Time Series Mining： A Computational Intelligence Approach

by

Jinbo Li

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Software Engineering and Intelligent Systems

Department of Electrical and Computer Engineering
University of Alberta

Abstract

Time series has become prevalent in a broad range of real-world applications such as weather, health care, agricultural production, satellite image analysis, speech recognition, industrial process control, and others. This type of data comes as a collection of observations obtained chronologically, describing different aspects of a specific phenomenon. With the increasing availability of time series data, the discovery and extraction of available information (e.g., similar patterns, meaning rules) from them are essential to human. In this dissertation, three main time series mining tasks involving (i) anomaly detection, (ii) approximation /representation and (iii) predictive modeling will be concerned with developing Computational Intelligence (CI) related techniques on the one hand and with their application on complex real-world problems on the other hand. The primary objectives of this thesis are to develop a series of relatively comprehensive frameworks for these mining tasks.

Anomaly detection in the multivariate time series refers to the discovery of any abnormal behavior within the data encountered in a specific time interval. Here we develop and carry out two unsupervised and supervised frameworks of multivariate time series anomaly detection for amplitude and shape anomalies, namely cluster-centric anomaly detection models and Hidden Markov Models based model with the aid of the transformation of multivariate time series to univariate time series respectively. In the first unsupervised model, the modified Fuzzy C-Means clustering was used to capture the structure of multivariate time series. A reconstruction error serves as the fitness function of the PSO algorithm and also has been considered as the level of anomaly detected in each subsequence. In the other model, several transformation techniques involving Fuzzy

C-Means (FCM) clustering and fuzzy integral are studied. A Hidden Markov Model (HMM), one of the commonly encountered statistical methods, is engaged to detect anomalies in multivariate time series.

Before implementing most tasks of time series data mining, one of the essential problems is to approximate or represent the time series data because of its massive data size and high dimensionality. We establish FCM clustering based approximation methods. We carry out a comprehensive analysis of relationships between reconstruction error and classification performance when dealing with various representation (approximation) mechanisms of time series.

Furthermore, we also elaborate on a novel Hidden Markov Model (HMM)-based fuzzy model for time series prediction. Here fuzzy rules (rule-based models) are employed to describe and quantify the relationship between the input and output time series while the HMM is regarded as a vehicle for capturing the temporal behavior or changes of the multivariate time series. A suite of experimental studies along with some comparative analysis is reported on both synthetic and real-world time series data sets.

Preface

The research conducted in this thesis was performed by Jinbo Li under the supervision of Prof. Witold Pedrycz.

Chapter 3 of this thesis includes the materials published as J. B. Li, W. Pedrycz and others "The Alberta Veterinary Surveillance Network Veterinary Practice Surveillance System for Cattle: Development of a Tool to Track Cattle Diseases and Movement in Alberta, Canada" proceedings of the *Conference of Research Workers in Animal Diseases*, December 7-9, 2014, Chicago, Illinois, USA and submitted to *IEEE Transactions on Systems, Man and Cybernetics: Systems* as J. B. Li, H. Izakian, W. Pedrycz, and I. Jamal "Cluster-Centric Anomaly Detection in Multivariate Time Series Data." I was responsible for the idea and coding development, data collection and collection, as well as the manuscript composition. W. Pedrycz was the supervisory author and was involved with the concept formation and manuscript composition.

Chapter 4 of this thesis has been published in *Applied Soft Computing* as J. B. Li, W. Pedrycz, and I. Jamal "Multivariate Time series Anomaly Detection: A Framework of Hidden Markov Models." I was responsible for the idea and coding development, data collection and collection, as well as the manuscript composition. W. Pedrycz was the supervisory author and was involved with the concept formation and manuscript composition.

Chapter 5 of this thesis has been submitted to *Neurocomputing* as J. B. Li, W. Pedrycz, and A. Gacek "Time Series Reconstruction and Classification: A Comprehensive Comparative Study." I was responsible for the idea and coding development, data collection and collection, as well as the manuscript composition. W. Pedrycz was the supervisory author and was involved with the concept formation and manuscript composition.

Chapter 6 of this thesis has been published in *International Journal of Approximate Reasoning* as J. B. Li, W. Pedrycz, and X. M. Wang "A rule-based development of incremental models" and submitted to *Expert Systems with Applications* as J. B. Li, W. Pedrycz, and X. M. Wang "A Hidden Markov Model-Based Fuzzy Modeling of Multivariate Time Series." I was responsible for idea and coding development, data

collection and collection, as well as the manuscript composition. W. Pedrycz was the supervisory author and was involved with the concept formation and manuscript composition.

Acknowledgements

First and foremost, I would like to owe a great debt of gratitude and most profound appreciation to my supervisor, Professor Witold Pedrycz. I had the profound honor of being his Ph.D. student. His enthusiasm, creativity, diligence, patience, motivation, and encouragement have been invaluable to me. I truly appreciate all his contribution and could not have wished for a better supervisor.

I would also like to express my gratitude to the members of my supervisory committee, Professor Marek Reformat, Professor Mojgan Daneshmand, Professor Ergun Kuru, Professor Petr Musilek, and Professor Francesco Marcelloni. Their brilliant suggestions and inspiration helped me a lot throughout the whole Ph.D. journey. I appreciate all of their contributions and ideas and the time.

Last but not least, I would like to thank my parents and family for their support, encouragement and incredible tolerance.

Jinbo Li
University of Alberta
August 2018

Table of Contents

List of Tables

List of Figures

List of Symbols

Multivariate time series:

$\boldsymbol{X}$	A multivariate time series;
$\boldsymbol{x}_k$	The k^{th} instance (data) in a multivariate time series;
$\boldsymbol{U}$	Partition matrix $\boldsymbol{U}=[u_{ik}]$;
u_{ik}	Membership degree of k^{th} instance to i^{th} cluster;
$\boldsymbol{v}_i$	The i^{th} prototype;
$\boldsymbol{w}_j$	The j^{th} multivariate subsequence present in the sliding window;
n	The number of variables of multivariate time series;
c	The number of clusters;
m	Fuzzification coefficient;
p	The length of the multivariate time series;
q	The length of subsequence;
r	The length of movement of the sliding window;
N	The number of subsequences;
λ_i	The weight of i^{th} feature;
s_j	Anomaly score of j^{th} subsequence;
g	Fuzzy measure;
$Q=\{q_1,q_2,\ldots,q_M\}$	The hidden state set;
M	The number of hidden states;
$R=\{r_1,r_2,...,r_L\}$	The observed state set;
L	The number of observed states;
$\boldsymbol{O}=o_1,o_2,\ldots,o_T$	An observed state sequence;
$\boldsymbol{I}=i_1,i_2,\ldots,i_T$	A hidden state sequence;
$\psi=(\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi})$	A Hidden Markov Model;
T	The length of training multivariate time series;
T'	The length of testing multivariate time series;

Univariate time series:

$\boldsymbol{X}$	A dataset of univariate time series;
$\boldsymbol{x}_k$	The k^{th} univariate time series in the dataset $\boldsymbol{X}$;
N	The number of univariate time series;
n	The length of variables of each univariate time series;
m	The number of essential features;

Chapter 1

Introduction

In many real-world applications including fault diagnosis [1], energy consumption of electric vehicle (EV) charging station [2], healthcare [3], wind speed prediction [4-7] and so on [8-16], collected data arise in the form of time series. With the increasing availability of time series, the discovery and extraction of information (e.g., similar patterns, meaningful rules) from them are essential to human. In this study, we are concerned with three core time series data mining tasks, which are anomaly detection, approximation/representation, and modeling. For these tasks mentioned above, a series of Computational Intelligence (CI) related techniques have been developed and applied in real-world data. Specifically, anomaly detection in this type of data refers to the discovering of any abnormal behavior within the data encountered in a specific time interval. Time series representation (or approximation), which maps the original time series to the feature space of (usually) lower dimensionality comes with a variety of approximation methods completed in time or frequency domains. Time series prediction, referring to the development of models of dynamic systems realized by using a collection of the past observations, has been one of the essential research pursuits in time series analysis [17-20].

1.1 Motivation

1.1.1 Time series anomaly detection

Anomaly detection of multivariate time series has been widely used in numerous applications [21-26]. For instance, cardiologists are interested in identifying anomalous parts of ECG signals to diagnose heart disorders. Economists are interested in anomalous parts of share prices to analyze and build economic models. Meteorologists are interested in anomalous parts of weather data to predict future consequences. Therefore, it is a beneficial challenge to design and develop frameworks for anomaly detection in multivariate time series. Although numerous time series anomaly detection techniques have been reported in the literature, see e.g., [27-30], most of these techniques are

concerned with univariate time series. Compared with the techniques which deal with univariate time series anomaly detection, the algorithms purpose has to consider all variables at the same time to determine an anomaly score [29-31]. Therefore, we turn to cluster-centric based model and the transformation methods from multivariate time series to univariate time series.

1.1.2 Time series approximation/representation

To overcome the challenge of massive data size and high dimensionality, the approximation or representation of time series is often performed before most other tasks of time series data mining. Time series representation methods can offer tangible benefits when we assume that the subsequent step is time series classification. First, in most cases, they can improve the classification accuracy of the classifiers being realized in the developed representation space, which commonly comes with implicit noise removal. Second, having the representation coefficients of the raw time series positioned in the low dimensional feature space, these representation methods have the ability to speed up the classification process, also improving efficient storage of data. Third, the concise and essential characteristics of the original time series can be captured through the representation methods. Thus, most representation methods should exhibit some highly desirable properties, e.g., supporting dimensionality reduction, offering high reconstruction quality, exhibiting noise robustness, etc. so that they can support a sound way of achieving high classification performance. Despite the diversity of the existing methods used in time series representation and classification, it is quite uncommon to encounter studies that report on the relationships between the classification error and the representation properties (e.g., reconstruction quality) of different time series representations. Here we reveal, quantify, and visualize the relationships between the reconstruction error and classification error (classification rate) for a number of commonly encountered representation methods.

1.1.3 Time series modeling

To produce prediction results of high accuracy, there has been a tremendous wealth of techniques or algorithms along with diversified architectures, learning strategies, and numerous hybrid mechanisms focused on the efficient modeling and forecasting time

series, e.g., fuzzy systems [17, 18, 32], recurrent neural networks [33-35], evolutionary algorithms [36, 37] among others. One can refer here to an impressive plethora of the time series prediction approaches existing in the literature [38-40]. Despite the visible diversity of available approaches, almost all of them try to realize modeling the temporal relationship and capturing the characteristic of time series to estimate the future/unknown values in the series. The research challenges in time series prediction associate with the non-linearity and inherent volatility of time series, especially those generated by complex systems [40-42]. In general, most studies assume that a single model can fit a given time series adequately. However, in practice, the time series generated by complex systems may not satisfy the assumption because of the complexity and variability of its structure. Therefore, in this work, Hidden Markov Model (HMM) is applied to model the temporal dependence of multivariate time series where a set of fuzzy rules are introduced to describe the relationship between input and output time series.

1.2 Objectives and originality

The primary objectives of this study are listed below:

1. Developing a general framework for anomaly detection (both in shape and amplitude) in multivariate time series.

2. Developing an augmented version of the Fuzzy C-Means clustering to reveal the available structure within multivariate time series data.

3. Investigate the multivariate time series anomaly detection problem by involving different transformation methods and HMM.

4. Design and propose an HMM-based anomaly detector for multivariate time series and compare different transformation approaches in HMM-based anomaly detection methods.

5. Perform and quantify comparative study of different time series approximation algorithms on an extensive set of time series and report the relationship between the classification error and the reconstruction error of various time series representation methods.

6. Investigate an impact of the parameters of the representation methods (such as those present in clustering methods and others) on the resulting reconstruction error and the classification rate.

7. Design and develop a Hidden Markov Model (HMM)-based fuzzy model, which dwells upon the Hidden Markov Model and fuzzy rules by bringing them together. In this framework, HMM is applied to model the temporal dependence of the multivariate time series where a set of fuzzy rules are introduced to describe the relationship between the input and output time series.

In this dissertation, three core time series data mining tasks have been archived through using the proposed techniques or methods. As for time series anomaly detection, unsupervised and supervised frameworks are proposed. Then, as for different time series approximation method, the time series classification error, the reconstruction error, and their relationships are revealed, quantified and visualized. Additionally, compared with the fuzzy model without the aid of HMM, the HMM-based fuzzy model can produce better prediction results producing lower values of the corresponding criterion (RMSE), which results in that HMM can model the temporal changes of multivariate time series.

This research exhibits a significant level of originality:

- The cluster-centric anomaly detection method exhibits a certain deal of originality by offering a unified framework for detecting anomalous segments of data with respect to the amplitude and/or shape information in multivariate time series data.
- Detecting anomalous parts of multivariate time series data with the use of available clusters within the data is comes as a novel idea proposed here.
- An augmented Fuzzy C-Means clustering technique established in this dissertation exhibits a certain level of originality.
- Investigate some transformation methods and study their performance with respect to abilities to retain useful information (e.g., amplitude or amplitude change) in HMM-based anomaly detection methods.
- As for time series approximation/representation, The FCM based time series approximation method and its corresponding reconstruction used also exhibit some level of originality in the sense.

