed method with well-known and the state-of-the-art methods and show that the proposed method mostly outperforms the compared methods. In the second subsection, an analysis is presented with different combinations of pointwise features. The computational complexity of the proposed method is analyzed and compared with another feature-based method given in [114] in the last subsection.

Table 4.1: Information about the datasets used in experiments

Dataset	**No.of Classes**	**Train Size**	**Test Size**	**Length**	**Type**
50 Words	50	450	455	270	Image
Adiac	37	390	391	176	Image
Beef	5	30	30	470	Spectro
CBF	3	30	900	128	Simulated
ChlorineCon	3	467	3840	166	Simulated
CinCECGTorso	4	40	1380	1639	ECG
Coffee	2	28	28	286	Spectro
CricketX	12	390	390	300	Motion
CricketY	12	390	390	300	Motion
CricketZ	12	390	390	300	Motion
DiatomSizeR	4	16	306	345	Image

ECG200	2	100	100	96	ECG
ECGFiveDays	2	23	861	136	ECG
FaceAll	14	560	1690	131	Image
FaceFour	4	24	88	350	Image
FacesUCR	14	200	2050	131	Image
Fish	7	175	175	463	Image
GunPoint	2	50	150	150	Motion
Haptics	5	155	308	1092	Motion
InlineSkate	7	100	550	1882	Motion
ItalyPower	2	67	1029	24	Sensor
Lightning2	2	60	61	637	Sensor
Lightning7	7	70	73	319	Sensor
MALLAT	8	55	2345	1024	Simulated
MedicalImages	10	381	760	99	Image
MoteStrain	2	20	1252	84	Sensor
OSULeaf	6	200	242	427	Image
OliveOil	4	30	30	570	Spectro
SonyAI	2	20	601	70	Sensor
SonyAI-II	2	27	953	65	Sensor
StarLightC	3	1000	8236	1024	Sensor
SwedishLeaf	15	500	625	128	Image
Symbols	6	25	995	398	Image
Trace	4	100	100	275	Sensor
TwoLeadECG	2	23	1139	82	ECG
TwoPatterns	4	1000	4000	128	Simulated
SynthControl	6	300	300	60	Simulated
UWaveX	8	896	3582	315	Motion
UWaveY	8	896	3582	315	Motion
UWaveZ	8	896	3582	315	Motion
Wafer	2	1000	6174	152	Sensor

WordSynonyms	25	267	638	270	Image
Yoga	2	300	3000	426	Image

4.5.1 Comparative Results

The performance of the proposed method is compared to four different methods. Two well-known methods, dynamic time warping (DTW) and shapelet transform (ST), are selected for comparison similarly to state-of-the-art studies [114, 14]. These two methods provide the best classification performance for some of the selected datasets in the repository. The other two methods are feature-based methods [114, 14] and chosen due to their similarities with the proposed approach. Below, a brief explanation on the reported performance of each compared method is provided. We preferred not to include the Euclidean distance measure in the results because of its low performance for all datasets compared to the state-of-the-art methods including the proposed method.

DTW: In this work, the results of the version of DTW with warping window [88] are used. DTW competes with the state-of-the-art methods with the 1-NN classifier. The reported classification success of this combination has the best performing results for some of the datasets as can be seen in Table 4.2.

Shapelet Transform (ST): The results of shapelet transform are based on the results given in University of East Anglia website.[1] The classification result of ECG200 dataset is missing for this approach. Therefore, all comparisons with the shapelet approach are evaluated for 42 datasets only.

TSBF: In [14], authors report their results for both uniform and random selection of features in 45 datasets. Their results show that random selection of features gives better results for most of the datasets. Therefore, we take the results with randomly selected features similarly to other methods [114, 9] that compare their methods with TSBF.

HOG1D+DTW-MDS: Provided that the approach presented in [114] is achieved by

---

[1] https://www.uea.ac.uk/computing/machine-learning/shapelets/shapelet-results

combining a set of features, the authors presented their results for each combination separately. The best results are achieved for the case where all features are used. The results of this combination for 43 datasets are selected for comparison.

Table 4.2: Misclassification rates of the methods for a subset of the datasets in UCR repository.

Dataset	**DTW**	**ST**	**TSBF**	**HOG1D**	**CovNN**	**CovSVM**
50 Words	0.242	0.281	0.209	0.402	0.222	**0.200**
Adiac	0.391	0.435	0.245	0.320	0.217	**0.164**
Beef	0.467	0.167	0.287	0.367	0.100	**0.067**
CBF	0.004	0.003	0.009	**0.000**	**0.000**	**0.000**
ChlorineCon	0.350	0.300	0.336	0.307	0.294	**0.255**
CinCECGT	0.070	0.154	0.262	0.249	0.003	**0.000**
Coffee	0.179	**0.000**	0.004	**0.000**	**0.000**	**0.000**
CricketX	0.236	0.218	0.278	**0.195**	0.244	0.236
CricketY	**0.197**	0.236	0.259	0.205	0.210	0.251
CricketZ	**0.180**	0.228	0.263	0.185	0.239	0.215
DiatomSizeR	0.065	0.124	0.126	**0.016**	0.052	0.043
ECG200	0.310	-	0.145	**0.060**	0.080	0.070
ECGFiveDays	0.203	**0.001**	0.183	0.012	0.116	0.002
FaceAll	0.192	0.263	0.234	**0.082**	0.194	0.199
FaceFour	0.114	0.057	0.051	0.034	0.023	**0.000**
FacesUCR	0.088	0.087	0.090	0.090	**0.066**	**0.066**
Fish	0.160	**0.023**	0.080	0.034	0.074	0.051
GunPoint	0.087	0.020	0.011	0.007	**0.000**	**0.000**
Haptics	0.588	0.523	0.488	**0.471**	0.558	0.520
InlineSkate	0.613	0.615	0.603	**0.551**	0.598	0.600
ItalyPower	0.045	0.048	0.096	0.070	0.035	**0.030**
Lightning2	**0.131**	0.344	0.257	0.148	**0.131**	0.148
Lightning7	0.288	0.260	0.262	0.205	0.178	**0.151**
MALLAT	0.086	0.060	0.037	**0.035**	0.042	**0.035**

MedicalImages	0.253	0.396	0.269	**0.230**	0.262	0.258
MoteStrain	0.134	0.109	0.135	0.090	**0.074**	0.084
OliveOil	0.167	0.100	0.090	0.167	**0.033**	0.033
OSULeaf	0.384	0.285	0.329	**0.120**	0.281	0.273
SonyRobot	0.305	0.067	0.175	**0.042**	0.107	0.118
SonyRobotII	0.141	0.115	0.196	0.084	0.084	**0.078**
StarLightC	0.095	0.024	**0.022**	0.040	0.027	0.027
SwedishLeaf	0.157	0.093	0.075	0.061	0.066	**0.046**
Symbols	0.062	0.114	0.034	0.036	0.030	**0.022**
SynthCtrl	0.017	0.017	0.008	0.007	**0.000**	**0.000**
Trace	0.010	0.020	0.020	**0.000**	**0.000**	**0.000**
TwoLeadECG	0.132	**0.004**	0.046	0.007	0.182	0.029
TwoPatterns	0.002	0.059	**0.001**	0.004	0.042	0.024
UWaveX	0.227	0.216	**0.164**	0.280	0.206	0.213
UWaveY	0.301	0.303	**0.249**	0.399	0.272	0.280
UWaveZ	0.322	0.273	**0.217**	0.321	0.265	0.267
Wafer	0.005	0.002	0.004	**0.001**	0.002	0.002
WordSyn	**0.252**	0.403	0.302	0.483	0.282	0.332
Yoga	0.155	0.195	0.149	0.182	**0.134**	0.175

In this section, we compare the performance of the proposed method with other approaches. The performance of 1-NN and SVM classifiers are reported excluding the other one. Hereinafter, we use CovNN and CovSVM for the union of the covariance representation with 1-NN and SVM classifiers, respectively. Firstly, as can be seen in Table 4.2, CovNN provides the best classification performance in 19 of 43 datasets. For 4 of these 19 datasets, the proposed method has identical classification performance with some of the compared methods. For the rest of the repository, the compared methods, DTW, ST, TSBF and HOG-1D+DTW-MDS provide the best classification performance for 4, 4, 6 and 15 of the datasets, respectively (including the ties). The scatter plots are provided in Figure 4.5 for pairwise evaluation of CovNN with the compared methods. CovNN individually outperforms DTW, ST, TSBF and

HOG1D+DTW-MDS in 34, 33, 34 and 22 datasets, respectively. The numbers of the datasets for which other methods have better results are 8, 9, 9 and 18 for the same order of the methods. CovNN has the best and same performance as DTW and HOG-1D+DTW-MDS for 1 and 3 of the datasets, respectively.

CovSVM provides the best classification performance in 20 of 43 datasets. For 4 of these 19 datasets, the proposed method has identical classification performance with some of the compared methods. For the rest of the repository, the compared methods, DTW, ST, TSBF and HOG-1D+DTW-MDS provide the best classification performance for 4, 4, 6 and 14 of the datasets, respectively (including the ties). The scatter plots are provided in Figure 4.6 for pairwise evaluation of CovSVM with the compared methods. CovSVM individually outperforms DTW, ST, TSBF and HOG1D+DTW-MDS in 35, 34, 35 and 25 datasets, respectively. The numbers of the datasets for which other methods have better results are 8, 8, 8 and 15 for the same order of the methods. CovSVM has the best and same performance as HOG-1D+DTW-MDS for 3 of the datasets.

Another critical performance parameter is the rank of the proposed approach among other methods. The rank is crucial in the sense that it indicates the consistency of the performance of a method among various results. For this purpose, we calculate the average rank of CovNN and CovSVM in a pool with other 4 methods. As mentioned before, the result of ST is missing for ECG200 dataset and therefore this dataset is excluded for this analysis. As can be seen in Figure 4.7, CovNN and CovSVM significantly outperform the compared four methods. The average ranks of the CovNN and CovSVM are 2.07 and 1.92, whereas the ranks of the nearest competitors are 2.52 and 2.59, respectively.

As can be derived from the comparison with the other methods, CovSVM has better performance than CovNN. A pairwise comparison between CovNN and CovSVM is performed on 43 datasets. The scatter plot for the comparison between CovNN and CovSVM is provided in Figure 4.8. The number of the datasets for which CovSVM has better results is 24, whereas CovNN is better for 12 datasets. For remaining 7 datasets, CovNN and CovSVM have equal classification accuracies. As a general observation, CovNN and CovSVM have similar performances. They perform well in

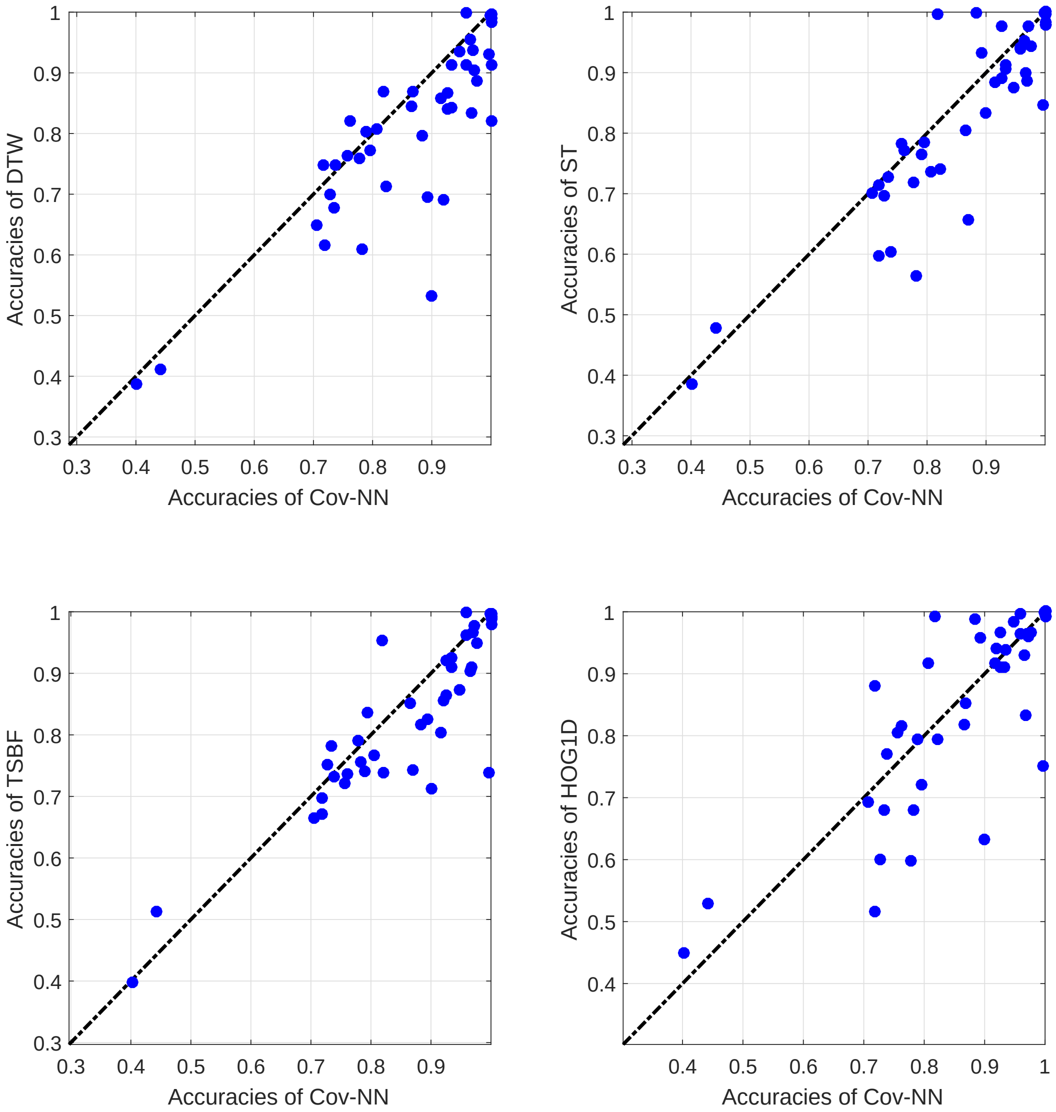


Figure 4.5: Comparison of CovNN with other methods on 43 datasets based on the results presented in Table 4.2. CovNN is better than DTW, ST, TSBF and HOG1D+DTW-MDS for 34, 33, 34 and 22 of the 43 datasets, respectively.

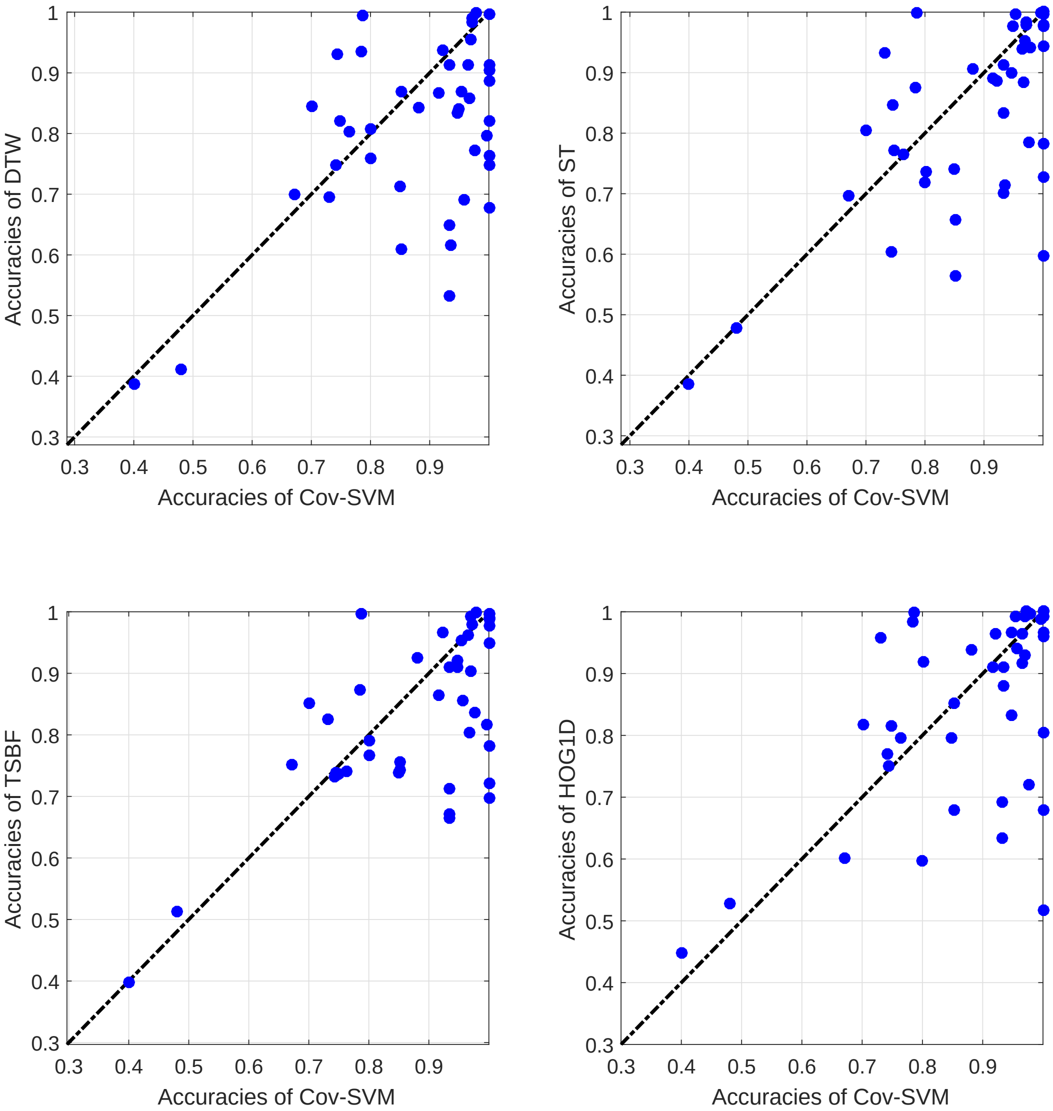
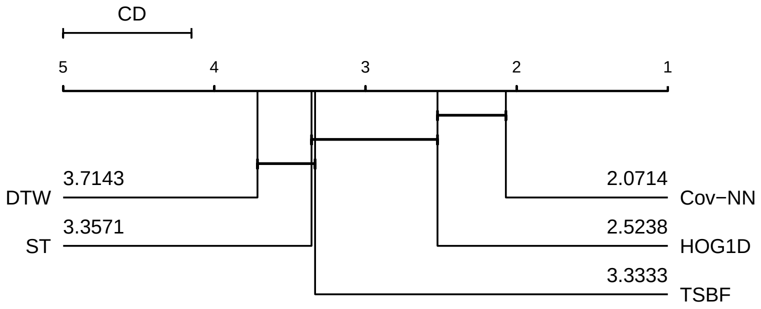
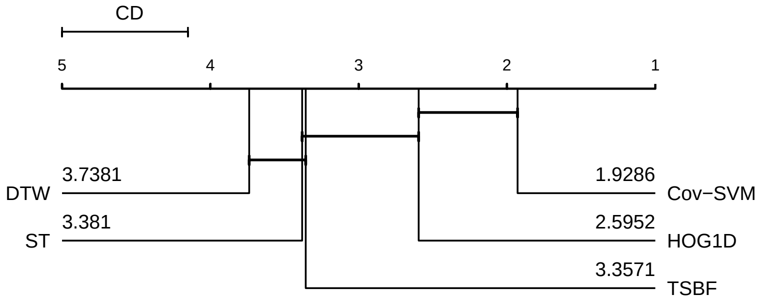


Figure 4.6: Comparison of CovSVM with other methods on 43 datasets based on the results presented in Table 4.2. CovSVM is better than DTW, ST, TSBF and HOG1D+DTW-MDS for 35, 34, 35 and 25 of the 43 datasets, respectively.



(a) Critical difference diagram for CovNN



(b) Critical difference diagram for CovSVM

Figure 4.7: Critical difference diagrams for the proposed approach with two classifiers and compared approaches.

the datasets where the covariance representation performs well.

When we conducted a more detailed analysis of the performance of the proposed method, it has been observed that the proposed method with both classifiers performs well except for the motion data. In the UCR repository, there are six different types of data as listed in Table 4.1. These types are image outline, motion, ECG, spectro, sensor reading and simulated. There are 11 motion datasets (*CricketX, CricketY, CricketZ, GunPoint, Haptics, InlineSkate, ToeSegmentation1, ToeSegmentation2, UWaveX, UWaveY, UWaveZ*). For 9 of them, except *ToeSegmentation1* and *ToeSegmentation2*, we have already presented the results in Table 4.2. CovNN and CovSVM have the best result for only *GunPoint*. For remaining 2 datasets, among the compared methods, only DTW and Shapelet Transform have the results. TSBF and HOG1D have not reported their results in these datasets. We obtained the error rates for these 2 datasets and reported in Table 4.3.

As can be seen from Table 4.3, CovNN and CovSVM do not have the best classification accuracies in these 2 datasets. When we analyze the motion datasets in

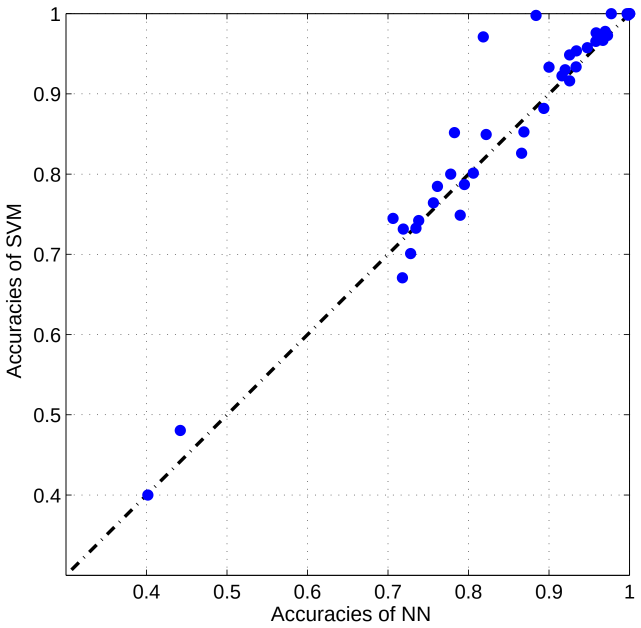


Figure 4.8: Comparison of NN and SVM classifiers on 43 datasets based on the results presented in Table 4.2.

UCR repository, we have observed that the motion data might be shifted in the time domain. In more detail, same portions of the motions can occur in different time intervals. These time intervals are generally close; however, in some circumstances, the proposed method cannot handle such situations.

To the best of our knowledge, in time series classification literature, there is only one study [9] which outperforms our results in UCR datasets. The work in [9] outperforms the proposed method in 24 of the 43 datasets. On the other hand, the proposed method outperforms [9] for 17 datasets and presents identical performance for 2 datasets. On the other hand, covariance representation with SVM classifier has better classification accuracies for 19 of 42 datasets. For 21 datasets, the method given in [3] outperforms the combination of covariance representation and SVM classifier. For 2 datasets, we have a tie. We want to remind that the classification accuracy for ECG200 dataset has not been reported in [3]. These results show that the proposed method gives classification accuracies that are comparable to a state-of-the-art ensemble method. It should be reminded that the main purpose of this work is to introduce a novel representation for time series and present its results with basic classifiers.

Table 4.3: Error rates for remaining motion datasets.

Dataset	**DTW**	**ST**	**CovNN**	**CovSVM**
ToeSeg1	0.250	0.044	0.1711	0.206
ToeSeg2	0.092	0.146	0.1231	0.146

4.5.2 Analysis on Pointwise Features

In the previous subsection, classification performances of CovNN and CovSVM are presented using the complete set of pointwise features as given in Eq. 4.4. In Eq. 4.4, we define a feature vector of six pointwise features. As already mentioned, the covariance descriptor is based on the second-order correlations between the pointwise features. Therefore, this feature vector represents the information of the relative changes within the time series. The value, derivative and time index are local features, and the rank, cumulative sum and the difference between the mean are global features. Using the local features, we emphasize on the characteristics of a point locally. Therefore, in the proposed approach, we consider the local features as the core features, by which the time series is principally represented as in our previous work [34]. In addition to the local features, we add the global (or auxiliary) features into the feature vector in this work. The global features are derived from the whole subsequence and give information about the prominence of the point within the subsequence. In the following, an analysis is performed to capture the performance of the proposed method with reference to a different combination of global (auxiliary) features by concatenating them to the complete set of local (core) features.

When overall accuracy in all of 43 datasets is considered, the feature vector that includes all pointwise features as defined in Eq. 4.4 gives the best results as listed in Table 4.5. However, the feature vector of six pointwise features does not give the best results for some datasets as can be seen in Figures 4.9 and 4.10. For 1-NN and SVM classifiers, better results are achieved in 15 and 14 of 43 datasets, respectively, when some of the global features are omitted. This result implies that the covariance representation may show better performance with an appropriate selection of pointwise features for other time series analysis problems. In CovNN case, when the best performing feature set is selected separately for each dataset, the average rank moves from 2.07 to 2. The number of datasets for which the proposed approach

Table 4.4: Analysis on global pointwise features. Average accuracies for seven combinations of pointwise features are listed. Abbreviations stand for LF: three Local Features together, CS: Cumulative Sum, DM: Difference between Mean.

	NN		**SVM**	
Features	**Average Accuracy**	**Standard Deviation**	**Average Accuracy**	**Standard Deviation**
LF+Rank	0.838	0.153	0.842	0.152
LF+CS	0.832	0.147	0.848	0.149
LF+DM	0.834	0.151	0.835	0.155
LF+Rank+CS	0.855	0.139	0.867	0.141
LF+Rank+DM	0.837	0.154	0.839	0.154
LF+CS+DM	0.830	0.146	0.850	0.146
LF+Rank+CS+DM	0.858	0.137	0.867	0.143

Table 4.5: Analysis on local pointwise features. Average accuracies for seven combinations of pointwise features are listed. GF stands for global features.