- The novelty/originality of time series prediction realized here stems from the fact that the optimized HMM can model the temporal relationship through governing a series of generated fuzzy rules.
- In contrast to the traditional HMM which manages a finite number of states/values, the proposed system where the value varies within a continuous range also comes with some originality.

1.3 Organization

The thesis is structured into the following chapters:

Chapter 2 covers a series of time series mining techniques involving time series modeling, approximation/representation and anomaly detection, which are most related to the fuzzy rule-based model.

Chapter 3 proposes a framework of Hidden Markov Models for multivariate time series anomaly detection. Several transformation techniques involving Fuzzy C-Means (FCM) clustering and fuzzy integral are studied. In the sequel, a Hidden Markov Model (HMM), one of the commonly encountered statistical methods, is engaged here to detect anomalies in multivariate time series.

Chapter 4 poses a new cluster-centric approach to detect anomalies in the amplitude as well as the shape of multivariate time series. An augmented fuzzy clustering is developed to reveal a structure present within the generated multivariate subsequences. A reconstruction error serves as the fitness function of the PSO algorithm and also has been considered as the level of anomaly detected in each subsequence.

Chapter 5 introduce the FCM based time series approximation method and its corresponding reconstruction. The study also realized several commonly countered time series approximation methods and their corresponding reconstruction process. The relationships between the reconstruction error and classification error (classification rate) for some commonly encountered representation methods have been revealed, quantified, and visualized.

Chapter 6 discusses the cooperation of rule-based representations (e.g., fuzzy rules) and temporal model (e.g., HMM) is insightful to elaborate on the joint dynamic behavior of multivariate time series. A novel Hidden Markov Model (HMM)-based fuzzy model for

time series prediction has been proposed. The proposed strategies control the contribution of different fuzzy rules so that the proposed model can well model the dynamic behavior of time series.

Chapter 7 draws a series of conclusions of this dissertation and suggests a series of potential directions.

Chapter 2

Background and literature review

This chapter firstly covers a series of time series modeling techniques which are most related to the fuzzy rule-based model. Then a variety of time series anomaly detection methods which consist of similarity-based, clustering-based, classification-based, transformation-based and modeling-based algorithms have been reported. The final subsection will review the relevant literature in the area of time series approximation/representation.

2.1 Anomaly detection of time series

Various anomaly detection techniques for univariate time series data have been proposed in the literature (refer e.g., to [29-31]). Compared to univariate time series, anomaly detection in multivariate time series has been more challenging since more than a single variable must be considered simultaneously when detecting anomalous segments of data. Methods of anomaly detection in time series data can be divided into a set of categories, namely similarity-based methods [43, 44], clustering-based methods [45, 46], classification-based methods, modeling-based methods [47, 48], frequency-based methods [49], and probability-based methods [50]. A thorough survey of time series anomaly detection techniques is reported in [29-31]. In what follows, we briefly recall the main features of these approaches.

2.1.1 Similarity-based methods

A simple technique to determine anomalies in time series is to use a similarity measure along with a brute force algorithm [51]. Subsequences of time series having the highest differences from the other subsequences are considered as anomalies. In similarity-based techniques, selecting a suitable similarity measure might have a substantial impact on the performance of the method and directly depends on the specific application-purpose [52]. For instance, when the time series are collected at different sampling rates, Dissim distance [53] can be considered since it uses a finite set to define a time series. Dynamic time warping distance (DTW) [54] is another effective technique that can be considered when

the lengths of time series are unequal and there are temporal shifts in data. Some other similarity measures such as Longest common subsequence (LCSS), Edit Distance on Real sequence (EDR) and alike are widely reported in the literature [55]. Another approach [56] to detect anomalies contains two main steps: firstly, employ sparse coding to extract features, and then use latent semantic analysis (LSA) to learn relationship. Finally, the squared reconstruction errors are considered as anomaly scores. Note that the reference time series used in this method should be without any abnormal subsequences. In [57], a linear method for anomaly detection is proposed; it demonstrates the advantages of low time complexity and lower number of parameters. However, only the top discords k can be detected and there is no detailed guidance to determine the preferred value of k.

2.1.2 Clustering-based methods

Applying clustering methods in multivariate time series is another option. In general, clustering-based techniques are composed of two main steps [58]. In order to generate subsequences of multivariate time series, the first stage is to implement a clustering method such as K-Means or Fuzzy C-Means (FCM) over the time series. Other clustering techniques can also be considered as well [59, 60]. At the second stage, they determine an anomaly score on the basis of some measurement of the fitness of subsequences [58] to the different clusters. Then the following way to determine the anomaly score is to use the distance between instances and cluster centers.

In [61], the weighted Euclidean distance between the observations is computed and an improved ant colony method is exploited to complete clustering. In [62], a novel clustering-based compression method is proposed to reduce the time complexity of the approach. K-Means algorithm is utilized to convert a correlation matrix of multivariate time series to a matrix of multiple clusters. According to the multivariate normal distributions, the anomaly scores were estimated. In [63], a Bounded Coordinate System (BCS) approach is considered to evaluate the similarity between two multivariate time series. A modified version of K-Means is employed to cluster the multivariate time series dataset. The top k outlier observations were detected using a two-pruning rule-based nested loop algorithm.

2.1.3 Classification-based methods

In classification-based methods, instances or subsequences of multivariate time series are classified into two classes: normal and abnormal. The classifier is trained through a training set composed of normal instances and then it can be used to assign an anomaly score to each instance of the testing set [29]. In [50], the authors proposed to determine whether a data point is abnormal by exploiting the linear regression model firstly and then learn a Bayesian maximum likelihood classifier on the basis of anomalies identified. For the testing dataset, the classifier labels each test data point. However, for classification-based methods, collecting training data that are used to train a classifier is known to be a time expensive process.

2.1.4 Transformation-based methods

An option in dealing with multivariate time series anomaly detection is to reduce the multivariate time series to univariate time series. Then a univariate anomaly detection technique can be employed to detect anomalies. Two common transformation methods are time series projection and independent component analysis [26, 64]. The objective is to reduce the dimensionality of multivariate time series. However, it should be noted that the loss of information in the process of transformation might reduce the accuracy of the multivariate time series anomaly detection [65]. Parthasarathy et al. [66] proposed a novelty dissimilarity measurement for comparing multivariate time series data based on principal component analysis and provided a point anomaly detection algorithm in multivariate time series. The dissimilarity measurement contains distance (Euclidean or Mahalanobis distances), rotation and variance components. The value of dissimilarity was calculated by combining these three components. In order to add potential domain knowledge and improve the flexibility of the method, some coefficients were added into the method to allow potential users to assign different weights to the different components of the time series.

2.1.5 Modeling-based methods

Multivariate time series modeling is another anomaly detection method reported in the literature. In [67], the authors introduced a framework for discovering anomalous regimes in time series (called DARTS). The method is based on the presupposition that the vector

autoregressive process can represent the time series. Given a training time series, local autoregressive models for each subsequence of time series were derived. Then the probability density of the coefficients of each model was computed. Similar to the training time series, the local models of each subsequence of testing time series were calculated. Finally, the probability of the coefficients of each model of testing time series was obtained. The lower probability of the coefficients stands for the higher anomaly degree. Qiu al. etc. [68] proposed to learn graphical models of Granger causality and then exploit the Kullback-Leibler (KL) divergence to compute the anomaly score for each variable. A threshold that is from reference data determines whether the observation is abnormal. Cheng al. et [65] proposed the use of a weighted graph representation to model multivariate time series. RBF function was employed to obtain the similarity between a pair of multivariate time series. In the graph, nodes were subsequences or observations and edges were considered as the similarity between nodes. Random walk algorithm [69] was used to produce the connectivity values. Then the connectivity measure of nodes was considered to detect anomalies. However, for most existing modeling-based methods, their performance strongly depends on the observed data [70]. Moreover, the reference data are required to build models.

On the whole, based on various applications, definitions and background from the literature, there are different ways to formulate the problem of time series anomaly detection. The related techniques can be grouped into supervised and unsupervised categories according to whether the reference or training time series is available. Collecting and annotating training data is a highly time-consuming and not very practical for most real-world applications because domain-specific knowledge is required. For unsupervised approaches, a suitable similarity measure is essential for both the similarity-based methods and the clustering-based methods. Performance of clustering-based methods heavily depends on the available structure of time series captured by the clustering methods.

2.2 Time series approximation/representation

Time series and temporal data are a frequently encountered types of data present in numerous real-world phenomena (e.g., finance [71], agriculture [72], telecommunication

[73], etc. [74, 75]). In time series classification, we encounter a plethora of various techniques, just to recall naïve Bayes [76], support vector machines (SVMs) [77], decision trees [78], neural networks [79], nearest neighbor classifiers [80] and other methods [81-83]. Processes of building classifiers are intrinsically related to time series (sequences) under consideration and realized by learning the essential characteristics of each class. What is important is the fact that the representation of time series leading to their distribution in a certain space of parameters of the ensuing representation impacts the quality of ensuing classifiers directly. Also, it is essential to note that the running time of each classification module is essentially associated with the dimensionality of the space in which the time series are represented. From this perspective, time series representation (or approximation), which maps the original time series to the feature space of (usually) lower dimensionality comes with a variety of approximation methods completed in time or frequency domains. In this sense, there are several tangible benefits, which time series representation methods offer. First, in most cases, they can improve the classification accuracy of the classifiers being realized in the developed representation space, which commonly comes with implicit noise removal. Second, having the representation coefficients of the raw time series positioned in the low dimensional feature space, these representation methods have the ability to speed up the classification process, also improving efficient storage of data. Third, the concise and essential characteristics of the original time series can be captured through the representation methods. Thus, most representation methods should exhibit some highly desirable properties, e.g., supporting dimensionality reduction, offering high reconstruction quality, exhibiting noise robustness etc. so that they can support a sound way of achieving high classification performance.

Formally, the i^{th} time series in the available dataset of time series $\boldsymbol{X} = [\boldsymbol{x}_i], i = 1,2, \dots, N$ is regarded as a collection of numeric values represented sequentially over time, say $\boldsymbol{x}_i = x_{i1}, x_{i2}, \dots, x_{in}$, where n and N are the length of each time series and the number of time series under consideration. In other words, the experimental data are represented as a set of N $n-$dimensional vectors located in the input space R^n. The objective of the time series representation methods is to reduce the dimensionality of the original time series $\boldsymbol{x}_i$, namely $\boldsymbol{x}_i \in R^n \to \hat{\boldsymbol{x}}_i \in R^m$ where $\hat{\boldsymbol{x}}_i$ is the corresponding

representation and typically $n \ll m$. The compressed representation $\hat{\boldsymbol{x}}_i = \hat{x}_{i1}, \hat{x}_{i2}, \ldots, \hat{x}_{ik}, \ldots, \hat{x}_{im}$ of length m approximates the original time series $\boldsymbol{x}_i = x_{i1}, x_{i2}, \ldots, x_{ij}, \ldots, x_{in}$ of length n and extracts its m essential features. In what follows, let us recall the essence of each time series representation method and its reconstruction.

2.2.1 Piecewise aggregate approximation (PAA)

The piecewise aggregate approximation (PAA) [84, 85], a dimensionality reduction technique, determines mean values for successive equal-sized windows. More specifically, such segments are made using a non-overlap sliding window technique. Taking into consideration m temporal segments, the average of the k^{th} segment is calculated in the following form.

$$\hat{x}_{ik} = \frac{1}{n} \sum_{l=s_k}^{} x_{il} \tag{2.1}$$

Where s_k and $e_k$are the start and end of the k^{th} segment. An example of piecewise aggregate approximation representation is shown in Figure 2.1; for illustrative purposes, we show the results when changing the number of segments.

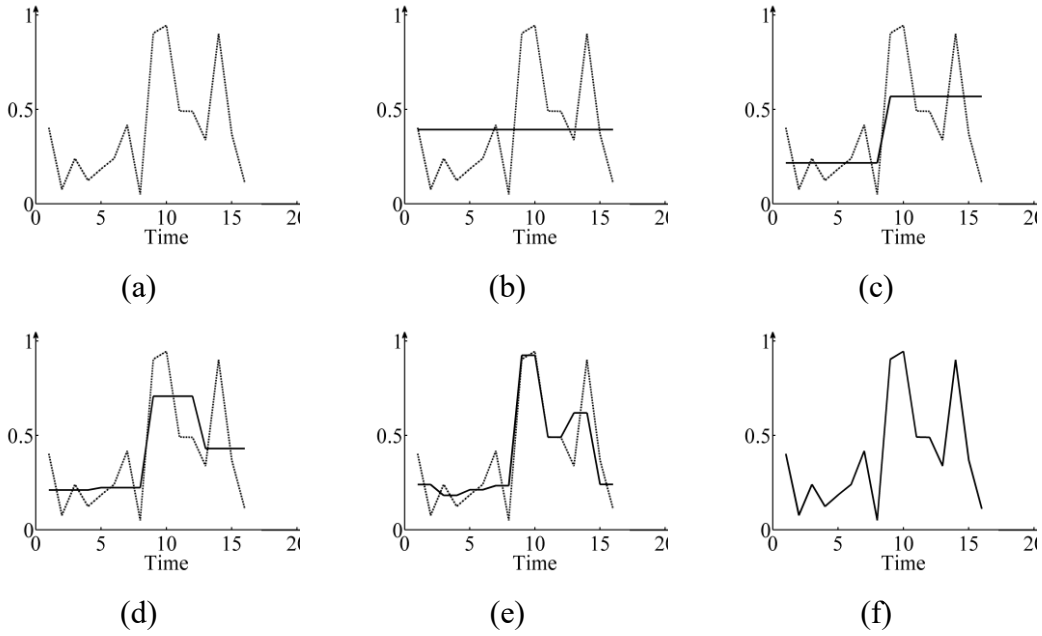


Figure 2.1 PAA time series representations: (a) the original time series of length 16; (b) m=1; (c) m=2; (d) m=4; (e) m=8; (f) m=16;

Note that its PAA representation returns the mean value of the original time series when m=1 while its PAA representation is the same to the original time series when m=n. One can note that the length n of the original time series is not divisible by the number m of segments, which is commonly encountered in practice. The adjacent segments will share some points on the basis of the length of each segment.