	NN		**SVM**	
Features	**Average Accuracy**	**Standard Deviation**	**Average Accuracy**	**Standard Deviation**
GF+Value	0.830	0.155	0.828	0.160
GF+Diff	0.848	0.148	0.852	0.146
GF+Time	0.839	0.149	0.842	0.152
GF+Value+Diff	0.848	0.147	0.857	0.144
GF+Value+Time	0.839	0.149	0.845	0.152
GF+Diff+Time	0.857	0.140	0.865	0.140
GF+Value+Diff+Time	0.858	0.137	0.871	0.140

has the best performance remains as 18. On the other side, for CovSVM, when the best performing feature set is selected separately for each dataset, the average rank moves from 1.93 to 1.86. The number of datasets for which the proposed approach has the best performance increases from 20 to 21. Moreover, for both classifiers, we observe a correlation between the number of features used and the stability of the performance when we observe the standard deviations of the classification accuracies (see the last column of Table 4.5). The minimum standard deviation is achieved when the complete set of (six) features are utilized.

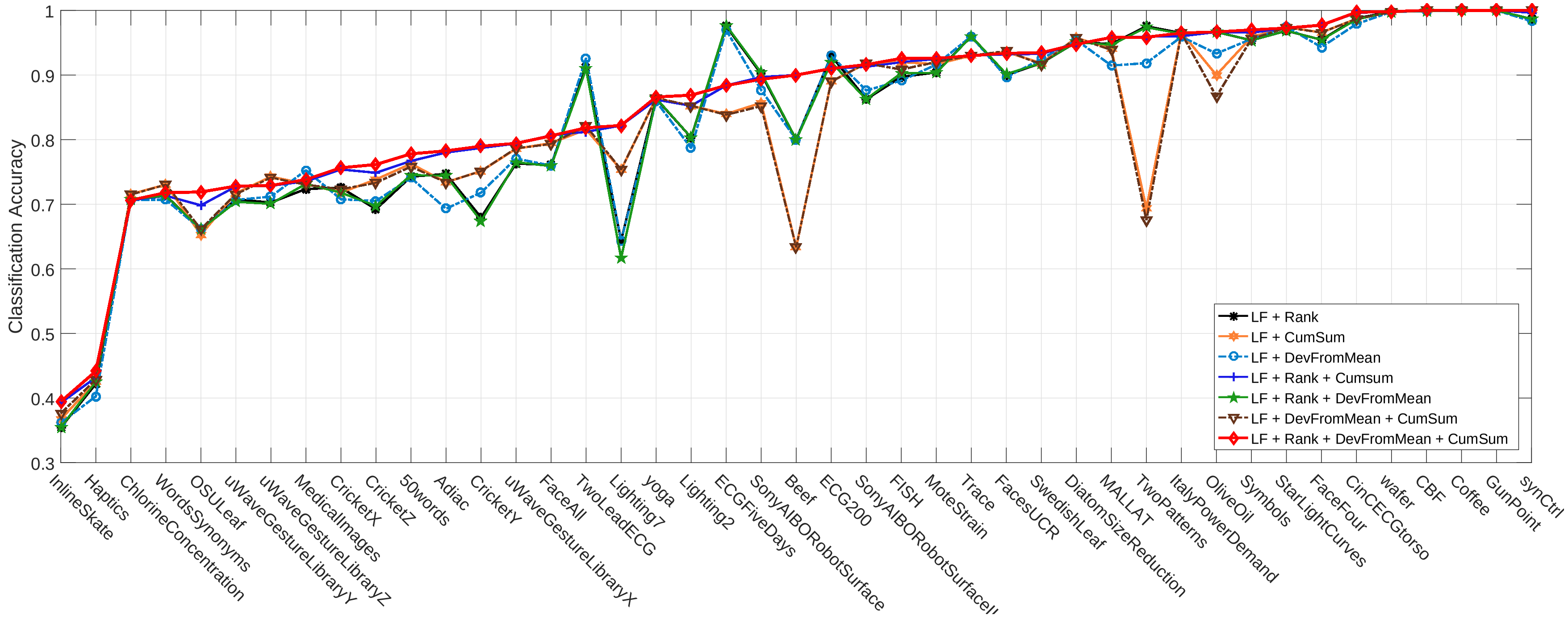


Figure 4.9: Accuracy values for different combinations of global pointwise features for 1-NN classifier. LF stands for local features, namely *value*, *derivative* and *time index*. The datasets are sorted according to accuracy values obtained in the case of six pointwise features.

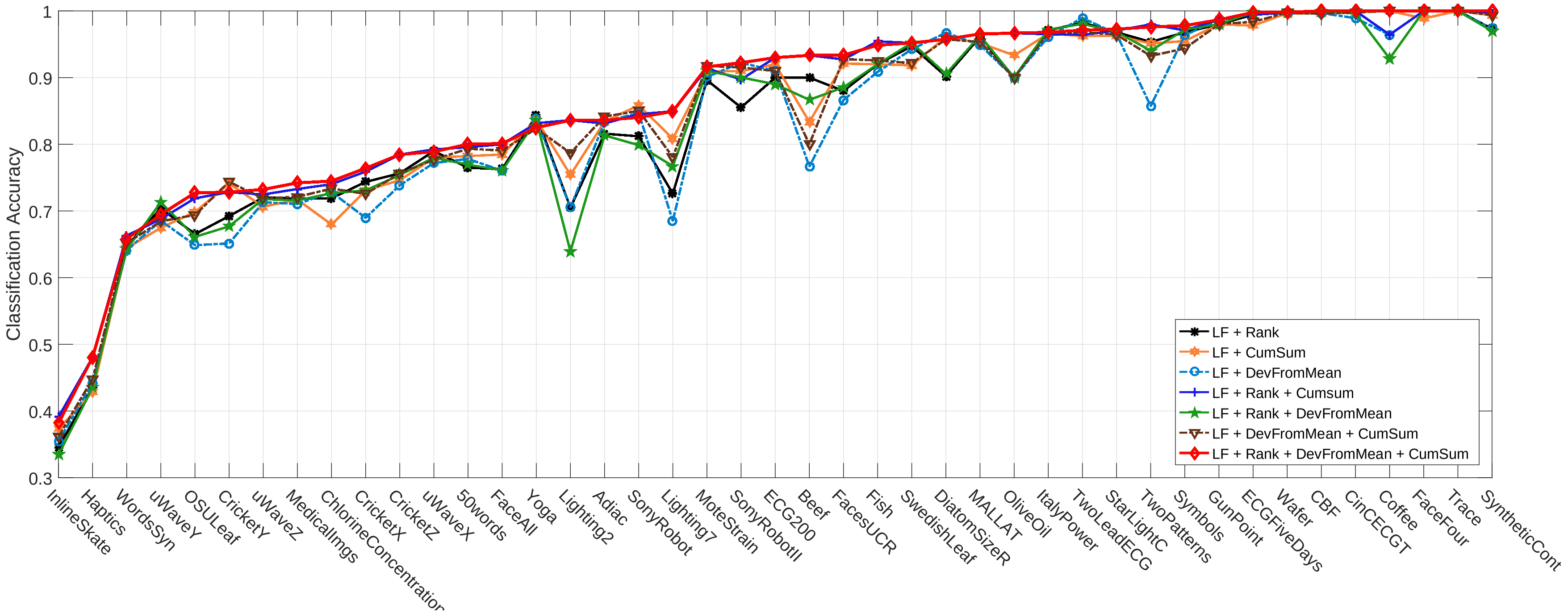


Figure 4.10: Accuracy values for different combinations of global pointwise features for SVM classifier. LF stands for local features, namely *value*, *derivative* and *time index*. The datasets are sorted according to accuracy values obtained in the case of six pointwise features.

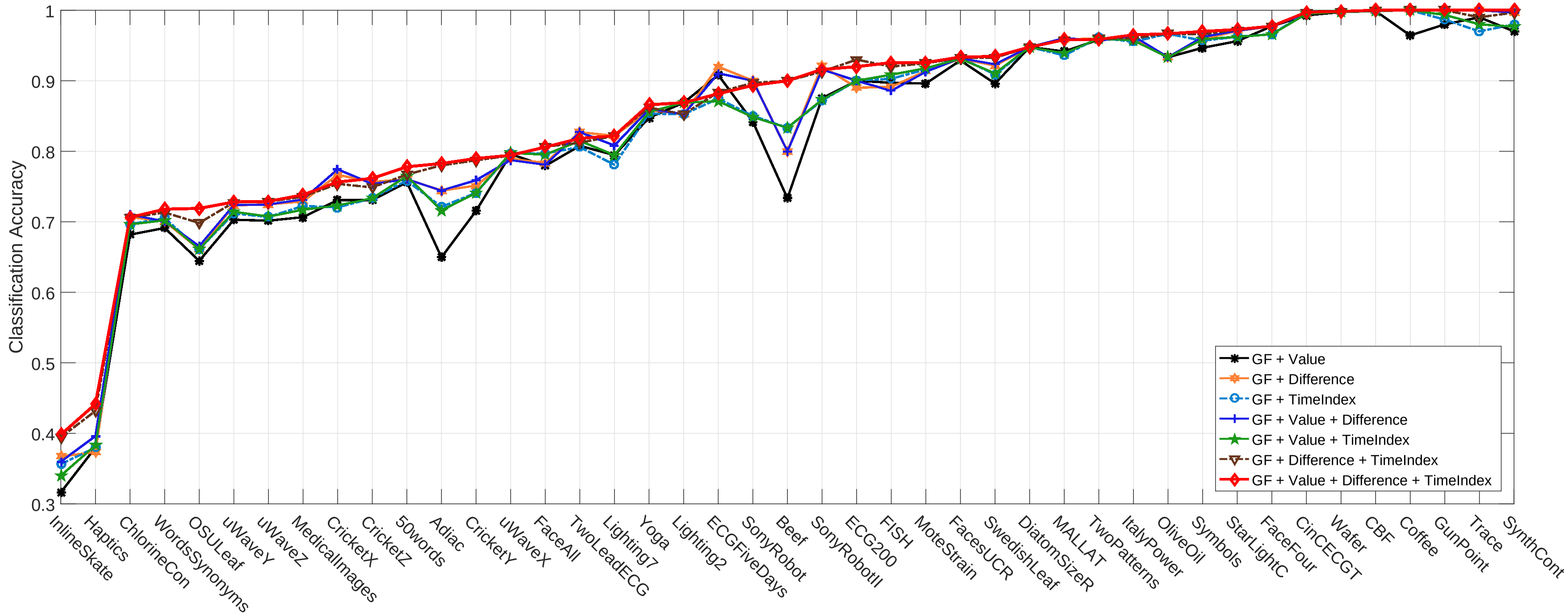


Figure 4.11: Accuracy values for different combinations of local pointwise features for NN classifier. GF stands for global features. The datasets are sorted according to accuracy values obtained in the case of six pointwise features.

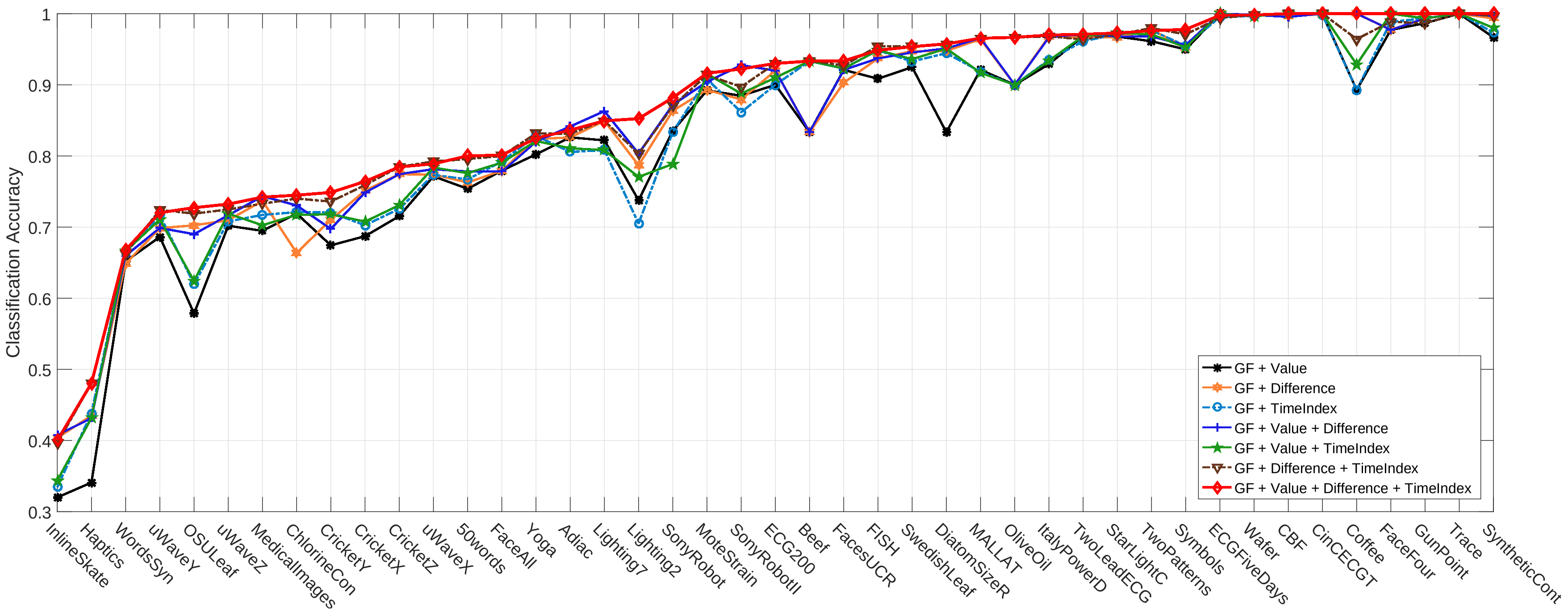


Figure 4.12: Accuracy values for different combinations of local pointwise features for SVM classifier. GF stands for global features. The datasets are sorted according to accuracy values obtained in the case of six pointwise features.

4.5.3 Analysis on Computational Complexity

In the first subsection, we compare the proposed method with well-known and state-of-the-art methods. It has been shown that the proposed method mostly outperforms the compared methods. In the previous subsection, we show that better results can be acquired with a different selection of pointwise features. In this subsection, we analyze the computational complexity of the CovNN and CovSVM and compare it with the HOG-1D+DTW-MDS method given in [114].The reasons why we choose of HOG-1D+DTW-MDS for comparison are that it is the closest competitor and is feature-based similarly to the proposed method. In Figure 4.13, the total computational times reported separately for each dataset are given for CovNN, CovSVM and HOG-1D+DTW-MDS. Since there are remarkable differences in total computational times between our method and compared method, the computational times are depicted using a log-axis. The reported computation times are obtained in MATLAB 2016a with a desktop machine with 4 cores, Intel i5-6500 CPU, 8GB RAM.

The reported values for HOG1D+DTW-MDS [114] are obtained by running the publicly available code[2]. The code does not include some steps such as determination of DTW-MDS features in the method. For this reason, the computational time for the *StarLightCurves* dataset is not reported. Therefore, the comparison for the computational times is based on 42 datasets. The reported computational times for our approach include all the calculations from start to finish for both classifiers.

As can be seen from Figure 4.13, CovNN finishes the classification within 10 seconds for most of the datasets. Moreover, for 12 datasets, it takes less than one second. Therefore, y-axis starts 10^{-1} in Figure 4.13. In the maximum case, the proposed method takes less than five minutes which occurs on *yoga* dataset. 1-NN classification takes 92.1 percent of the total calculation time on average. The average computational time for the proposed method is 46.6 seconds, while it is 4674 seconds for HOG1D+DTW-MDS. Consequently, it can be asserted that CovNN is 100 times faster than HOG1D+DTW-MDS technique. These results indicate that the proposed method is not only significantly accurate in classification, but also computationally efficient when compared to its competitor.

---

[2] https://github.com/jiapingz/TSClassification, lastly accessed in 04/09/2016

For CovSVM case, the computational times are still better than the work given in [114]. It is even better than CovNN in average. It begins to become faster when datasets become larger. This is due to the use of LIBSVM. Since LIBSVM is written in C++, it is normal to handle the larger datasets faster. The average time of CovSVM is about 6.98 seconds. When we examine the results more closely, we see that classification takes about %6.73 of the total time. Since we use MATLAB for 1-NN classifier, it takes so much time compared to the use of LIBSVM.

4.6 Conclusion

A novel approach is proposed for time series classification. Moreover, a novel distance calculation approach for time series is proposed by combining the covariance descriptor with a distance metric defined for covariance matrices. Conducted experiments on UCR time series datasets show that the proposed method yields results which are mostly outperforming some established methods such as DTW and shapelet transform and some state-of-the-art techniques such as TSBF and HOG-1D+DTW-MDS. Besides, in terms of computation time, the proposed method is very efficient when compared to a similar feature-based technique.

In addition to these satisfactory results with an efficient computation time, some challenging issues for time series such as missing data can also be addressed by using the covariance descriptor. Although the datasets in UCR repository do not include any missing data or time series of different lengths, these two issues can occur in a real time series classification problem. The covariance descriptor of the data with missing points will be again a square matrix with a length of number of features used. The missing points are inherently occupied in feature vectors since time index is used as a feature. Besides, the distance calculation between the time series of different lengths is generally problematic. In the proposed approach, invariant of the lengths of the time series, the dimensions of covariance matrices remain the same. Therefore, the proposed method is capable of measuring the distance between time series of different lengths.

A possible future work can be the comparison of the proposed method with standard

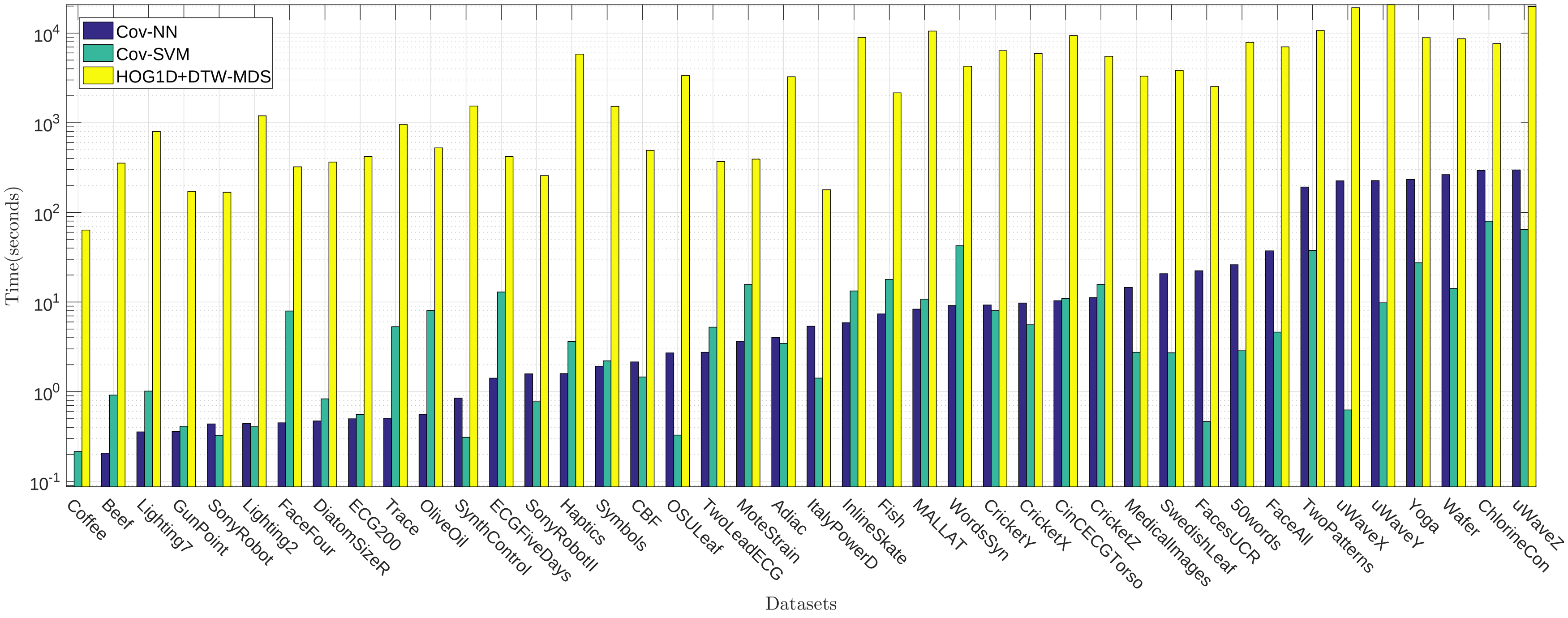


Figure 4.13: Comparison of the computational times of CovNN, CovSVM and HOG1D+DTW-MDS.

methods after creating a dataset which includes missing points and time series of different lengths. Moreover, the proposed method can be utilized in other time series analysis problems such as anomaly detection and clustering.

CHAPTER 5

SKELETAL ACTION RECOGNITION WITH RANK DISTANCE

In this chapter, we study the problem of action recognition using skeletal data extracted from depth cameras. The problem is treated as a multidimensional time series classification problem. As in previous chapters, we utilize feature covariance matrices to represent the time series data. Feature covariance matrices are previously used to solve the action recognition problem [49, 19, 104, 18].

Different from previous problems dealt with in previous chapters, the feature covariance matrices have relatively higher dimensions. This is because of the approach in the construction of the feature covariance matrix. This increase in the dimension leads to search a new distance measure for the feature covariance matrices. The novelty of this chapter comes at this point: we propose a new distance measure for feature covariance matrices and in general for SPD matrices. Several experiments have been conducted on state-of-the-art action recognition datasets. The use of the proposed distance measure gives outperforming results compared to other widely used distance measures [5, 85, 92].

5.1 Introduction

With the improvements on the depth camera technology, approaches developed for the depth cameras are getting more attention every day. There are two fundamental modalities of the data grabbed from depth cameras. The first one is RGB data as in standard cameras. The second one is the depth information. The human skeleton data is derived from these common modalities. Although it is a derived information, it is utilized to recognize human poses, gestures, motions and activities. For these applications, there are some approaches to represent 3D skeleton data. In a recent review paper [41], methods for the human representations based on 3D skeleton data are summarized. As shown in Figure 1 of the paper, the number of publications on skeletal human representation increases every year. Our observation is that the main reason for this increase is the action recognition problem.

5.2 Related Work

In this section, we give a brief overview of the approaches proposed for skeletal action recognition. For a review of the distance measures used for SPD matrices, we refer to Chapter 2.

It is better to start with methods that use feature covariance matrices as the representation of skeletal information. In [49], Hussein et al. build a hierarchy of covariance matrices on parts of the entire sequence. In [104], instead of calculating the covariance matrix directly, the inner product of the feature matrix is calculated after a passing a kernel function. The action recognition results are reported together with the results of other applications which utilizes covariance representation. In [19], a very similar approach is utilized, however, they decompose the covariance expression and use kernel trick for feature matrices. In [112], Gram and Hankel matrices are used to embed the joint trajectories a Riemannian manifold. There are also some recent works [46, 47] that aim to learn the nonlinear structure of the data and exploit the skeletal action recognition problem as a test bed.

The former methods are generally implementation of the classical methods in machine learning or computer vision for the skeletal action recognition problem. In [67], authors build a graph to model dynamics of the actions and use a bag of 3D points for representing the features. Spatiotemporal features are fused in [118] by using random forests. Similar to HOG features, in [109], the histogram of 3D locations of the joints are assigned to the related bins. An HMM is used as the classification method.

LSTM based approaches [71, 117, 72] are arising with the increase of the size of the data. These algorithms need the large datasets. Although there are some datasets that can be called as large size like HDM05 [82], in [91], a very large-scale RGB+D action recognition dataset introduced. The authors also introduce a form of LSTM called part-aware LSTM for action recognition.

5.3 Representation of Skeletal Data with Feature Covariance Matrices

In the skeletal action recognition problem, the skeletons have J joints and the action sequence has N frames. If we use 3D coordinates of the joints there will be a 3JxN feature matrix. Such a feature matrix is given in Equation 5.1. Other derived point-wise features such as 3D velocities as mentioned in previous chapters can be inserted into the feature matrix. Most of the results are achieved by using a feature matrix of 6JxN. For such cases, the covariance matrix will be of order 6Jx6J. Such an insertion increases the dimension of covariance matrices and makes the classical distance measures less usable.

$$F = \begin{bmatrix} x_1(t=1) & x_1(t=2) & \dots & x_1(t=N) \\ y_1(t=1) & y_1(t=2) & \dots & y_1(t=N) \\ z_1(t=1) & z_1(t=2) & \dots & z_1(t=N) \\ x_2(t=1) & x_2(t=2) & \dots & x_2(t=N) \\ y_2(t=1) & y_2(t=2) & \dots & y_2(t=N) \\ z_2(t=1) & z_2(t=2) & \dots & z_2(t=N) \\ \vdots & \vdots & \dots & \vdots \\ x_J(t=1) & x_J(t=2) & \dots & x_J(t=N) \\ y_J(t=1) & y_J(t=2) & \dots & y_J(t=N) \\ z_J(t=1) & z_J(t=2) & \dots & z_J(t=N) \end{bmatrix} \tag{5.1}$$

Feature covariance matrix is determined as in Equation 5.2

$$C = \frac{1}{K} \sum_{k=1}^{K} (F_k - \mu)(F_k - \mu)^T \tag{5.2}$$

After that, we obtain the distances between the samples of training and test sets. The classification is done through the distance matrices. At this point, as in previous

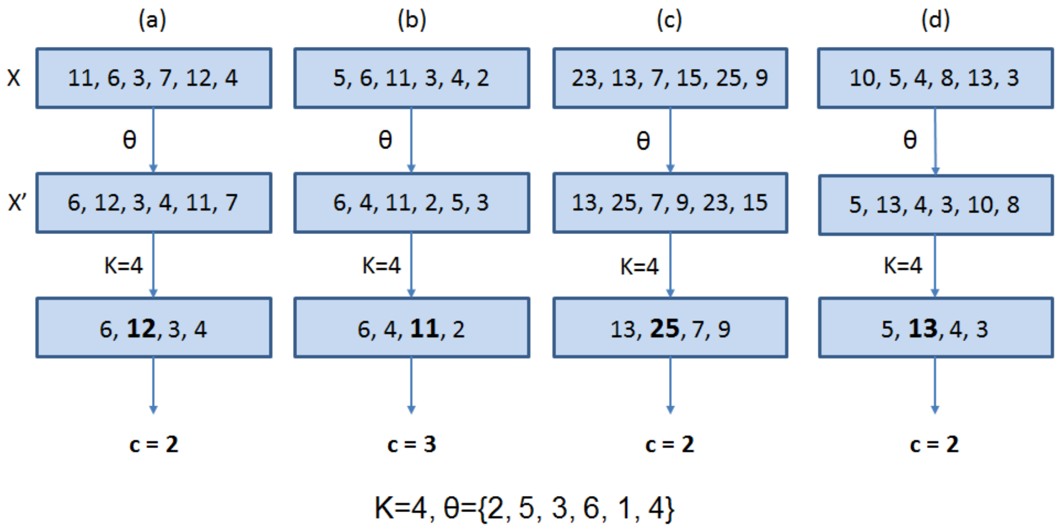


Figure 5.1: An example of WTA hashing with 6-dimensional input vectors, K = 4, and $\theta = (1, 4, 2, 5, 0, 3)$.

chapters, we firstly utilize kNN classifier to evaluate the performance of the distance measure. Then, we use SVM classifier as in previous approaches [104, 19].