2.2.2 Singular Value Decomposition (SVD)

Singular value decomposition (SVD) [86] decomposes a collection of original time series into component matrices and extract interesting and useful properties of the dataset [87]. We construct discrimination "principal components", which are orthogonal to each other. Formally speaking, given a $N \times n$ matrix X, the SVD can be expressed in the following form.

$$X_{N\times n} = U_{N\times n} \times \Lambda_{N\times n} \times V^{T}_{n\times n} \tag{2.2}$$

Here U, Λ and V are the column-orthonormal matrix, the diagonal matrix of eigenvalues and the column-orthonormal matrix, respectively. More detailed discussion of pertinent computations is provided in [88]. The first m columns of U can be used as the SVD coefficients of the original collection of time series.

It is worth noting that SVD is different from other representation methods because SVD is a global mapping/projection while the other methods focus on each time series transformation. In other words, the entire dataset is processed during SVD decomposition while the other representation methods process each time series one by one.

2.2.3 Discrete Fourier Transformation (DFT)

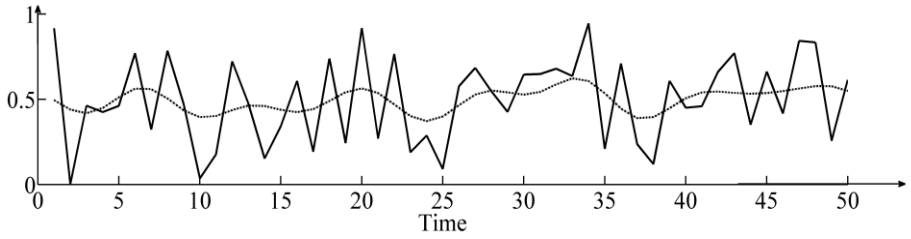
Discrete Fourier Transformation (DFT), which decomposes a time series into a finite number of sine/cosine waves, has been commonly used in various applications, such as discrimination of digital scintillation pulses [89], induction motor bar fault detection [90], image processing [91] and etc. [92]. Each wave comes with a complex number called Fourier coefficient. In essence, both the amplitude and phase of these time series are represented through a collection of sine and/or cosine waves after the application of DFT to the original time series. Consider each time series $\boldsymbol{x}_i$ being described in the frequency domain in the form $\boldsymbol{f}_i = [f_{i1}, f_{i2}, \dots, f_{in}]$, while their Fourier coefficients are calculated in a standard manner

$$f_{ik} = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j e^{\left(-\sqrt{-1}2\pi kj/n\right)} k = 0,1,...,n-1 \tag{2.3}$$

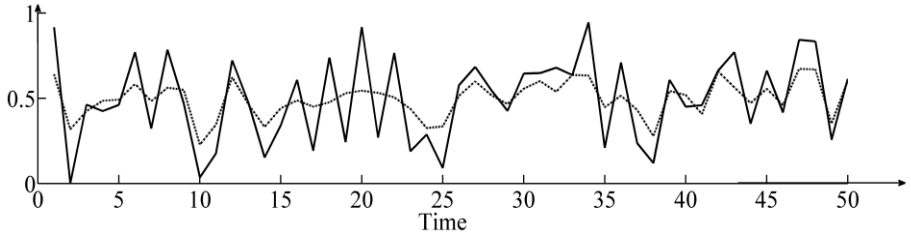
The time series can be recovered through the inverse discrete Fourier transformation

$$\frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} f_k e^{\left(-\sqrt{-1}2\pi kj/n\right)} j = 0,1,...,n-1 \tag{2.4}$$

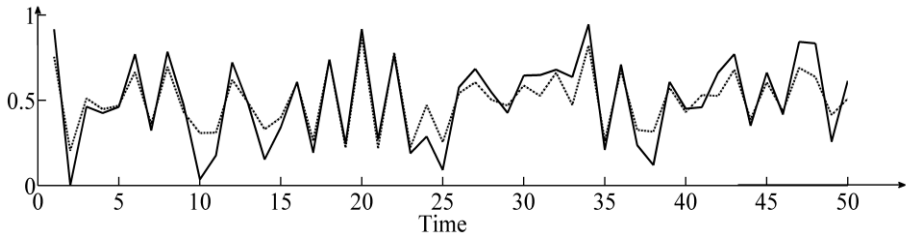
As to the dimensionality of the time series, which remains unchanged after the completion of the DFT transformation; however, the main aspect worth emphasizing is that the reconstruction of original time series can be realized in terms of a few of Fourier coefficients, as shown in Figure 2.2. Omitting most of the coefficients with low amplitude, the final reconstruction of the original time series (without too much information loss) can be archived.



(a)



(b)



(c)

Figure 2.2 Time series reconstruction with the different number of Fourier coefficients. Solid line: input time series/Dotted line: its reconstruction version. (a) *m*=10; (b) *m*=20; (c) *m*=30;

2.2.4 Discrete Wavelet Transformation (DWT)

Discrete Wavelet Transformation (DWT) [93]offers another useful option to handle the dimensionality reduction problem of time series by representing the data by using the sum and difference of the basis functions (wavelets), which is also known as a multi-resolution representation of time series. The Haar Wavelet transform is used as a common vehicle in the wavelet family to represent time series because of its short processing time and easy implementation. As shown in Figure 2.3, for the different number of wavelet coefficients, we encounter different reconstruction versions of the time series.

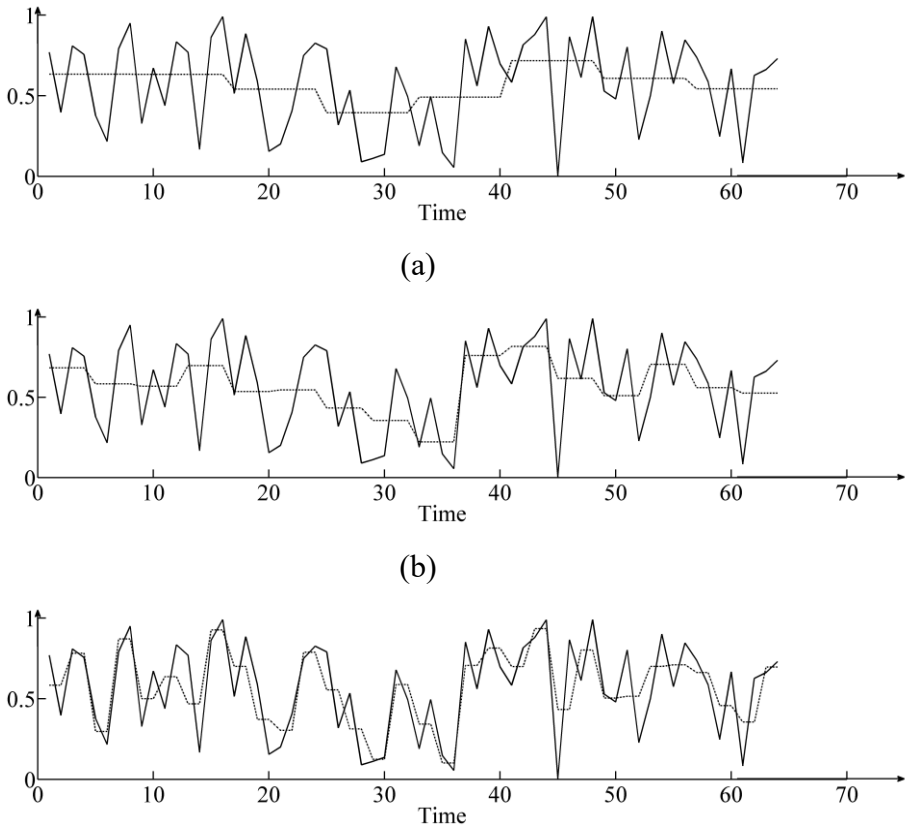


Figure 2.3 Time series reconstruction with the use of the different number of wavelet coefficients. Solid line: input time series; Dotted line: the result of reconstruction. (a) m=8; (b) m=16; (c) m=32;

With regard to dimensionality reduction, the basic idea of DWT is similar to that of DFT. The first few wavelet coefficients can archive coarse approximation of the original

time series. However, being different from the DFT time series representation method that only considers frequencies, DWT focuses on the analysis completed both on time and frequency domain of the time series. In this sense, during the reconstruction process, wavelet coefficients can incorporate location contributions. Note that a major drawback of this method is associated with the length of time series that must be a power of 2. In general, padding with zeros is a practice to ensure its feasibility.

2.2.5 Discrete Cosine Transformation (DCT)

Discrete Cosine Transformation [94, 95] comes as one of the spectral methods and is concerned with a linear transformation, which can serve here as a vehicle to provide dimensionality reduction of time series [96]. The underlying idea is closely related with that of DFT. Only a few coefficients of the DCT representation can represent the original time series adequately and are able to reconstruct it accurately; note that only the cosine function is regarded as the basis function. In [96], the authors compared different ways of time series approximation and pointed out that its better performance is achieved when the input data are highly correlated.

2.3 Time series modeling

Fuzzy systems as an efficient tool of time series models describing vague and imprecise information by means of linguistic variables, fuzzy relations and fuzzy logic. A set of IF-THEN rules can be extracted to capture the qualitative behavior of time series, which is similar to the experience and intuition of human beings. Therefore, the time series prediction by using fuzzy systems has being well-developed and well-documented in the literature [97-100]. For instance, In the fuzzy systems [101], its identification of the premise part has been performed via a combination of island model parallel genetic algorithm and a space search memetic technique while the corresponding consequence part is accomplished by using an improved QR Householder least-squares method. In [102], the authors have introduced a novel evolving the fuzzy system to model time-varying and real-time industrial automation systems. Different fuzzy rules are derived from the various sets of the effective input variables because selecting inputs locally at the rule level is essential. In the proposed evolving heterogeneous fuzzy inference systems (eHFIS), the premise and consequence parameters learning have been archived by using an incremental

evolving clustering and a weighted recursive fuzzily weighted least-squares estimator, respectively. In [103], a correntropy-based evolving fuzzy neural system (CEFNS) has been proposed for time series prediction under both noise-free and noisy conditions. The maximum correntropy criterion is used to tune the parameters of this system because of its better outlier rejection ability than that of mean-square error criterion. One can refer to [103, 104] for more information about the evolving fuzzy systems (EFS) which can self-adapt to the structures and parameters of the proposed fuzzy systems. In [105], the authors proposed the process Takagi-Sugeno (PTS) model to perform time series prediction. In contrast to the standard version, the continuous functions characterize the input and output of the PTS model. The antecedent and consequent parameters are identified by the relational clustering algorithm and the least square method in the function vector space, respectively. All in all, the process of building the fuzzy rules-based systems consists of two parts: (i) structural leaning (e.g. the determination of the number of rules, the partition of the universe of discourse); (2) parametric tuning (e.g. the tuning of parameters in the antecedent and consequent parts). The selection of techniques or algorithms for the construction of antecedent and consequent parts is important in the design process of fuzzy systems. Several common encountered techniques or methods used in time series prediction are summarized briefly in 错误!未找到引用源。.

Table 2-1 A brief collection of design strategies and optimization techniques – selected examples

Construction of antecedent (or premier/condition) parts of the rules	Construction of consequent (or conclusion) parts of the rules	References
A modified adaptive spline modeling (MASMOD)	Expectation-maximization (EM) algorithm	[106]
Gaussian membership functions for fuzzy partition of the input space	Wavelet neural network	[107]
Fuzzy C-means (FCM) clustering method	Wavelet functions and the least square method	[108]
Complex fuzzy sets (CFSs)	Autoregressive integrated moving average (ARIMA)	[109]

H^{∞} estimation theory	Least squares estimation	[110]
Premise parts are static/fixed or from experts	Recurrent neural networks	[111]
The backpropagation algorithm	The backpropagation algorithm	[112]
Random hidden-layer structure	Fast sparse coding identification method	[113]
A novel clustering algorithm with a new validation criterion	Orthogonal least square (OLS) method	[114]
Levenberg–Marquardt (LM) optimization method	Levenberg–Marquardt (LM) optimization method	[115]
Relational clustering algorithm	Least square method	[105]
A modified differential harmony search (MDHS) technique	A modified differential harmony search (MDHS) technique	[116]
Fuzzy C-Means (FCM) clustering method	Recurrent neural networks	[117]
Fuzzy clustering method	Recursive least square algorithm	[118]
Heuristic optimization approaches (e.g. genetic algorithm and artificial bee colony)	Extreme learning machine	[119]

By and large, there are two different categories for the design of fuzzy systems [39, 120, 121]: (i) global learning and (ii) local learning. Irrespectively from the diversity of approaches, fuzzy rule-based models share a visible commonality: complex phenomena are modeled locally through a series of local models (which are less complicated than a single global model). Their differences are mainly due to how to construct the local models, how to aggregate the formed rules and where the rules are extracted. Each of these two phases comes with various augmentations. In **错误!未找到引用源。**, we offer a highlight of the visible representatives of fuzzy models, their design strategies and

optimization tools being used. One can refer to [99, 121] for more details about design categories and optimization tools of the fuzzy systems.