5.4 Action Recognition with Rank Distance Measure

As detailed in the previous section, the feature covariance matrices become high dimensional for action recognition problem. Previous works utilize the classical distance measures for measuring the distance between two covariance matrices. As the dimension of the covariance matrices increase, the modeling of the manifold structure will become an issue. For this purpose, some recent works try to learn the manifold structure [31, 46], and some other works handle the problem as a distance learning problem [48]. We follow a different way to measure the distance between two high-dimensional covariance matrices.

Rank metric is used to measure distances between high dimensional data. It gives very promising results in [28, 53] where it is used to calculate the distances between feature descriptors. Its computational speed is also satisfactory since it uses hashing mechanism. The authors named the hash functions they utilize as “Winner Take All” (WTA) since it takes the index of the maximum of the permuted values.

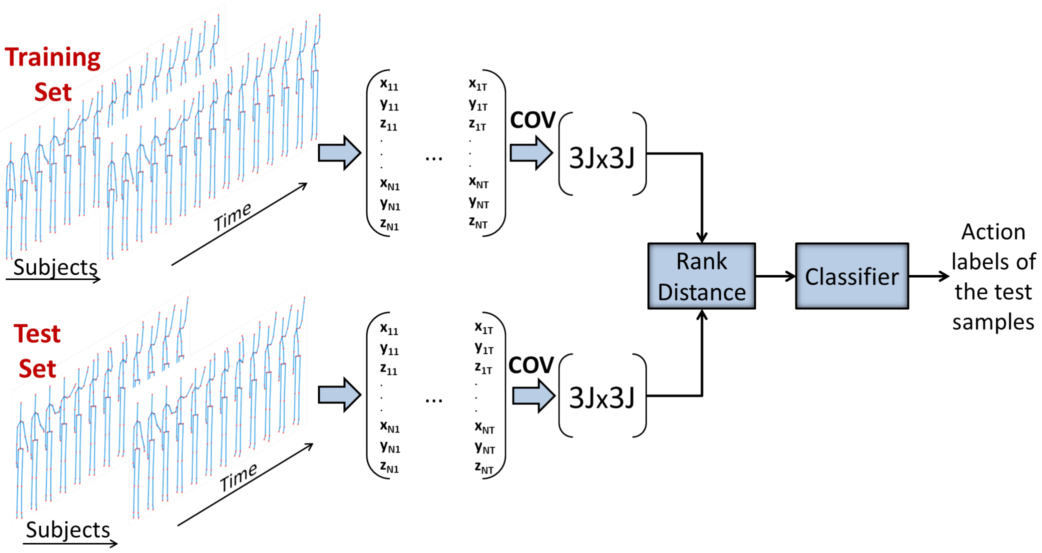

Training Set
Subjects
Time
x11 y11 z11 ... xN1 yN1 zN1
x1T y1T z1T ... xNT yNT zNT
COV
3Jx3J
Test Set
Subjects
Time
x11 y11 z11 ... xN1 yN1 zN1
x1T y1T z1T ... xNT yNT zNT
COV
3Jx3J
Rank Distance
Classifier
Action labels of the test samples

Figure 5.2: General flow diagram of the skeletal action recognition approach.

In order to determine the distances between covariance matrices via rank-distance measure, we use the vectorized form of the upper (or lower) diagonal part of covariance matrices. Using same permutations for all covariance matrices, we construct the codes shown in Figure 5.1 by following the algorithm given in Algorithm 1. In [53], there is also another version of the WTA hash on polynomial kernels. It does not give better results compared to the original WTA hash given in Algorithm 1.

```
Input: A set of m permutations $\Theta$, window size K, vectorized form of upper
       (or lower) triangular part of covariance matrix, $X$.
Output: Sparse code
1. For each permutation $\theta_i$ in $\Theta$
    a. Permute elements of $C$ according to $\theta_i$ to get $C^{'}$.
    b. Initialize $i^{th}$ sparse code $c_{x_i}$ to 0.
    c. Set $c_{x_i}$ to the index of the maximum value in $X^{'}(1, \dots, K)$.
        i. For $j = 0$ to $K - 1$
            A. If $X^{'}(j) > X^{'}(c_{x_i})$ then $c_{x_i} = j$
2. $C_X = [c_{x0}, c_{x1}, \dots, c_{x_{m-1}}]$, $C$ contains $m$ codes, each taking a value between
   0 and $K - 1$.
```

Algorithm 1: WTA hashing of covariance matrices

5.5 Experiments

In order to test the rank-based distance measure, we first compare it with well-known distance measures using a synthetic dataset. The synthetic dataset is generated by exploiting the software supplemented in [42]. Two datasets [67, 103] are utilized to evaluate the performance of the rank-based distance measure on the action recognition problem. Before going into details of the results, it is better to give a brief information about the datasets.

Synthetic Data: Synthetic data is generated in such a way that it includes 4 classes. The number of training and testing samples are selected as 50 for each class. We create synthetic SPD matrices of dimensions of 10, 20, 30, 40, 50, 100 and 200 in order to compare the distance measure with the classical distance measures.

MSR-DailyActivity3D: MSR-DailyActivity3D dataset is generated using Microsoft Kinect. The dataset contains 320 sequences of 16 action classes. In each class, there are 20 sequences. The number of joints is 20. For this dataset, usually, the performances are reported using the velocities. Therefore, the dimension of the feature covariance matrices is 120x120. For the separation of the training and test sets, a cross-subject setting is used as in previous works [104, 101]. In the cross-subject setting, odd-indexed subjects are selected for training whereas even-indexed subjects are used for the testing.

MSR-Action3D: As MSR-DailyActivity3D, MSR-Action3D dataset is generated from depth sequences grabbed by Microsoft Kinect. The dataset contains 567 sequences of 20 action types, 10 people (subjects) and each subject performs each action 2 or 3 times. However, as in [101], 23 sequences are removed from the dataset due to missing data they contain. For this dataset, The number of joints is 20 and usually, the performances are reported using the velocities. Therefore, the dimension of the feature covariance matrices is 120x120. As in MSR-DailyActivity3D, the separation of the training and testing sets is based on the index of the subjects.

We use kNN and SVM classifiers in action recognition problem. Since the number of training samples is not adequate in MSR-DailyActivity3D, we use a 1-NN classifier. On the other hand, for MSR-Action3D, we have more training samples, we use kNN classifier with $k = 3$. The main reason behind the use of kNN classifier is to isolate the performance of the classifier. In other words, we only measure the performance of the distance measure as far as we can. A nonlinear SVM is used as in [104, 19] to compare the rank-based distance measure with some works that propose different scheme for calculating the covariance matrices or covariance-like matrices.

The only paramater that should be optimized in our tests is the window size of the maximum operation. We perform a cross-validation for the action recogntion datasets and set the value of the window size parameter as K=4. All the reported values for classification accuracies are achieved by setting the number of permutations as 10000. We give an analysis about how the number of permutations effects the classification accuracy in Section 5.5.3. The other parameters such as used in SVM classifier are kept fixed. Lastly, we should note that all the results obtained using the rank-based distance are determined by using Monte-carlo simulations since the permutations are random.

5.5.1 Comparison with Other Distance Measures

In order to calculate the rank-based distance measure with other distance measures, we measure the classification performances on datasets. The synthetic dataset and two action datasets are exploited for this purpose. kNN classifier is used in all the experiments after distances obtained between training and test sets.

Table 5.1: The comparison of the rank based distance measure with well-known distance measures on a synthetic data.

Dimension	**AIRM**	**LogEuc**	**Rank**	**Stein**
10	1	1	0.95	1
20	0.815	0.78	0.92	0.8
30	0.635	0.645	0.96	0.62
40	0.49	0.475	0.92	0.51
50	0.42	0.44	0.945	0.385
100	0.330	0.360	0.830	0.355
200	0.290	0.310	0.550	0.250

We firstly exploit the synthetic dataset for comparing the rank-based distance with the classical distance measures. The dimensions of the created datasets are from

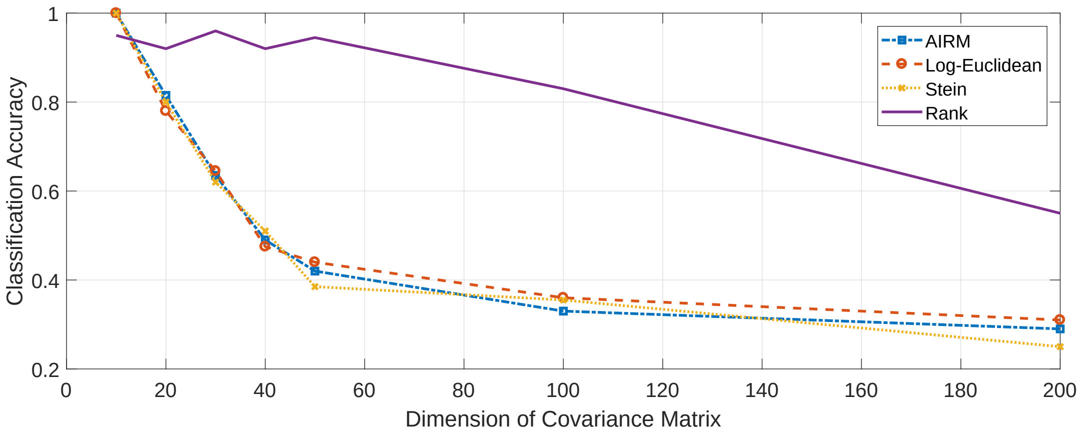


Figure 5.3: Analysis on how classification accuracies change with respect to the dimension of covariance matrices.

10 to 200 as can be seen in Table 5.1 and Figure 5.3. For the synthetic datasets of different dimensions, we report the classification accuracies of the 1-NN classifier fed by the distances. When the dimension of the covariance matrices is low, the classical distance measures give better results compared to the rank-based distance measure. When the dimension of the data increases, the advantage of the rank-based distance measure arises. the difference between the rank-based distance and classical measures become dramatic when the dimension of the covariance matrices is getting higher.

After the synthetic dataset, a similar comparison is realized on the real skeletal action recognition datasets. For MSR-DailyActivity3D dataset, we utilize 1-NN classifier due to the inadequate number of training samples per class. For MSR-Action3D, kNN classifier with $k = 3$ is utilized. In real datasets, the results are given in Table 5.2. The first observation is that the performances of AIRM and Stein metrics are very poor. The closest competitor is log-Euclidean distance measure. This result is consistent with the results obtained in previous chapters.

Table 5.2: Classification accuracies of different distance measures using the kNN classifier on MSR-Action3D and MSR-DailyActivity3D datasets.

Distance Measure	**MSR-Action3D**	**MSR-DailyActivity3D**
AIRM	0.677	0.806
LogEuc	0.808	0.844
Rank	0.933	0.914
Stein	0.600	0.575

Table 5.3: Comparison of the action recognition results with the techniques based on covariance matrix and other state-of-the-art approaches.