Table 2-2 A collection of design strategies and optimization tools of selected examples

Construction of condition part of the rules	Construction of conclusion part of the rules	References
Hybrid algorithm (genetic algorithms and complex method)	Least square error method	[122-124]
Resilient propagation (RPROP) Original heuristic search	Standard gradient descent (back propagation) Resilient propagation (RPROP)	[125-127]
Genetic algorithm	Recursive least squares approach	[128-130]
Fuzzy clustering (K-Means, the Gath–Geva algorithm and the Gustafson–Kessel algorithm)	Weighted recursive least squares algorithm with forgetting factor	[131, 132]
Fuzzy C-Means (FCM) clustering	Weighted Least Square Estimation (WLSE) method	[133, 134]
Fuzzy C-Means (FCM) clustering algorithm modified fuzzy c-regressive model clustering algorithm (NFCRMA)	Orthogonal least square (OLS)	[135, 136]
Gravitational search (GSA)-based hyper-plane clustering algorithm (GSHPC)	Orthogonal least square (OLS)	[137-139]
Hard C-means clustering method Genetic algorithms (GAs)	Least square error method	[123, 140]
Modified fuzzy c-regression model (FCRM)	Orthogonal least squares	[141-143]
Cluster estimation method	Least mean squares estimation	[144, 145]
Back propagation learning rule	Least square method	[146, 147]

Extended vector quantization	Recursive weighted least-squares approach	[148-150]
A new projection concept	Enhanced recursive least square method	[151-154]
Iterative vector quantization algorithm	Regularized sparsity-constrained-optimization	[155-157]
Subtractive clustering method	Linear least squares estimation	[145, 158]

Hidden Markov Model (HMM) is sought as one of the temporal models to portray dynamic behavior of time series. Recently, a series of studies try to apply HMM in the fuzzy domain. For instance, in [159], based on the HMM, the authors proposed a new time series forecasting model, which also dwells upon fuzzy relations and fuzzy time series. The role of the Monte Carlo method is to estimate the final prediction result. In [160], on the basis of the above paper, in order to deal with the zero-probability problem which may deteriorate the quality of the forecasting accuracy, fuzzy smoothing was introduced to cope with the fuzziness of HMM-based fuzzy time series. In [161-163], HMM were also applied to these novel hybrid time series prediction models. However, the evident role of HMM is to identify similar data patterns through their HMM-log-likelihood values. In order to construct the connection between the information of fuzzy sets and the state concept, authors improve the traditional HMM to fuzzy HMMs (FHMMs) through using fuzzy measures and fuzzy integrals in [164, 165]. One can refer [161, 163, 166] to for more details about the HMM in the fuzzy realm.

Chapter 3

Multivariate Time series Anomaly Detection: A Framework of Hidden Markov Models[a]

In this chapter, we start to elaborate a supervised approach to multivariate time series anomaly detection focused on the transformation of multivariate time series to univariate time series. Several transformation techniques involving Fuzzy C-Means (FCM) clustering and fuzzy integral are studied. In the sequel, a Hidden Markov Model (HMM), one of the commonly encountered statistical methods, is engaged here to detect anomalies in multivariate time series. We construct HMM-based anomaly detectors and in this context compare several transformation methods.

3.1 Problem Formulation

Let us assume a multivariate time series $\boldsymbol{X} = \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{T+T'}$ of length $T + T'$. T and T' are the lengths of training and testing time series, respectively. After applying *Z*-score normalization [167], we run FCM, and then determine both Sugeno integral and Choquest integral to produce training (and testing) (observed) state sequences $\boldsymbol{O} = o_1, o_2, \ldots, o_T$ (and $\boldsymbol{O'} = o'_1, o'_2, \ldots, o'_{T'}$). In this study, we consider the cluster prototypes obtained by FCM as the states in the sequences. Compared with the FCM, an additional step, namely mapping/vector discretion (from continuous to discrete), is necessary to produce state sequences and construct fuzzy integral based detectors.

In the construction of the HMM, we use a labeled training state sequence coming in the form of data-label pairs (o_k, i_k), $k = 1, 2, \ldots, T$ where o_k is one-dimensional observed state and k_k stands for its label (normal or abnormal). The temporally ordered labels are regarded as a hidden state sequence of the HMM.

Considering the time series anomaly detection problem, for HMM, the number of hidden states is equal to 2, which correspond to the normal or abnormal state. Thus, the

initial vector $\boldsymbol{\pi}$, state transition matrix $\boldsymbol{A}$, and emission matrix $\boldsymbol{B}$ are expressed as follows.

$$\boldsymbol{\pi} = \begin{bmatrix} \pi_1(normal) & \pi_2(abnormal) \end{bmatrix} \tag{3.1}$$

$$\boldsymbol{A} = \begin{bmatrix} a_{11}(normal \to normal) & a_{12}(normal \to abnormal) \\ a_{21}(abnormal \to normal) & a_{22}(abnormal \to abnormal) \end{bmatrix} \tag{3.2}$$

$$\boldsymbol{B} = \begin{bmatrix} b_{11} & \cdots & b_{1L} \\ b_{21} & \cdots & b_{2L} \end{bmatrix} \tag{3.3}$$

When we consider q and o as the hidden state and visible state respectively, we calculate the state transition matrix and the emission as follows.

$$\pi_i = \frac{|q_i|}{\sum q} \tag{3.4}$$

$$a_{ij} = \frac{|q_{ij}|}{\sum_{j=1}^{N} |q_{ij}|} \tag{3.5}$$

$$b_{ik} = \frac{|o_{il}|}{\sum_{l=1}^{M} |o_{il}|} \tag{3.6}$$

Once the parameters of HMM have been determined, the Viterbi algorithm is then used to determine the label of testing observed state sequence $\boldsymbol{O'} = o'_1, o'_2, \ldots, o'_{T'}$. The most likely state sequence is $\boldsymbol{I'} = i'_1, i'_2, \ldots, i'_{T'}$. Let us highlight the essence of the proposed methods as shown in Figure 3.1; we point at the two key methodological steps encountered there, namely a transformation from multivariate time series to univariate time series followed by HMM-based detection. After application of the z-score normalization, we invoke different transformation methods, namely FCM, Sugeno integral and Choquet integral, which implement the transformation. Subsequently, we estimate the essential parameters of the HMM by using the labels of the training set. More specifically, the collected normal and abnormal time points are considered to estimate the emission and transition probabilities of the HMM. After training the HMM, we apply the Viterbi algorithm to test the observed state sequence and compute the most likely hidden state sequence consisting of the two states (normal and abnormal).

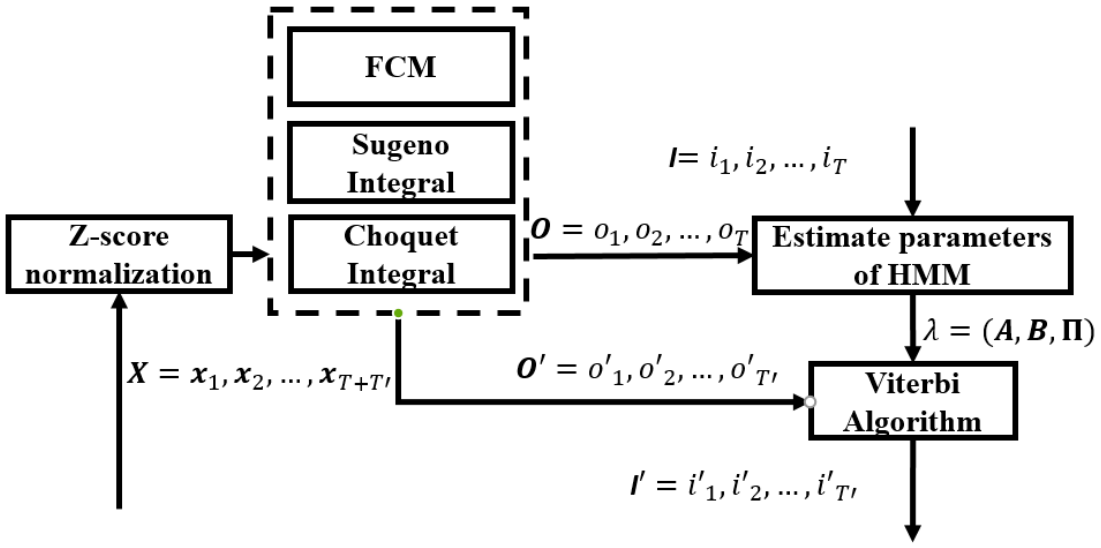


Figure 3.1 Overall processing realized by the anomaly detector

3.2 Hidden Markov Model

HMM can cope with time series that are generated by a certain Markov process. Two essential assumptions are made: (i) only the current states affect the next state, (ii), transition probabilities between the states do not vary over time (stationarity requirement). In particular, for each HMM, there are hidden/observed state sets and three probability matrices. Each hidden state emits one of the states that can be directly observed. The hidden state set $Q = \{q_1, q_2, ..., q_N\}$ comprises of N possible hidden states and the observed state set $R = \{r_1, r_2, ..., r_L\}$ consists of L possible observed states.

Let us assume an observed state sequence coming in the form $O = \{o_1, o_2, ..., o_T\}$. To gain a clear understanding of the HMM, assume $I = \{i_1, i_2, ..., i_T\}$ is the corresponding hidden state sequence of the above observed state sequence. For each HMM, it can be defined as follows.

$$\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}) \tag{3.7}$$

Where $\boldsymbol{A}$ and $\boldsymbol{B}$ denote the state-transition matrix and the emission matrix respectively which can be represented as follows.

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{bmatrix}$$

$$\boldsymbol{B} = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1L} \\ b_{21} & b_{22} & \dots & b_{2L} \\ \vdots & \vdots & \vdots & \vdots \\ b_{N1} & b_{N2} & \dots & b_{NL} \end{bmatrix}$$

$$\boldsymbol{\pi} = \begin{pmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_N \end{pmatrix}$$

Here $a_{ij} = P(i_{t+1} = q_j | i_t = q_i)$ stands for the probability from the state q_i (at the t^{th} time point) to the state q_j (at the $t + 1^{\text{th}}$ time point). And $b_{ik} = P(o_t = s_k | i_t = q_i)$ means the probability from the state q_i to the observation s_k at the same time point. $\pi = P(i_1 = q_i)$ characterizes the probability of the initial state with q_i.

In general, HMM deals with the three standard problems that arise in various applications:

- Given an HMM model, $\Psi = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$ and an observed sequence $O = o_1, o_2, \dots, o_T$, calculate the probability $P(\boldsymbol{O}|\lambda)$ that the observed sequence has been produced by this HMM $\Psi = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$.
- Given an observed sequence $\boldsymbol{O} = o_1, o_2, \dots, o_T$, estimate the parameters of the HMM model$\Psi = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$ that maximize the probability $P(\boldsymbol{O}|\Psi)$ of observations given the model.
- Given an HMM model $\Psi = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$ and an observed sequence $\boldsymbol{O} = o_1, o_2, \dots, o_T$, decide the most likely state sequence $\boldsymbol{I}$.

The Viterbi algorithm, realizing an algorithm of dynamic programming algorithm, estimates the most probable state sequence [168]. It can determine the optimal hidden state sequence $I = i_1, i_2, \dots, i_T$ based on the HMM model $\Psi = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$ and the given observed state sequence $\boldsymbol{O} = o_1, o_2, \dots, o_T$.

Consider that $\delta_t(i)$ stands for the probability of state i at i^{th} time moment defined as follows

$$\delta_t(i) = \max_{i_1, i_2, \dots, i_{t-1}} P(i_t = i, i_{t-1}, \dots, i_1, o_t, o_{t-1}, \dots, o_1 | \lambda) = \max_{1 \le j \le N} \left(\delta_t(j) a_{ji}\right) b_{io_{t+1}}$$

The detailed algorithm comes as the following sequence of steps.