Method	**MSR-Action3D**	**MSR-DailyActivity3D**
Cov-LogEuc	0.923	0.856
Cov-J_H-SVM [74]	0.804	0.755
KerCov [19]	0.962	0.963
Ker-RP-POL [104]	0.962	**0.969**
Ker-RP-RBF [104]	**0.969**	0.963
Cov-Rank-NN	0.933	0.914
Cov-Rank-SVM	0.960	0.949

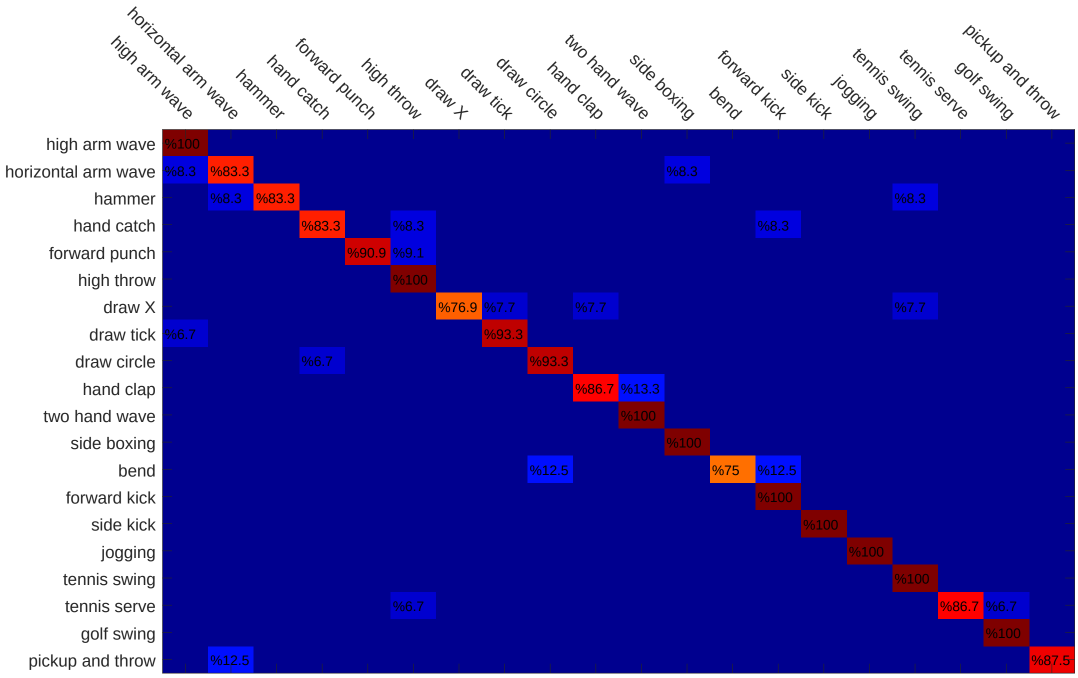
5.5.2 Action Recognition Results

In this section, the rank-based distance measure is tested on skeletal action recognition datasets by using a nonlinear SVM classifier as in previous works [104, 19]. We also report the results obtained with log-Euclidean distance measure. As can be seen from Table 5.3, the rank-based distance measure provides comparable results in two datasets. From the table, we can also observe that the advantage of SVM classifier. For SVM classifier, Libsvm toolbox is not used. Instead of Libsvm, the toolbox given in [3] is used. The results obtained with this toolbox are slightly better than the results obtained with Libsvm.

For action recognition, we use the scheme given in Figure 5.2. In Table 5.3, we report the results of both classifiers. We also report the results of the combination of the log-Euclidean distance measure and SVM classifier. The work in [74] follows a similar approach. It uses the combination of Bregman divergence and SVM classifier. Lastly, we obtain again better results compared to the results obtained with the log-Euclidean distance measure. The confusion matrices for MSR-Action3D and MSR-DailyActivity3D datasets are shown in Figure 5.4 and Figure 5.5, respectively.

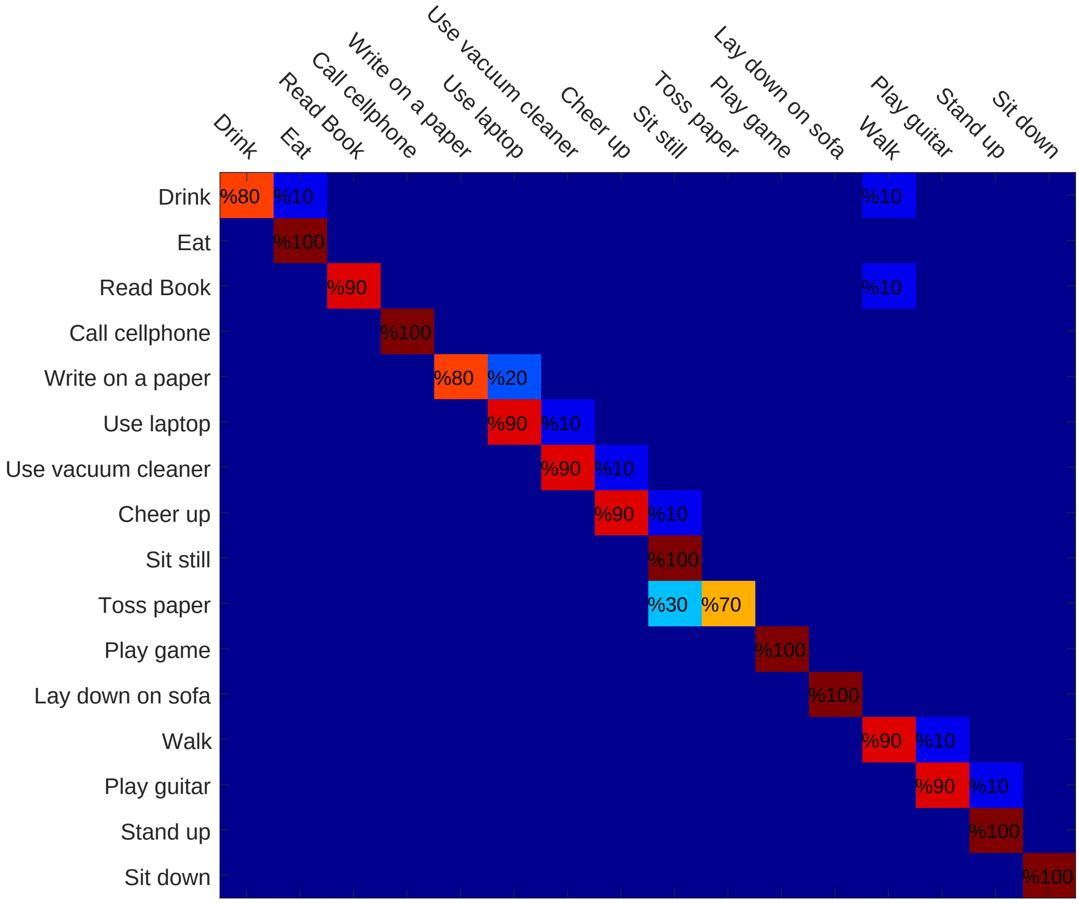
5.5.3 Analysis on Number of Permutations

We have analyzed the effect of the number of permutations for classification accuracy. For this purpose, on the MSR-Action3D and MSR-DailyActivity3D datasets, classi-



	high arm wave	horizontal arm wave	hammer	hand catch	forward punch	high throw	draw X	draw tick	draw circle	hand clap	two hand wave	side boxing	bend	forward kick	side kick	jogging	tennis swing	tennis serve	golf swing	pickup and throw
high arm wave	%100																			
horizontal arm wave	%8.3	%83.3										%8.3								
hammer		%8.3	%83.3														%8.3			
hand catch				%83.3		%8.3								%8.3						
forward punch					%90.9	%9.1														
high throw						%100														
draw X							%76.9	%7.7		%7.7							%7.7			
draw tick	%6.7							%93.3												
draw circle				%6.7					%93.3											
hand clap										%86.7	%13.3									
two hand wave											%100									
side boxing												%100								
bend									%12.5				%75	%12.5						
forward kick														%100						
side kick															%100					
jogging																%100				
tennis swing																	%100			
tennis serve						%6.7												%86.7	%6.7	
golf swing																			%100	
pickup and throw		%12.5																		%87.5

Figure 5.4: Confusion matrix for MSR-Action3D dataset.



	Drink	Eat	Read Book	Call cellphone	Write on a paper	Use laptop	Use vacuum cleaner	Cheer up	Sit still	Toss paper	Play game	Lay down on sofa	Walk	Play guitar	Stand up	Sit down
Drink	%80	%10											%10			
Eat		%100														
Read Book			%90										%10			
Call cellphone				%100												
Write on a paper					%80	%20										
Use laptop						%90	%10									
Use vacuum cleaner							%90	%10								
Cheer up								%90	%10							
Sit still									%100							
Toss paper									%30	%70						
Play game											%100					
Lay down on sofa												%100				
Walk													%90	%10		
Play guitar														%90	%10	
Stand up															%100	
Sit down																%100

Figure 5.5: Confusion matrix for MSR-DailyActivity3D dataset.

Table 5.4: Classification accuracies with respect to the number of permutations for the MSR-Action3D and MSR-DailyActivity3D datasets using kNN classifier.

Number of Permutations	MSR-Action3D	MSR-DailyActivity3D
10	0.179	0.560
20	0.392	0.674
50	0.587	0.739
100	0.771	0.816
200	0.832	0.844
500	0.896	0.879
1000	0.922	0.891
2000	0.932	0.896
5000	0.932	0.904
10000	0.939	0.914

fication accuracies are observed for several number of permutations. The results are given in Table 5.4 and shown in Figure 5.6 and Figure 5.7. As a benchmark, we show the results of the log-Euclidean and other distance measures on the figures. As can be seen from the table and figures, the rank-distance measure outperforms the log-Euclidean distance measure using only 200 permutations. When compared to Stein metric and AIRM, the rank-based distance provides better classification accuracies using only 100 permutations.

Another observation for the rank-based distance is that 10000 permutations are enough to reach steady state. While obtaining the results, as mentioned before, the value of window size of the maximum operation,K, is set to 4. In other words, it is enough to take 10000 permutations from infinite permutations for a feature vector which has a dimension of 120.

5.5.4 Analysis on Computational Complexity

Since the rank-based distance measure utilizes hash functions, its computational complexity is very low compared to the classical distance measures. In this section, we give a comparison of the computational times of classical distance measures and rank-based distance measure on the MSR-Action3D dataset. The computation times are obtained in MATLAB2017a and in a machine that has 8GB memory. Just as a re-

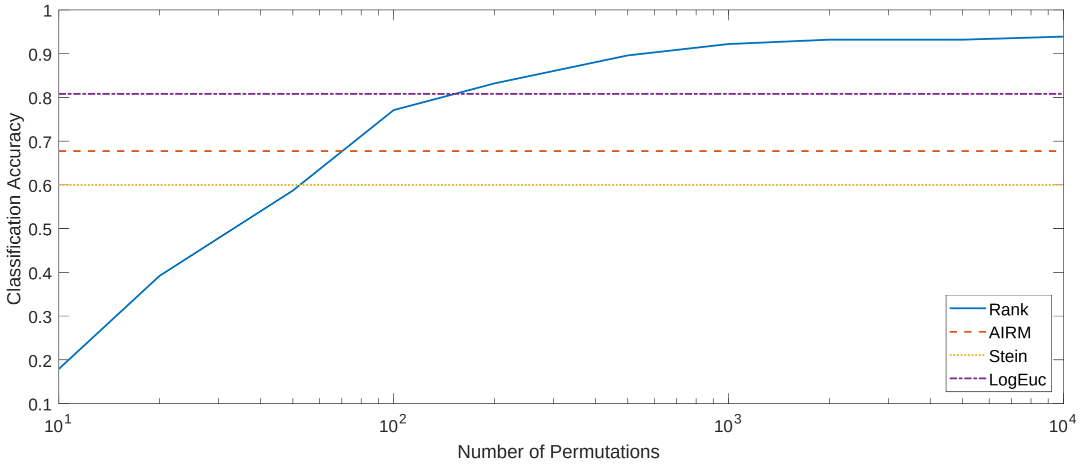


Figure 5.6: Analysis of the effect of number of permutations on the classification accuracy for MSR-Action3D dataset.

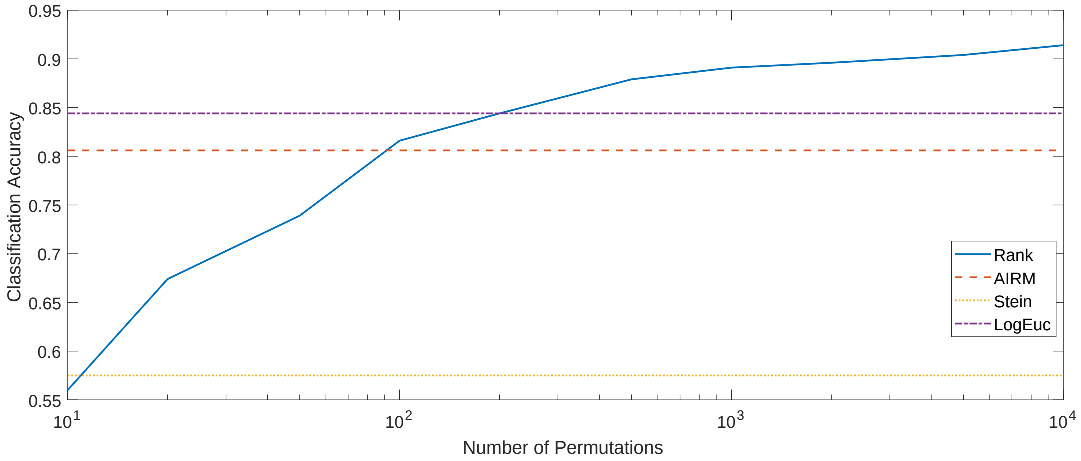


Figure 5.7: Analysis of the effect of number of permutations on the classification accuracy for MSR-DailyActivity3D dataset.

Table 5.5: Computation times of distances between training and test sets for different distance measures on MSR-Action3D dataset.

Distance Measure	Computation Time (seconds)
AIRM	94.39
LogEuc	7.1
Rank	5.2
Stein	21.38

minder, MSR-Action3D dataset consists of 284 training and 260 test samples after eliminating some noisy samples.

The rank-based distance is also computationally efficient as can be seen from Table 5.5. The performance of the distance measures in computation time is similar to the performance in the classification. The only change in the ranking is between Stein and AIRM. Since AIRM calculates the generalized eigenvalues between covariance matrices each time, it becomes computationally inefficient. It can be assumed that the computation time of the rank distance is only dependent on permutation number since the actual distance calculation between the codes is achieved through the Hamming distance.

5.6 Conclusion

In this chapter, we study the problem of action recognition on skeletal data. The novel part of the chapter is the utilization of the rank-based distance measure for covariance matrices. The performance of the proposed measure is validated with a comparison of other distance measures on a synthetic dataset and state-of-the-art skeletal action recognition datasets. It can be claimed that the rank-based distance measure enables to learn the manifold structure of the data. Therefore, there are problems that can be attacked by using the proposed distance measure.

The only disadvantage of the rank-based distance measure is that its results are not deterministic. This is due to usage of random permutations. This can be handled by the learning of the favorable permutations from the training set during cross-validation. We can take the permutations that it provides higher classification accuracies during cross-validation as favorable permutations.

During the analysis of the rank-based distance measure, we also realize a benchmark study for other distance measures used for SPD matrices in this chapter. Among three metrics, log-Euclidean distance measure has the best performance for high dimensional SPD matrices. It was the distance measure that we utilized in previous chapters. It is also observed that the log-Euclidean distance measure is also the faster than the other distance measures, namely, Stein and affine-invariant Riemannian metric. This is due to the fact that its formula that does not contain any terms two covariance matrices together. It makes possible to calculate the matrix logarithms separately for training and test samples. The Frobenius norm is another factor for the speed of the log-Euclidean metric. However, it is not faster than the rank-based distance measure.

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this thesis, we mainly address the problems related to time series. The starting point of the thesis is the analysis of crowded scenes. For this purpose, we propose a novel representation for trajectories. The representation is based on feature covariance matrices of pointwise features. By this representation, we carry the time series into a Riemannian manifold space. We utilize the representation for solving the problems of anomaly detection and activity perception. For 1D time series, we attack the problem of time series classification. At this time, instead of representing the whole time series with a single covariance matrix, time series is divided into overlapping subsequences. Lastly, we apply feature covariance matrices for whole 3D time series as in previous chapters. In this part of the study, we propose a novel distance metric which is based on the rank measures of covariance matrices.

6.1 Conclusion

A novel approach is proposed by describing time series with feature covariance matrices. 2D time series or trajectories is the first application of this representation. For each point of the trajectory, simple pointwise features such as position and velocity are utilized to describe the whole trajectory. After representing trajectories, the problems of anomaly detection and activity perception are investigated to measure the strength of the representation. Conducted experiments show that covariance descriptor for trajectories yields satisfactory results compared to the state of the art. We have also introduced a sparse anomaly detector to decide the number and the weights of the nearest neighbors that should be used. This sparse representation can be applied to other similar problems. The only requirement is to have a training dataset for which annotated anomaly data is given.

The proposed representation is adapted to solve the classification of 1D time series. Feature covariance matrices are calculated for the subsequences and new pointwise features are defined for 1D time series. Conducted experiments on UCR time series datasets show that the proposed method yields results which are mostly outperforming some established methods such as DTW and shapelet transform and some state-of-the-art techniques such as TSBF and HOG-1D+DTW-MDS. Besides, in terms of computation time, the proposed method is very efficient when compared to a similar feature-based technique. In addition to these satisfactory results with an efficient computation time, some challenging issues for time series such as missing data can also be addressed by using the covariance representation. Although the datasets in UCR repository do not include any missing data or time series of different lengths, these two issues can occur in a real time series classification problem. The covariance representation of the data with missing points will be again a square matrix with a length of number of features used. The missing points are inherently occupied in feature vectors since time index is used as a feature. Besides, the distance calculation between the time series of different lengths is generally problematic. In the proposed approach, invariant of the lengths of the time series, the dimensions of covariance matrices remain the same. Therefore, the proposed method is capable of measuring the distance between time series of different lengths.

Rank-based distance measure gives promising results on skeletal action recogntion problem. It is compared with the classical distance metrics used for SPD matrices. It outperforms the classical distance measures when the dimension of the covariance matrices are high. Therefore, it can be applied on several problems that use covariance matrices or SPD matrices. On the other hand, the rank-based distance measure is not just a distance measure. It encodes the SPD matrices by using a hashing mechanism. It can be used as a manifold learning approach and can be applied into several problems. Also, it can be used as a dimensionality reduction approach on manifolds.

6.2 Future Work

Rank-based distance measure gives promising results in determining the similarity of two high-dimensional covariance matrices. In the rank-based distance measure, instead of random permutations, we can also use a learning mechanism. By such an approach, we learn the discriminative parts of the covariance matrices and model the manifold. There are also other applications such as DTI imaging [32], image set classification [48] and face recognition [42] like skeleton based action recognition problem.

Covariance matrix representation is a proven approach for action recognition problem. It gives satisfactory results with some modifications in several works [49, 19, 104]. On the other hand, RNN-based or more specifically LSTM-based methods also give satisfactory results. The combination of these two approaches could be a good future work. In more detail, instead of feeding 3D coordinates into LSTM networks directly such as in [71], covariance matrices can be fed into LSTM networks. Also, different RNN or more specifically LSTM architectures [83, 62] can be utilized for such a future work. The benchmark of these architectures is another candidate for a future work.

Taking into account the popularity of deep learning in recent years, RNN or LSTM based approaches can be applied to the problem of time series classification. However, for such a purpose, the datasets used in the data mining community are not too large. Therefore, creating such a large dataset must be the first step of deep learning based approaches. Another possible future work can be a benchmark study of the time series classification approaches on noisy and missing data. The datasets that will be used for the benchmark study should also contain the time series of different lengths. Moreover, the proposed method can be utilized in other time series analysis problems such as anomaly detection and clustering.

All the approaches developed in this study can be considered as offline. In other words, we need to have some batch data to determine the feature covariance matrices. At next time step, to update the feature covariance matrices, we should process all the data except the earliest one. An updating mechanism for feature covariance matrices could have several applications including the problems investigated in this study.

REFERENCES

[1] www.timeseriesclassification.com, time series classification website, lastly accessed on march 2017.

[2] Libsvm library, April 2017.

[3] Svm and kernel methods matlab toolbox, http://asi.insa-rouen.fr/enseignants/ arakoto/toolbox/, November 2017.

[4] N. Ahmed, A. F. Atiya, N.E. Gayar, and H. El-Shishiny. An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, 29(5-6):594–621, 2010.

[5] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Log-euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic resonance in medicine 56, no. 2 (2006): 411-421*, 2006.

[6] J. Aßfalg, H.P.Kriegel, P. Kröger, P. Kunath, A. Pryakhin, and M. Renz. Similarity search on time series based on threshold queries. In *Proceedings of the 10th International Conference on Extending Database Technology*, pages 276–294. Springer, 2006.

[7] S. Atev, O. Masoud, and N. Papanikolopoulos. Learning traffic patterns at intersections by spectral clustering of motion trajectories. In *IEEE Conf. Intell. Robots and Systems*, 2006.

[8] M. El Ayadi, M.S. Kamel, and F. Karray. Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, 44(3):572–587, 2011.

[9] A. Bagnall, J. Lines, J. Hills, and A. Bostrom. Time-series classification with cote: The collective of transformation-based ensembles. *IEEE Transactions on Knowledge and Data Engineering*, 27(9):2522–2535, 2015.

[10] A. Bailly, S. Malinowski, R. Tavenard, T. Guyet, and L. Chapel. Bag-of-temporal-sift-words for time series classification. In *ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data.*, 2015.

[11] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten. Classification of covariance matrices using a riemannian-based kernel for bci applications. *Neurocomputing*, 112:172–178, 2013.

[12] F. I. Bashir, A. A. Khokhar, and D. Schonfeld. Object trajectory-based activity classification and recognition using hidden markov models. *IEEE Transactions on Image Processing*, July 2007.

[13] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *European Conference on Computer Vision*, pages 404–417, 2006.

[14] M. G. Baydogan, G. Runger, and E. Tuv. A bag-of-features framework to classify time series. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2796–2802, 2013.

[15] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[16] P. Bilinski and F. Bremond. Video covariance matrix logarithm for human action recognition in videos. In *IJCAI 2015-24th International Joint Conference on Artificial Intelligence*, 2015.

[17] D. Buzan, S. Sclaroff, and G. Kollios. Extraction and clustering of motion trajectories in video. In *Proceedings IEEE International Conference on Pattern Recognition*, 2004.

[18] J. Cavazza, P. Morerio, and V. Murino. When kernel methods meet feature learning: Log-covariance network for action recognition from skeletal data. In *IEEE Computer Vision and Pattern Recognition (CVPR) workshops*, 2017.

[19] J. Cavazza, A. Zunino, M. San Biagio, and V. Murino. Kernelized covariance for action recognition. In *23rd IEEE International Conference on Pattern Recognition (ICPR), 2016 (pp. 408-413). .*, pages 408–413, 2016.

[20] Z. Chebbi and M. Moakher. Means of hermitian positive-definite matrices based on the log-determinant alpha-divergence function. *Linear Algebra and its Applications*, 2012.

[21] L. Chen and R. Ng. On the marriage of lp-norms and edit distance. In *Proceedings of the Thirtieth international conference on Very large data bases*, volume 30, pages 792–803, 2004.

[22] L. Chen, M.T. Özsu, and V. Oria. Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, page 491–502, 2005.

[23] Yanping Chen, Eamonn Keogh, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, and Gustavo Batista. The ucr time series classification archive, July 2015. `www.cs.ucr.edu/~eamonn/time_series_data/`.

[24] A. Cherian and S. Sra. Riemannian dictionary learning and sparse coding for positive definite matrices. *IEEE Transactions on Neural Networks and Learning Systems*, 2017.

[25] A. Cherian, S. Sra, A. Banerjee, and N. Papanikolopoulos. Efficient similarity search for covariance matrices via the jensen-bregman logdet divergence. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2399–2406. IEEE, 2011.

[26] C.Piciarelli, C.Micheloni, and G.Foresti. Trajectory-based anomalous event detection. *IEEE Transactions on Circuits and Systems For Video Technology*, 2008.

[27] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, 2005.

[28] T. Dean, M.A.Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik. Fast, accurate detection of 100,000 object classes on a single machine. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1814–1821, 2013.

[29] H. Deng, G. Runger, E. Tuv, and M. Vladimir. A time series forest for classification and feature extraction. *Information Science*, 239:142–153, 2013.

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

[31] Z. Dong, S. Jia, C. Zhang, M. Pei, and Y. Wu. Deep manifold learning of symmetric positive definite matrices with application to face recognition. In *AAAI*, pages 4009–4015, 2017.

[32] I. A. Dryden, A. Koloydenko, and D. Zhou. Non-euclidean statistics for covariance matrices, with applications to diffusion tensor imaging. *The Annals of Applied Statistics*, pages 1102–1123, 2009.

[33] E. Erdem and A. Erdem. Visual saliency estimation by nonlinearly integrating features using region covariances. *Journal of Vision*, 2013.

[34] H. Ergezer and K. Leblebicioğlu. Anomaly detection and activity perception using covariance descriptor for trajectories. In *European Conference on Computer Vision, Springer International Publishing*, pages 728–742, 2016.

[35] P. Esling and C. Agon. Time-series data mining. *ACM Computing Surveys (CSUR)*, 45(1), 2012.

[36] O. Freifeld. *Statistics on Manifolds with Applications to Modeling Shape Deformations*. PhD thesis, Brown University, 2014.

[37] W. Förstner and B. Moonen. A metric for covariance matrices. *Geodesy-The Challenge of the 3rd Millennium. Springer Berlin Heidelberg*, 2003.

[38] Z. Fu, G. Lu, K.M. Ting, and D. Zhang. Music classification via the bag-of features approach. *Pattern Recognition Letters*, 32(14):1768–1777, 2011.

[39] B. D. Fulcher and N. S. Jones. Highly comparative feature-based time-series classification. *IEEE Transactions on Knowledge and Data Engineering*, 26(12):3026–3037, 2014.

[40] K. Guo, P. Ishwar, and J. Konrad. Action recognition from video using feature covariance matrices. *IEEE Transactions on Image Processing*, 2013.