Initialization:

$$\delta_1(i) = \pi_i b_{io_1} \tag{3.8}$$

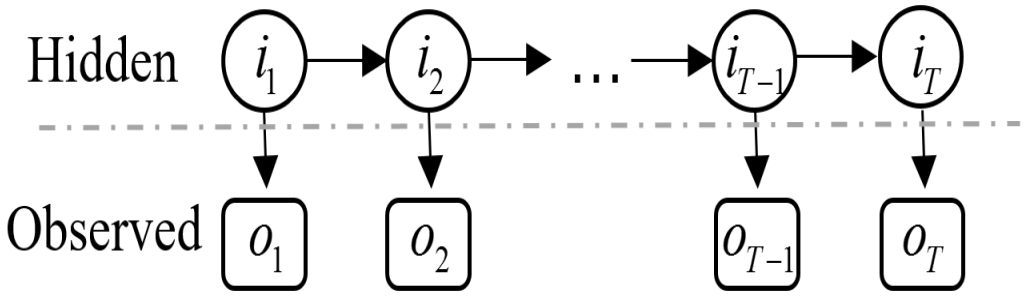
Recursion:

$$\begin{aligned}\delta_{t+1}(i) &= \max_{i_1,i_2,\ldots,i_t} P(i_{t+1}=i, i_t,\ldots,i_1, o_{t+1}, o_t,\ldots,o_1 \mid \lambda) \\ &= \max_{1\le j\le N}\left(\delta_t(j)\alpha_{ji}\right) b_{io_{t+1}}\end{aligned} \tag{3.9}$$

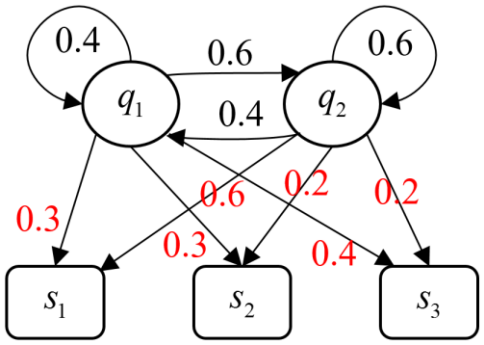
Termination:

$$P^* = \max_{1\le j\le N} \delta_T(i) \tag{3.10}$$

As an illustrative example, Figure 3.2 shows a simple example of the HMM when the numbers of hidden states and observed states are 2 and 3, respectively.



(a)



(b)

Figure 3.2 Illustrative example of HMM (red: emission probabilities; black: transition probabilities)

3.3 Multivariate time series transformation methods

In practice, the values of the multivariate time series are collected using different sensors. There are a number of information-retaining methods for transforming multivariate time series into an observed sequence. Here, we focus on FCM clustering methods and fuzzy integral methods, which were found useful in many applications

3.3.1 FCM Algorithm

A sound alternative to transform a multivariate time series to an observed sequence is to use the FCM clustering algorithm [169, 170]. Given a multivariate time series $\boldsymbol{X} = \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T$ of length T, the objective function Q used in the FCM is defined in the following way

$$Q = \sum_{i=1}^{c} \sum_{j=1}^{T} u_{ij}^{m} d^2 \left(\boldsymbol{x}_{j,} \boldsymbol{v}_i \right) \tag{3.11}$$

Here c stands for the number of clusters and $m(m > 1)$ denotes the fuzzification coefficient. $U = [u_{ij}]$ and $\boldsymbol{v}_i$ are the partition matrix and the i^{th} prototype, respectively. $d^2 \left(\boldsymbol{x}_{j,} \boldsymbol{v}_i \right)$ (as well as $\|.\|^2$) stands for the Euclidean distance (or its generalization) between $\boldsymbol{x}_{j,}$ and the prototype $\boldsymbol{v}_i$. The partition matrix and cluster centers (prototypes) are calculated iteratively as follows

$$\boldsymbol{v}_i = \frac{\sum_{j=1}^{T} u_{ij}^{m} \boldsymbol{x}_j}{\sum_{j=1}^{T} u_{ij}^{m}} \tag{3.12}$$

$$u_{ij} = \frac{1}{\sum_{l=1}^{c} \left(\frac{\| \boldsymbol{v}_i - \boldsymbol{x}_j \|}{\| \boldsymbol{v}_l - \boldsymbol{x}_j \|} \right)^{2/(m-1)}} \tag{3.13}$$

The sequence of iterations is carried out to realize the minimization of the objective function. Then, on a basis of the partition matrix generated by the FCM, each $\boldsymbol{x}_j$ belongs to the cluster to which it exhibits the highest membership degree.

3.3.2 Fuzzy measures and fuzzy integrals

Fuzzy integrals can combine different sources of uncertain information [171] and have been widely applied to a variety of fields, such as decision making [172], pattern recognition [173],supplier evaluation [174], gaze control of robotics [175], etc. [176]. Fuzzy integral is calculated with respect to a fuzzy measure that can capture the relationship among different variables. Let us recall that by a fuzzy measure we mean a set function g that satisfies the following set of conditions

Boundary conditions:

$$g(\varnothing) = 0 \quad g(\boldsymbol{X}) = 1 \tag{3.14}$$

Monotonicity:

$$if \quad A \subset B\left(A, B \in g(X)\right), \quad then \quad g(A) < g(B) \tag{3.15}$$

Continuity:

If $\{A_n\}, (1 \leq n \leq \infty)$ is a monotone sequence of measurable sets, then

$$\lim_{n\to\infty} g(A_n) = g\left(\lim_{n\to\infty} A_n\right) \tag{3.16}$$

Based on the above definition, Sugeno developed a certain type of fuzzy measures, namely λ-fuzzy measure [177]. Here the union of two disjoint sets A and B is determined as follows.

$$g(A \cup B) = g(A) + g(B) + \lambda g(A) g(B) \tag{3.17}$$

Based on the normalization condition, the parameter of λ describes a level of interaction between the two disjoint sets and is greater than -1. The determination of its value comes as a result of the solution to the following polynomial equation

$$\lambda + 1 = \prod_{i=1}^{n} (1 + \lambda g_i) \quad \lambda > -1 \tag{3.18}$$

where λ models several types of interaction: excitatory for its positive values, inhibitory for the negative values. The fuzzy measure is additive (no interaction) when $\lambda = 0$. In what follows, we recall a concept of the fuzzy integrals.

Sugeno fuzzy integral

Let g be a fuzzy measure. Let h be a function: $X \to [0,1]$. The Sugeno fuzzy integral of h with respect to the fuzzy measure g is calculated in the following form.

$$\int_A h(x) \circ \qquad \underset{\alpha \in [0,1]}{} \left[\min\left(\alpha, g(A \cap H_\alpha)\right)\right] \tag{3.19}$$

Where $H_\alpha = \{x | h(x) \geq \alpha\}$ is an α-cut of this function.

Choquet fuzzy integral

Let g be a fuzzy measure. As before h: $X \to [0,1]$. The Choquet fuzzy integral of h with respect to g is expressed in the following form

$$\int_A h(x) \circ \qquad \sum_{i=1} \left[h(x_i) - h(x_{i-1})\right] g(A_i) \tag{3.20}$$

Here $g(A_i) = g_i + g(A_i) + \lambda g_i g(A_i)$.

For Sugeno Integral and Choquet Integral determined with respect to the λ-fuzzy measure, the calculation of integral only requires information about fuzzy density [178] g_i. Higher values of g_i indicate that the i^{th} feature is increasingly essential. As an

illustrative example, we consider a single multivariate time series involving three variables reported at a certain time moment and recording measurement values of three sensors. Here the quality of information from each sensor can be regarded as the value of the fuzzy density. Higher values of the fuzzy density g_i indicate more essential entries at this time moment. The values of the multivariate time series are arranged in a vector form

$$h = [0.7 \quad 0.4 \quad 0.3]$$

The corresponding vector of the fuzzy densities assumes the following entries (those values can be estimated by experts or derived on a basis of some training data).

$$g = [0.21 \quad 0.35 \quad 0.05]$$

Following the above definitions, the results of Sugeno fuzzy integral and Choquet fuzzy integral are equal to 0.4400 and 0.7889, respectively.

3.4 Experimental Studies

In this section, we report on a series of numeric examples illustrating how the amplitude anomalies in multivariate time series are detected. Both synthetic data and the publicly available datasets with artificial anomalies are considered.

3.4.1 Synthetic data

The multivariate time series is generated in the form of sine and cosine functions of different frequencies, see Figure 3.3. The length of the series is equal to 2,000 samples and there are some visible changes at different time points of each variable of multivariate time series. Gaussian noise (with the zero mean and unit standard deviation) is added to each variable of the original multivariate time series to increase the difficulty of detecting the anomalies and make the data more realistic. These artificial anomalies are generated by randomly picking some time points and increasing their amplitude by multiplying them by a random value located in the interval [0, 3].

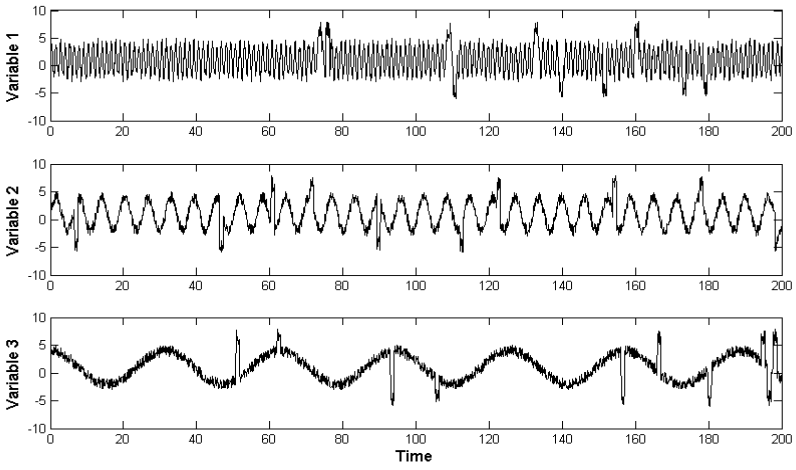


Figure 3.3 Synthetic multivariate time series

Two data sets, one covering time points from 1 to 140 (treated as a training set) and another one covering time points from 140.1 to 200 (testing set), have been considered in this experiment. For comparison, PCA is also exploited to transform a multivariate time series to an observed sequence. As the first component of the PCA transformed data captures the most information about the data [179], it would be possible to use only this component (the one with the highest eigenvalue) as a new 'combined' sequence, obtaining a transformation from multivariate time series to univariate time series.

To cluster the multivariate time series, there are two essential parameters of the FCM, namely a fuzzification coefficient and the number of clusters. Here we vary the values of the fuzzification coefficients ranging from 1.1 to 2.9 with a step of 0.1 while the number of clusters is taken from 2 to 198. For discretization, different values of the number of observed states located in the range [2, 80] have been considered leading to the optimal value of this parameter.

Figure 3.4 displays the experimental results produced by different methods. To make results more readable, for each detector, its objective (or quantitative) evaluation on this dataset have been reported to evaluate its performance. When *TP, FP, TN* and *FN* are the number of normal time points correctly detected as normal (True Positives), the number of abnormal time points that are detected as normal (False Positives), the number of abnormal time points that are detected as abnormal (True Negatives) and the number of normal time points that are detected as abnormal (False Negatives), Accuracy, sensitivity, specificity

and F-measure are defined as the following expressions, where are the objective (or quantitative) evaluation included in our experiments. Table 3-1 displays the confusion matrices produced by different methods

$$Accuracy = \frac{TN + TP}{TN + FP + FN + TP} \tag{3.21}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{3.22}$$

$$Specificity = \frac{TN}{TN + FP} \tag{3.23}$$

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3.24}$$

Where

$$Precision = \frac{TP}{TP + FP} \quad and \quad Recall = \frac{TP}{TP + FN} \tag{3.25}$$

Table 3-1 Confusion matrix produced by different methods

PCA+HMM			FCM+HMM		
	p	n		p	n
Y	459	90	Y	464	94
N	16	35	N	11	31
Sugeno Integral+HMM			Choquet Integral+HMM		
	p	n		p	n
Y	473	65	Y	426	53
N	2	60	N	49	72

As shown in Table 3-2, the proposed FCM+HMM-based detector has achieved higher accuracy in comparison with the accuracy obtained when using other detectors. The accuracy improvement of FCM, Sugeno fuzzy integral, Choquet fuzzy integral based detectors vis-a-vis the generic PCA-based detector is in the range 7-9 %.

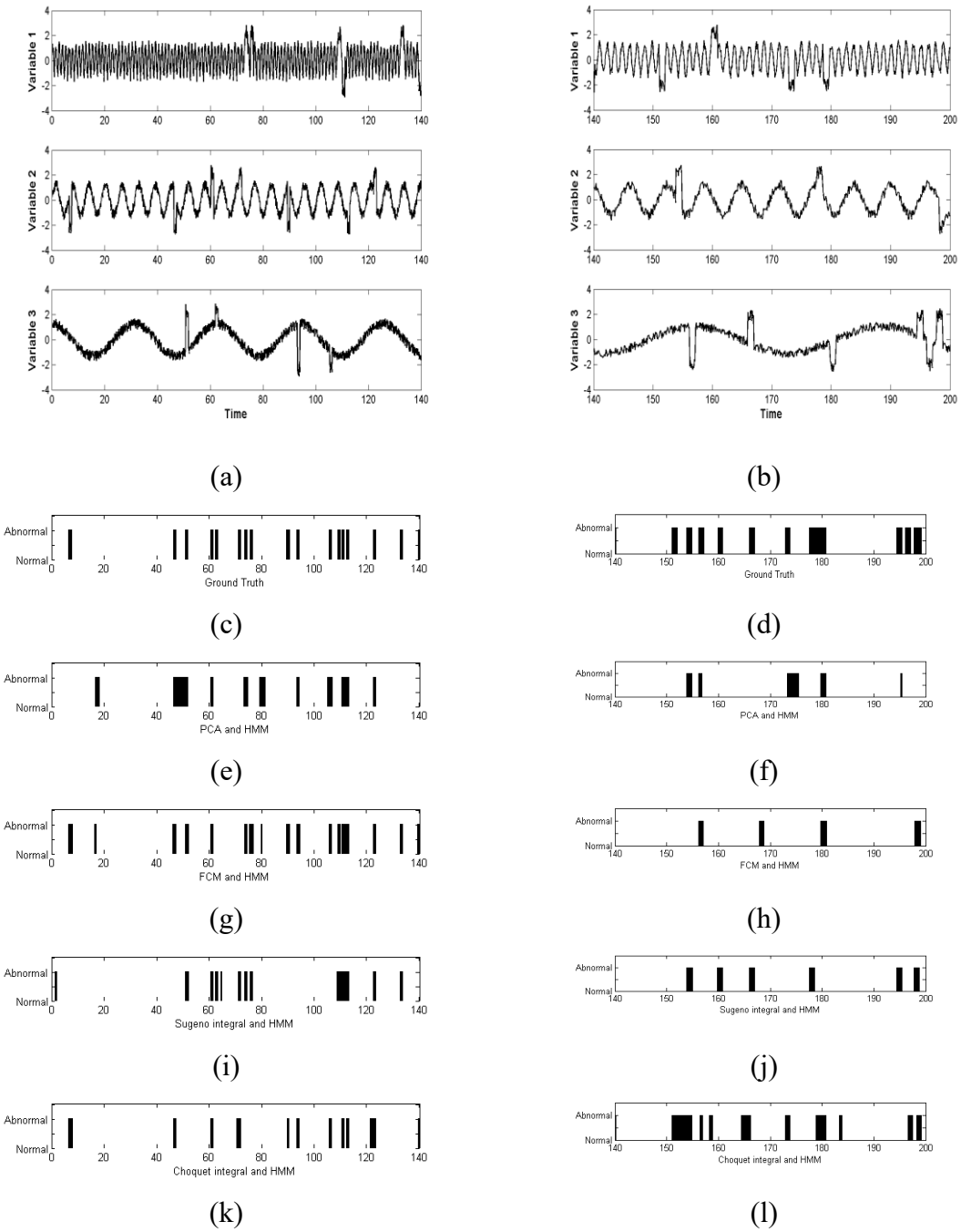


Figure 3.4 Synthetic multivariate time series: (a) training set, (b) testing set, (c) Ground truth of training set, (d) Ground truth of testing set, (e) Experimental results of PCA + HMM (training set), (f) Experimental results of PCA+HMM (testing set), (g) Experimental results of FCM + HMM (training set), (h) Experimental results of FCM + HMM (testing set), (i) Experimental results of Sugeno integral + HMM (training set), (j) Experimental results of Sugeno integral + HMM (testing set), (k) Experimental results of

Choquet integral + HMM (training set), (l) Experimental results of Choquet integral + HMM (testing set).

Table 3-2 Experimental results obtained for synthetic multivariate time series

Training Set	Accuracy	F-measure	Sensitivity	Specificity
PCA+HMM	0.8764	0.9294	0.9245	0.5266
FCM+HMM	0.9550	0.9742	0.9651	0.8817
Sugeno_Integral+HMM	0.9350	0.9634	0.9740	0.6509
Choquet_Integral+HMM	0.9393	0.966	0.9838	0.6154
Testing Set	Accuracy	*F*-measure	Sensitivity	Specificity
PCA+HMM	0.8233	0.8964	0.9663	0.2800
FCM+HMM	0.8250	0.8984	0.9768	0.2480
Sugeno_Integral+HMM	0.8883	0.9338	0.9958	0.4800
Choquet_Integral+HMM	0.8300	0.893	0.8968	0.5760

To quantify the obtained optimal number of clusters and the value of the fuzzification coefficient, Figure 3.5 shows the corresponding accuracy when considering different values of these parameters. It is evident that the increase of the number of clusters will affect the performance of the detector significantly. The fuzzification coefficient exhibits some impact on the performance of the detector. Note that here HMM would fail due to the unknown external observed states that do not appear in training set. In other words, for adding new observations, re-training different HMM for new observations is anticipated.

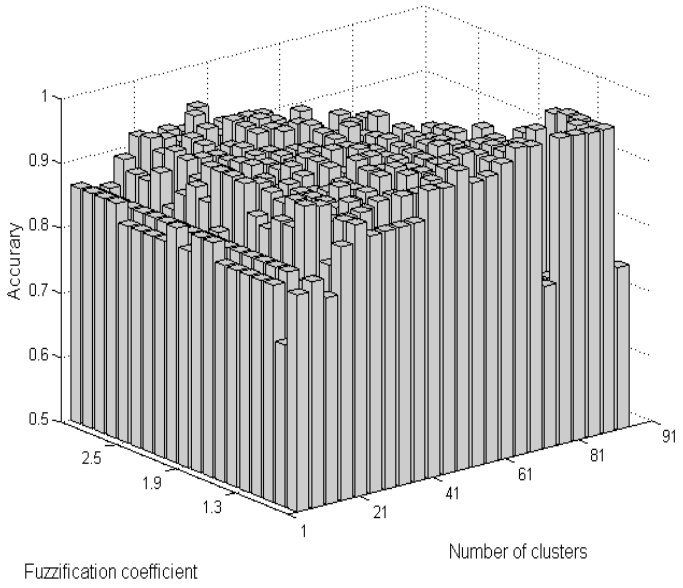


Figure 3.5 Performance comparison reported for various values of the fuzzification coefficient and the number of clusters

3.4.2 Publicly available datasets

In this subsection, we report on a variety of experiments on real-world multivariate time series from different repositories such as UCI machine learning repository and DataMarket. The parameter setting is in the same way as presented for the synthetic data.

Data Set #1 [U.S. Dollar Exchange Rate]: The historical intraday data (per day except for holidays and regular weekends) for three currencies (the US dollar exchange rate versus the Dutch guilder, the French franc and the German mark) in the period January 03, 1989 to December 31, 1998: 1) Dutch guilder (NLG); 2) French franc (FF); 3) German mark (DEM).

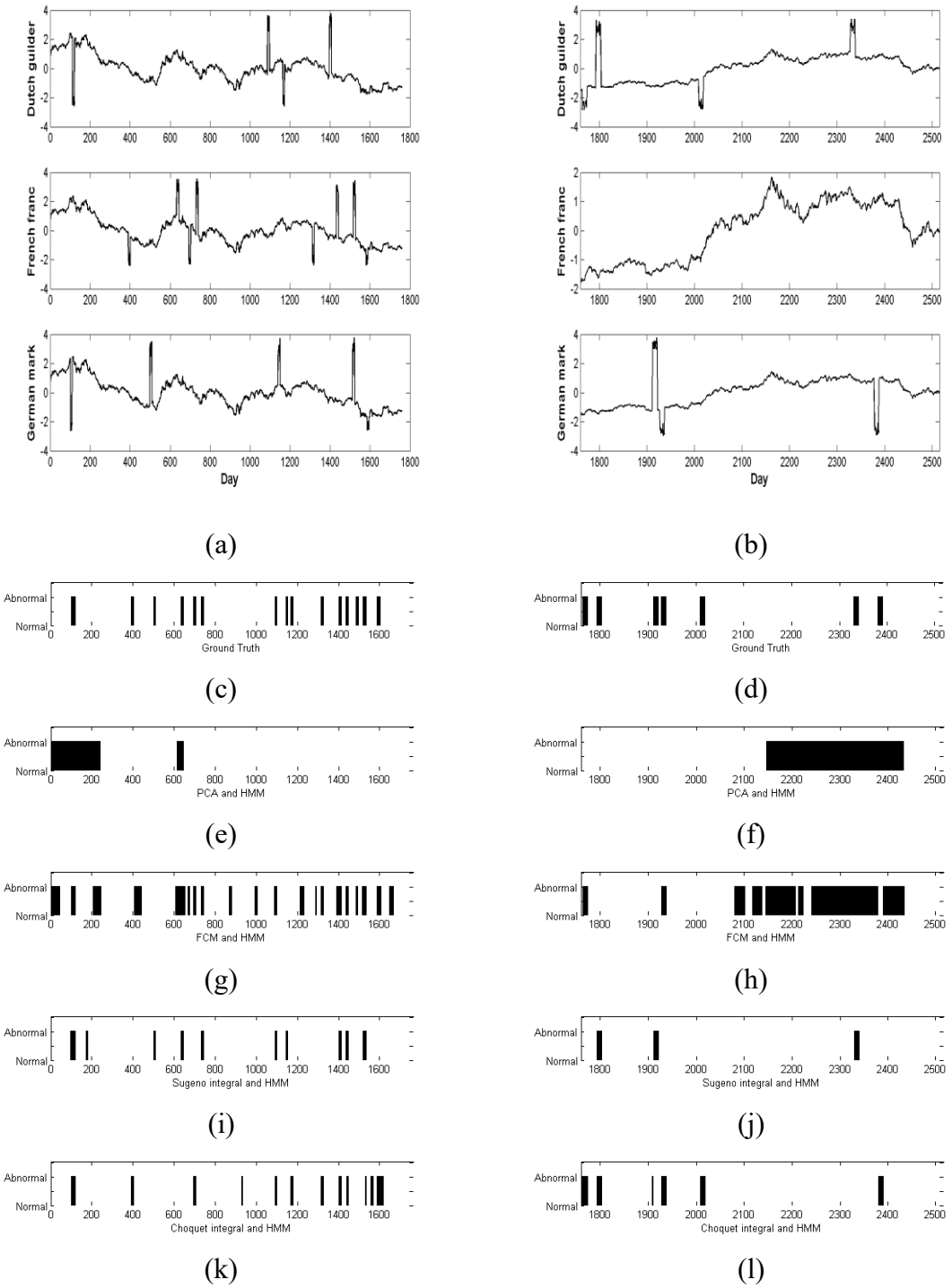


Figure 3.6 U.S. Dollar Exchange Rate Dataset: (a) training set, (b) test set, (c) Ground truth of training set, (d) Ground truth of testing set, (e) Experimental results of PCA + HMM (training set), (f) Experimental results of PCA+HMM (testing set), (g) Experimental results of FCM + HMM (training set), (h) Experimental results of FCM

+ HMM (testing set), (i) Experimental results of Sugeno integral + HMM (training set), (j) Experimental results of Sugeno integral + HMM (testing set), (k) Experimental results for Choquet integral + HMM (training set), (l) Experimental results of Choquet integral + HMM (testing set).

Table 3-3 Experimental results of U.S. Dollar Exchange Rate Dataset

Training Set	Accuracy	Sensitivity	Specificity	F-measure
PCA+HMM	0.7864	0.8516	0.1765	0.878
FCM+HMM	0.8386	0.8478	0.7529	0.9046
Sugeno_Integral+HMM	0.9523	0.9899	0.6000	0.974
Choquet_Integral+HMM	0.9483	0.9818	0.6353	0.9716
Testing Set	Accuracy	Sensitivity	Specificity	F-measure
PCA+HMM	0.5828	0.6131	0.2857	0.7272
FCM+HMM	0.5748	0.5898	0.4286	0.7156
Sugeno_Integral+HMM	0.9470	1.0000	0.4286	0.9716
Choquet_Integral+HMM	0.9669	0.9927	0.7143	0.982

Data Set #2 [EEG Eye State Dataset]: Three major EEG (eletroencephalogram) measurements (at a sampling frequency of 128 samples per second) acquired using the Emotiv EEG Neuroheadset: 1) AF3 (Intermediate between Fp and F); 2) F7 (Frontal left Hemisphere); 3) FC5 (Between F and C left Hemisphere).