[41] F. Han, B. Reily, W. Hoff, and H. Zhang. Space-time representation of people based on 3d skeletal data: A review. *Computer Vision and Image Understanding*, 158:85–105, 2017.

[42] M. T. Harandi, M. Salzmann, and R. Hartley. From manifold to manifold: Geometry-aware dimensionality reduction for spd matrices. In *European Conference on Computer Vision*, pages 17–32. Springer, 2014.

[43] J. Hills, J. Lines, E. Baranauskas, J. Mapp, and A. Bagnall. Classification of time series by shapelet transformation. *Data Mining and Knowledge Discovery*, 28(4):851–881, 2014.

[44] T. Hospedales, S. Gong, and T. Xiang. A markov clustering topic model for mining behaviour in video. In *IEEE International Conference on Computer Vision*, pages 1165–1172, 2009.

[45] W. Hu, D. Xie, Z. Fu, W. Zeng, and S. Maybank. Semantic-based surveillance video retrieval. *IEEE Transactions on Image Processing, vol. 16, no. 4, pp. 1168-1181*, April 2007.

[46] Z. Huang and L. Van Gool. A riemannian network for spd matrix learning. In *AAAI*, 2017.

[47] Z. Huang, C. Wan, T. Probst, and L. Van Gool. Deep learning on lie groups for skeleton-based action recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[48] Z. Huang, R. Wang, S. Shan, X. Li, and X. Chen. Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification. In *International Conference on Machine Learning*, pages 720–729, 2015.

[49] M.E. Hussein, M. Torki, M.A. Gowayyed, and M. El-Saban. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In *IJCAI*, volume 13, pages 2466–2472, 2013.

[50] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi. Kernel methods on the riemannian manifold of symmetric positive definite matrices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 73–80, 2013.

[51] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi. Kernel methods on riemannian manifolds with gaussian rbf kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(12):2464–2477, 2015.

[52] Y. Jeong, M. Jeong, and O. Omitaomu. Weighted dynamic time warping for time series classification. *Pattern Recognition*, 44(9):2231–2240., 2011.

[53] J.Yagnik, D. Strelow, D.A. Ross, and R.S Lin. The power of comparative reasoning. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2431–2438, 2011.

[54] I. Karlsson, P. Papapetrou, and H. Boström. Generalized random shapelet forests. *Data Mining and Knowledge Discovery*, 30(5):1053–1085, 2016.

[55] R. J. Kate. Using dynamic time warping distances as features for improved time series classification. *Data Mining and Knowledge Discovery*, 30(2):283–312, 2016.

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

[57] E. Keogh and J. Lin. Clustering of time-series subsequences is meaningless: implications for previous and future research. *Knowledge and Information Systems*, 8(2):154–177, 2005.

[58] E. Keogh, J. Lin, and A. Fu. Hot sax: Efficiently finding the most unusual time series subsequence. In *ICDM*, 2005.

[59] E. Keogh, S. Lonardi, and C.A. Ratanamahatana. Towards parameter-free data mining. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 206–215, 2004.

[60] E. Keogh and M. Pazzani. Scaling up dynamic time warping for datamining applications. In *International Conference on Knowledge Discovery and Data Mining*, 2000.

[61] E. J. Keogh and M. J. Pazzani. Derivative dynamic time warping. In *Proceedings of the 2001 SIAM International Conference on Data Mining*, 2001.

[62] J. Koutnik, K. Greff, F. Gomez, and J. Schmidhuber. A clockwork rnn. In *International Conference on Machine Learning*, pages 1863–1871, 2014.

[63] D. Kuettel, M. D. Breitenstein, L. Van Gool, and V. Ferrari. What's going on? discovering spatio-temporal dependencies in dynamic scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1951–1958, 2010.

[64] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005.

[65] R. Laxhammar and G. Falkman. Online learning and sequential anomaly detection in trajectories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

[66] J. M. Lee. *Riemannian manifolds: an introduction to curvature*, volume 176. Springer Science & Business Media, 2006.

[67] W. Li, Z. Zhang, and Zicheng Liu. Action recognition based on a bag of 3d points. In *IEEE International Workshop on CVPR for Human Communicative Behavior Analysis*, 2010.

[68] Weixin Li, Vijay Mahadevan, and Nuno Vasconcelos. Anomaly detection and localization in crowded scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

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

[70] J. Lines and A. Bagnall. Time series classification with ensembles of elastic distance measures. *Data Mining and Knowledge Discovery*, 29:565–592, 2015.

[71] J. Liu, A. Shahroudy, D. Xu, and G. Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *European Conference on Computer Vision*, pages 816–833, 2016.

[72] J. Liu, G. Wang, P. Hu, L. Duan, and A.C. Kot. Global context-aware attention lstm networks for 3d action recognition. In *Proceedings of Computer Vision and Pattern Recognition*, pages 1647–1656, 2017.

[73] D.G. Lowe. Object recognition from local scale-invariant features. In *The Proceedings of the Seventh IEEE international Conference on Computer Vision*, volume 2, pages 1150–1157, 1999.

[74] M. Salzmann M. Harandi and F. Porikli. Bregman divergences for infinite dimensional covariance matrices. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[75] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. Anomaly detection in crowded scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[76] P. F. Marteau. Time warp edit distance with stiffness adjustment for time series matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):306–318, 2009.

[77] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[78] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(4):349–361, April 2001.

[79] B. Morris and M. Trivedi. Learning trajectory patterns by clustering: Experimental studies and comparative evaluation. In *CVPR*, 2009.

[80] B. Morris and M. M. Trivedi. Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[81] A. Mueen, E. Keogh, and N. Youngin. Logical-shapelets: An expressive primitive for time series classification. In *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, page 1154–1162, 2011.

[82] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber. Documentation mocap database hdm05. Technical Report CG-2007-2, Universität Bonn, June 2007.

[83] D. Neil, M. Pfeiffer, and S.C. Liu. Phased lstm: Accelerating recurrent network training for long or event-based sequences. In *Advances in Neural Information Processing Systems 2016*, pages 3882–3890, 2016.

[84] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems 2*, 2002.

[85] X. Pennec, P. Fillard, and N. Ayache. A riemannian framework for tensor computing. *International Journal of Computer Vision*, 66(1):41–66, 2006.

[86] S. Pongpaichet, M. Tang, L. Jalali, and R. Jain. Using photos as micro-reports of events. In *In Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, pages 87–94. ACM, 2016.

[87] F. Porikli, O. Tuzel, and P. Meer. Covariance tracking using model update based on lie algebra. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[88] C. A. Ratanamahatana and E. Keogh. Making time-series classification more accurate using learned constraints. In *In Proceedings of SIAM International Conference on Data Mining (SDM04)*, page 11–22, 2004.

[89] C. A. Ratanamahatana, J. Lin, D. Gunopulos, E. Keogh, M. Vlachos, and G. Das. Mining time series data. *Data Mining and Knowledge Discovery Handbook*, pages 1049–1077, 2009.

[90] K. Sadatnejad and S. S. Ghidary. Kernel learning over the manifold of symmetric positive definite matrices for dimensionality reduction in a bci application. *Neurocomputing*, 179:152–160, 2016.

[91] A. Shahroudy, J. Liu, T.T. Ng, and G. Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1010–1019, 2016.

[92] S. Sra. A new metric on the manifold of kernel matrices with application to matrix geometric means.". In *Advances in neural information processing systems*, pages 144–152, 2012.

[93] S. Sra and A. Cherian. Generalized dictionary learning for symmetric positive definite matrices with application to nearest neighbor retrieval. *Machine Learning and Knowledge Discovery in Databases*, pages 318–332, 2011.

[94] J. Su, S. Kurtek, E. Klassen, and A. Srivastava. Statistical analysis of trajectories on riemannian manifolds: Bird migration, hurricane tracking and video surveillance. *The Annals of Applied Statistics*, 8(1):530–552, 2014.

[95] J. Su, A. Srivastava, F. Souza, and S. Sarkar. Rate-invariant analysis of trajectories on riemannian manifolds with application in visual speech recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 620–627, 2014.

[96] M. Tang, P. Agrawal, F. Nie, S. Pongpaichet, and R. Jain. A graph based multimodal geospatial interpolation framework. In *In 2016 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2016.

[97] M. Tang, F. Nie, and R. Jain S. Pongpaichet. From photo streams to evolving situations. *arXiv preprint arXiv:1702.05878*, 2017.

[98] M. Tang, X. Wu, P. Agrawal, S. Pongpaichet, and Ramesh Jain. Integration of diverse data sources for spatial pm2. 5 data interpolation. *IEEE Transactions on Multimedia*, 2016.

[99] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *European Conference on Computer Vision*, 2006.

[100] O. Tuzel, F. Porikli, and P. Meer. Pedestrian detection via classification on riemannian manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008.

[101] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 588–595, 2014.

[102] H. Wang, Y. Yi, and J. Wu. Human action recognition with trajectory based covariance descriptor in unconstrained videos. In *Proceedings of the 23rd ACM international conference on Multimedia 2015*, pages 1175–1178, 2015.

[103] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1290–1297, 2012.

[104] L. Wang, J. Zhang, L. Zhou, C. Tang, and W. Li. Beyond covariance: Feature representation with nonlinear kernel matrices. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4570–4578, 2015.

[105] X. Wang, K. T. Ma, G. W. Ng, and W. E. L. Grimson. Trajectory analysis and semantic region modeling using nonparametric hierarchical bayesian models. *International Journal of Computer Vision*, 2011.

[106] X. Wang, X. Ma, and W. E. L. Grimson. Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.

[107] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and Eamonn Keogh. Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, 26(2):275–309, 2013.

[108] S. Wu, B. E. Moore, and M. Shah. Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[109] L. Xia, C. Chen, and J. Aggarwal. View invariant human action recognition using histograms of 3d joints. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 20–27, 2012.

[110] L. Ye and E. Keogh. Time series shapelets: A new primitive for data mining. In *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, page 947–956, 2009.

[111] Q. Zhang, S.A. Goldman, W. Yu, and J.E. Fritts. Content-based image retrieval using multiple-instance learning. In *International Conference on Machine Learning*, volume 2, pages 682–689, 2002.

[112] X. Zhang, Y. Wang, M. Gou, M. Sznaier, and O. Camps. Efficient temporal sequence comparison and classification using gram matrix embeddings on a riemannian manifold. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4498–4507, 2016.

[113] Z. Zhang, K. Huang, and T. Tan. Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes. In *International Conference Pattern Recognition*, 2006.

[114] J. Zhao and L. Itti. Classifying time series using local descriptors with hybrid sampling. *IEEE Transactions on Knowledge and Data Engineering*, 28(3):623–637, March 2016.

[115] B. Zhou, X. Tang, and X. Wang. Measuring crowd collectiveness. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3049–3056, 2013.

[116] B. Zhou, X. Wang, and X. Tang. Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2871–2878, 2012.

[117] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, and X. Xie. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In *AAAI*, volume 2, page 8, 2016.

[118] Y. Zhu, W. Chen, and G. Guo. Fusing spatiotemporal features and joints for 3d action recognition. In *Computer Vision and Pattern Recognition Workshops*, page 486–491, 2013.

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Ergezer, Hamza
Nationality: Turkish (TC)
Date and Place of Birth: 1982, Kırşehir
Marital Status: Married
Phone: 0533 4706483
Fax: 0312 8475320

EDUCATION

Degree	Institution	Year of Graduation
B.S.	Electrical and Electronics Engineering, Hacettepe University	2004
M.S.	Electrical and Electronics Engineering, METU	2007

PROFESSIONAL EXPERIENCE

Year	Place	Enrollment
2004-present	ASELSAN/MGEO	Lead Design Engineer

PUBLICATIONS

Journal Publications

- H. Ergezer, K. Leblebicioglu, “Time Series Classification Using Feature Covariance Matrices”, Knowledge and Information Systems, accepted, 2017.

International Conference Publications

- H. Ergezer, K. Leblebicioglu, “Anomaly Detection and Activity Perception Using Covariance Descriptor for Trajectories”, Workshop on Crowd Understanding, ECCV 2016.
- E. Gundogdu, H. Ozkan, S. Demir, H. Ergezer, E. Akagunduz, K. Pakin, “Comparison of Infrared and Visible Imagery for Object Tracking: Toward Trackers with Superior IR Performance”, Workshop on Perception Beyond the Visible Spectrum, CVPR 2015.

National Conference Publications

- H. Ergezer, K. Leblebicioglu, “Time Series Classification Using Point-wise Features”, SIU 2017.
- H. Ergezer, K. Leblebicioglu, “Anomaly Detection in Trajectories”, SIU 2016.
- Ekin Gedik, Umut Çinar, Ersin Karaman, Yasemin Yardımcı, Uğur Halıcı, Kubilay Pakin, Hamza Ergezer, "Automatic Water Canal Detection in Multispectral Satellite Images", SIU 2013.
- H. Ergezer, K. Leblebicioglu, “Visual Tracking of Objects via Rule-Based Multiple Hypothesis Tracking”, SIU 2008.
- H. Ergezer, K. Leblebicioglu, “Visual Detection and Tracking of Moving Objects”, SIU 2007.