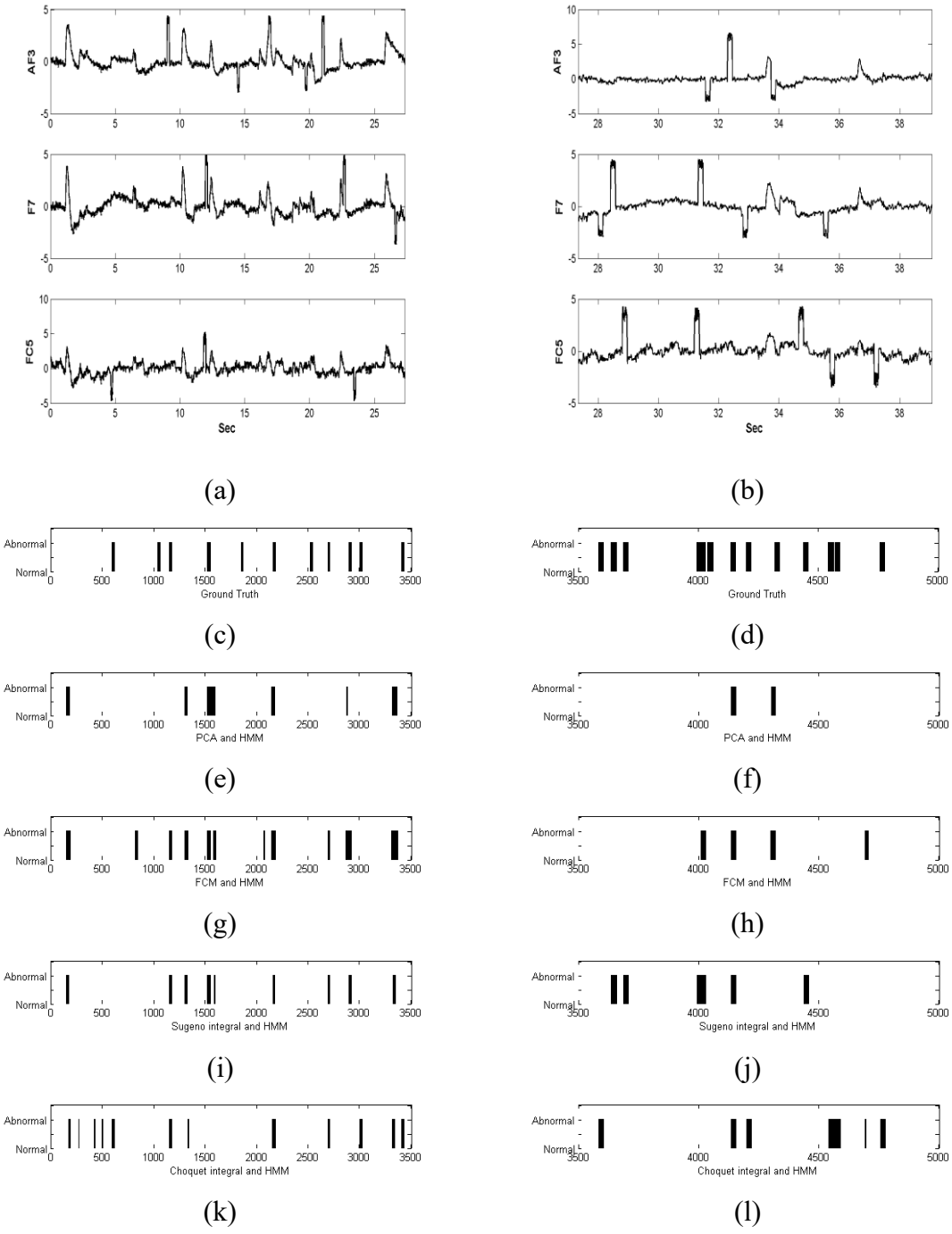


Figure 3.7 EEG Eye State Dataset: (a) training set, (b) test set, (c) Ground truth of training set, (d) Ground truth of testing set, (e) Experimental results of PCA + HMM (training set), (f) Experimental results of PCA+HMM (testing set), (g) Experimental results of FCM + HMM (training set), (h) Experimental results of FCM + HMM (testing set), (i) Experimental results of Sugeno integral + HMM (training set), (j)

Experimental results of Sugeno integral + HMM (testing set), (k) Experimental results of Choquet integral + HMM (training set), (l) Experimental results of Choquet integral + HMM (testing set).

Table 3-4 Experimental results of EEG Eye State Dataset

Training Set	Accuracy	Sensitivity	Specificity	F-measure
PCA+HMM	0.8997	0.9488	0.2203	0.9464
FCM+HMM	0.9003	0.9305	0.4831	0.9456
Sugeno_Integral+HMM	0.9454	0.9795	0.4746	0.971
Choquet_Integral+HMM	0.9477	0.9822	0.4703	0.9722
Testing Set	Accuracy	Sensitivity	Specificity	F-measure
PCA+HMM	0.8320	0.9879	0.0742	0.907
FCM+HMM	0.8340	0.9735	0.1563	0.9068
Sugeno_Integral+HMM	0.9067	1.0000	0.4531	0.9468
Choquet_Integral+HMM	0.9013	0.9904	0.4688	0.9434

Data Set #3 [Air Quality Dataset]: Three major chemical sensors (related to hourly average concentrations for Total Nitrogen Oxide, Nitrogen Dioxide and Ozone) produced by an Air Quality Chemical Multi-sensor Device that placed in a polluted field of an Italian city in the period March 10, 2004 to June 2, 2004: 1) PT08S3 (NOx); 2) PT08S4 (NO2); 3) PT08S5 (O3).

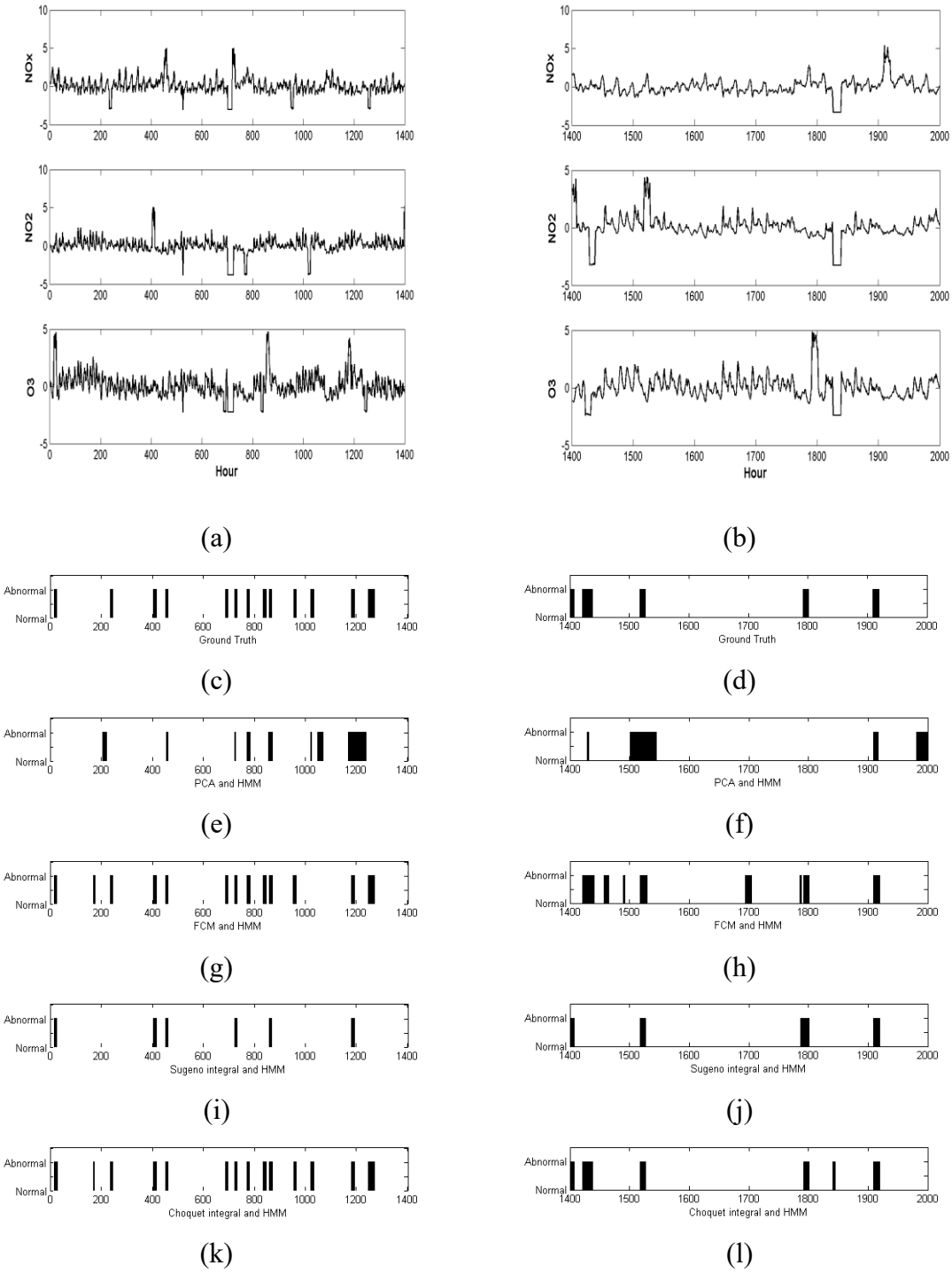


Figure 3.8 Air Quality Dataset: (a) training set, (b) test set, (c) Ground truth of training set, (d) Ground truth of testing set, (e) Experimental results of PCA + HMM (training set), (f) Experimental results of PCA+HMM (testing set), (g) Experimental results of FCM + HMM (training set), (h) Experimental results of FCM + HMM (testing set), (i)

Experimental results of Sugeno integral + HMM (training set), (j) Experimental results of Sugeno integral + HMM (testing set), (k) Experimental results of Choquet integral + HMM (training set), (l) Experimental results of Choquet integral + HMM (testing set).

Table 3-5 Experimental results obtained for Air Quality Dataset

Training Set	Accuracy	Sensitivity	Specificity	F-measure
PCA+HMM	0.8586	0.9220	0.3007	0.9214
FCM+HMM	0.9807	0.9865	0.9301	0.9892
Sugeno_Integral+HMM	0.9429	1.0000	0.4406	0.9692
Choquet_Integral+HMM	0.9971	0.9968	1.0000	0.9984
Testing Set	Accuracy	Sensitivity	Specificity	F-measure
PCA+HMM	0.8567	0.9048	0.3704	0.92
FCM+HMM	0.9367	0.9469	0.8333	0.9646
Sugeno_Integral+HMM	0.9633	0.9908	0.6852	0.98
Choquet_Integral+HMM	0.9917	0.9908	1.0000	0.9954

As illustrated in the figures and tables, since fuzzy integral and FCM has been used to combine multivariate time series, the performance improvement of the corresponding detectors is quite apparent. This is related to the fact that more useful information is contained in the transformation to an observed sequence. Similar to the experimental results of the synthetic dataset, fuzzification coefficient and the number of clusters (or observed states) are also associated with the performance of these detectors.

Table 3-6 Improvement of the proposed detectors vis-à-vis the basic detector with PCA (%)

	FCM+HMM	Sugeno_Integral+HMM	Choquet_Integral+HMM
U.S. Dollar Exchange Rate Dataset	6.6474	21.0983	20.5925
Air Quality Dataset	14.2263	9.8170	16.1398

EEG Eye State Dataset	0.0635	5.0810	5.3350

Table 3-6 summarizes the improvement of the proposed detectors vis-à-vis the basic detector with PCA when the optimal parameters have been utilized. Compared to the results obtained by applying the PCA to the multivariate time series to form a univariate time series through a linear transform, the fuzzy integral is more flexible as the relative importance of different variables is also considered. In summary, the improvement of the ability of detecting anomalies can be attributed to them containing more useful information in the transformation, which might provide more help for HMM-based detectors.

3.5 Summary

In this chapter, we have investigated the multivariate time series anomaly detection problem by involving different transformation methods and HMM. The objective of this study was to compare different transformation approaches in HMM-based anomaly detection methods. Fuzzy integral and FCM clustering methods can retain more useful information in the transformation process and offer more help for HMM-based detectors to deliver better performance. A series of experiments involving synthetic and real dataset is completed to demonstrate the performance of the proposed detectors. Although the proposed anomaly detectors show good performance, there is a major limitation of intensive computing, especially in case of fuzzy integral based detectors. To overcome this problem, a certain alternative would be to engage experts in specifying some initial values of degrees of importance. The method comes with some limitations as we have only concentrated on amplitude anomalies in multivariate time series. Therefore, detecting other types of anomalies (e.g., shape anomalies) for larger datasets is a useful further direction. Another pursuit worth investigating is to quantify information loss when transforming from multivariate time series to univariate time series.

Chapter 4

Cluster-Centric Anomaly Detection in Multivariate Time Series Data[b]

The last chapter covers a supervised multivariate time series anomaly detection. Unfortunately, in the real world, there are not enough available training samples in most cases. Also, the task of labeling samples is very time-consuming. Therefore, in this chapter, we propose a new unsupervised cluster-centric approach to detect anomalies in the amplitude as well as the shape of multivariate time series. First, the sliding window technique is considered to generate a set of multivariate subsequences. Next, an augmented fuzzy clustering is developed to reveal a structure present within the generated multivariate subsequences. Finally, a reconstruction criterion is employed to reconstruct the multivariate subsequences with the help of the optimal cluster centers and the partition matrix. A certain index is constructed to quantify a level of anomaly detected in the series. Particle Swarm Optimization (PSO) is employed as an optimization vehicle to carry out anomaly detection.

4.1 Cluster-Centric Anomaly Detection

In this section, we introduce a cluster-centric approach for anomaly detection in multivariate time series data and highlight its main features. Figure 4.1 and Figure 4.2 show overall processing phases carried out by running the method for detecting anomalies in the spaces of amplitude and shape, respectively. Firstly, a fixed-length sliding window (in both Figure 4.1 and Figure 4.2) is employed to divide the long multivariate time series into a set of multivariate shorter subsequences. In the next step of this proposed framework, an augmented version of the FCM clustering is used to reveal the normal structure within the data. For this purpose, a reconstruction criterion along with a particle swarm optimization is considered. Finally, using the revealed clusters and the reconstruction criterion, an anomaly score associated with each subsequence is assigned. Comparing with the proposed method for detecting amplitude anomalies, there is an extra component for

shape anomaly detection (see Figure 4.2). In fact, before carrying out anomaly detection with respect to shape information, an autocorrelation representation of time series has been considered to remove time shifts within data. In what follows we describe each component of the proposed method more formally.

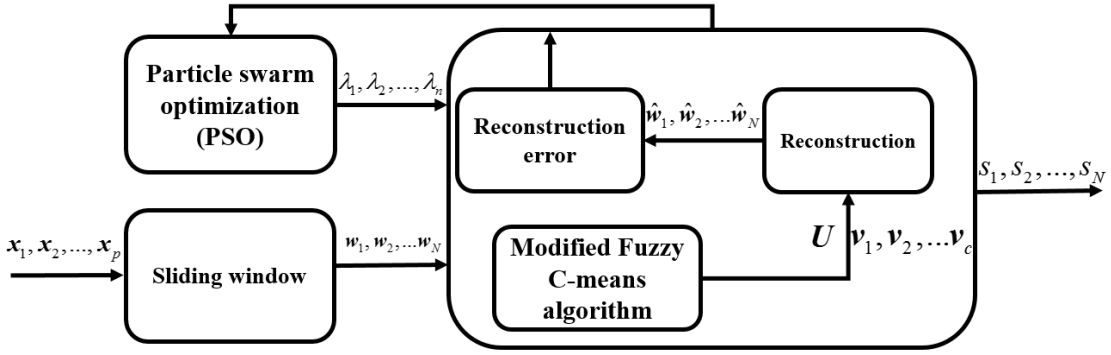


Figure 4.1 Overall scheme of anomaly detection in amplitude

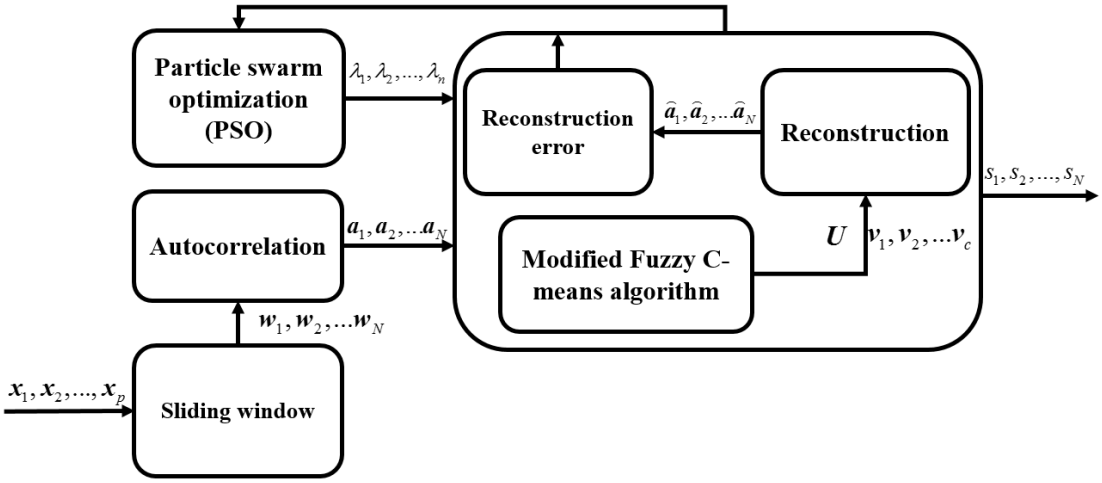


Figure 4.2 Overall scheme of anomaly detection in shape

4.1.1 Sliding window

Let us assume that $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_p$ is multivariate time series of length p. The k^{th} point in this time series is expressed using n variables, viz. $\boldsymbol{x}_k = [x_{k1}, x_{k2}, \ldots, x_{kn}]$, where n is the number of variables present in the multivariate time series. Using a fixed-length sliding window, one may generate a set of N subsequences of length q. Figure 4.3 shows an example of generating multivariate subsequences using the sliding window technique. The

sliding window moves across the time series at each movement, and the data inside the sliding window is considered as a subsequence. The first and the second subsequences are visualized. Assuming r to be the length of each movement of the sliding window, the number of generated subsequences can be determined as follows

$$N = \frac{p - q}{r} + 1 \tag{4.1}$$

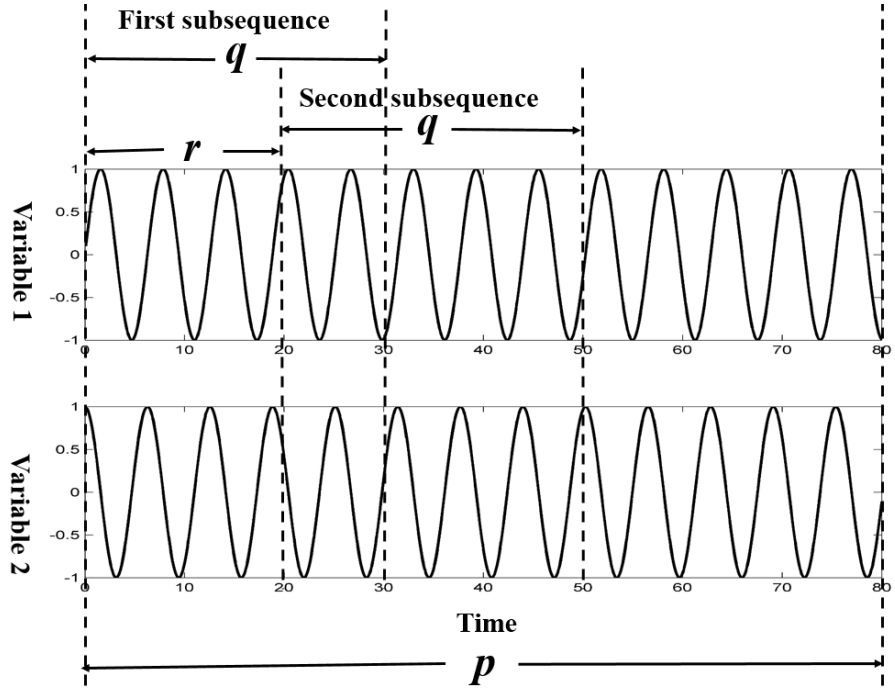


Figure 4.3 The use of the sliding window to generate multivariate subsequence.

4.1.2 An augmented Fuzzy C-Means for clustering multivariate time series

The sliding window component generates a set of multivariate subsequences. The FCM is responsible for clustering the generated multivariate subsequence to reveal the available structure within the data. Since each multivariate subsequence consists of two or more univariate subsequences, and these univariate subsequences may have different characteristics and structures, clustering this type of data using a standard FCM technique may lead to a bias towards one or more variables in the data.

To deal with this issue, we introduce a novel augmented version of fuzzy clustering for multivariate time series (subsequences). Here the augmented version of the Euclidean distance function has been considered to control the impact of each variable in evaluating

the similarity between multivariate time series. In the proposed augmented distance function, the squared Euclidean distance between a multivariate subsequence $\boldsymbol{w}_j$ and a cluster center $\boldsymbol{v}_j$ can be calculated as follows:

$$d^2(\boldsymbol{w}_j, \boldsymbol{v}_i) = \lambda_1 \|w_{j1} - v_{i1}\|^2 + \cdots + +\lambda_n \|w_{jn} - v_{in}\|^2,\ \lambda_i \geq 0, \sum_{i=1}^{n} \lambda_i = 1 \tag{4.2}$$

Using distance function in (2), one may control the impact of each variable in the clustering process of multivariate time series. A higher value of λ_i, leads to the increase of the impact of the i^{th} variable and the decrease of the impact of the other variables in the clustering process. By inserting the augmented distance function in FCM objective function, we obtain at the following expression.

$$Q = \sum_{i=1}^{c} \sum_{j=1}^{N} u_{ij}^m d^2(\boldsymbol{w}_j, \boldsymbol{v}_i) \tag{4.3}$$

where c is the number of clusters, $m(m > 1)$ is the fuzzification coefficient, and N is the number of multivariate subsequences. U and $\boldsymbol{v}_i$ are membership matrix and prototype i. The optimization process of the proposed augmented objective function is realized in an iterative fashion. Here we compute the partition matrix and cluster centers using the following expressions (4.4) and (4.5).

$$\boldsymbol{v}_i = \frac{\sum_{j=1}^{N} u_{ij}^m \boldsymbol{w}_j}{\sum_{j=1}^{N} u_{ij}^m} \tag{4.4}$$

$$u_{ij} = \frac{1}{\sum_{l=1}^{c} \left(\frac{\|\boldsymbol{v}_i - \boldsymbol{w}_j\|}{\|\boldsymbol{v}_l - \boldsymbol{w}_j\|} \right)^{\frac{2}{(m-1)}}} \tag{4.5}$$

Using the proposed augmented Fuzzy C-Means, one may control the impact of each variable in clustering multivariate time series data. However, an optimal impact of each variable in the clustering process is required. In other words, an optimal value of each coefficient $\lambda_i, i = 1,2, \ldots, n$ has to be estimated.

Since the augmented distance function expressed in (2), there are n weights $\lambda_i, i = 1,2, \ldots, n$, considering all possible combinations of these values (to find an optimal impact of each variable in the clustering process) is time consuming and for higher values of n is

not even feasible. As a result, a particle swarm optimization (PSO) has been considered to find (near) optimal values of the weights because of its superiority in solving the complex global optimization problem.

In brief, PSO produces some potential global solutions for the search problem and improves the quality of particles (solutions) using some search strategies [180, 181]. In this chapter, each particle comprises a *n*-dimensional vector that corresponds to the coefficients of the augmented FCM and satisfies the constraint imposed by (2). It starts with producing randomly a number of particles and their velocity vectors. For each particle, we evaluate its quality using a reconstruction criterion. In fact, the reconstruction criterion as discussed in the next section serves as a fitness function of the PSO technique. The particles and their velocity vectors are updated using the following expressions.

$$v_{ki}^{t+1} = w \times v_{ki}^{t} + c_1 r_{1i}(pbest_{ki}^{t} - z_{ki}^{t}) + c_2 r_{2i}(gbest_{i}^{t} - z_{ki}^{t}) \tag{4.6}$$

$$z_{ki}^{t+1} = z_{ki}^{t} + v_{ki}^{t+1}, \qquad k = 1,2,\dots,M; i = 1,2,\dots,n \tag{4.7}$$

where, $v_{ki} \in [v_{min}, v_{max}]$, M is the number of particles and n is the number of dimensions of each particle. $pbest$ and $gbest$ are best location achieved by particle and the best location found by the whole swarm, respectively. The inertia weight, w, controls the impact of the previous velocity on the current one. c_1 and c_2 are acceleration coefficients and r_{1i} and r_{2i} are random values drawn from a uniform distribution over the $[0,1]$ interval.

4.1.3 Reconstruction criterion

As discussed in the previous subsection, a reconstruction criterion serves as a fitness function considered in the PSO method. In fact, using this technique one may evaluate the quality of clusters in terms of data granulation and degranulation. The essence of the reconstruction criterion is to reconstruct the original data (subsequences) through the revealed cluster centers and their membership values [60]. Considering the cluster centers and the partition matrix generated using the augmented FCM, one may reconstruct the original subsequences by minimizing the following sum of distances:

$$F = \sum_{i=1}^{c} \sum_{j=1}^{N} u_{ik}^{m} \, d^2\big(\widehat{\boldsymbol{w}}_j, \boldsymbol{v}_i\big) \tag{4.8}$$

where, $\widehat{\boldsymbol{w}}_j$ is the reconstructed version of $\boldsymbol{w}_j$. By zeroing the gradient of F with respect to $\widehat{\boldsymbol{w}}_j$, one has

$$\widehat{\boldsymbol{w}}_i = \frac{\sum_{j=1}^{c} u_{ij}^m \boldsymbol{v}_j}{\sum_{j=1}^{c} u_{ij}^m} \tag{4.9}$$

After reconstructing all data points using (9), we calculate the reconstruction error as the following sum of distances:

$$E = \sum_{j=1}^{N} \left\| \boldsymbol{w}_j - \widehat{\boldsymbol{w}}_j \right\|^2 \tag{4.10}$$

Lower values of E indicate the higher quality of clusters in terms of data granulation and degranulation. The PSO technique described in the previous subsection, optimizes the coefficients $\lambda_i, i = 1,2, \dots, n$ by minimizing the reconstruction error (10).

4.1.4 Reconstruction error as anomaly score

Optimizing the weights in the augmented FCM technique will lead to the determination of an optimal impact of different variables in the clustering process. Therefore, the revealed cluster centers and partition matrix corresponding to the optimal weights are considered as optimal ones. In the next step, using the optimal cluster centers and partition matrix, one assigns an anomaly score to each multivariate subsequence. For this purpose, again the reconstruction criterion is considered. In fact, the anomaly score for each multivariate subsequence is calculated as the squared Euclidean distance between the subsequence and its reconstructed version. Formally, an anomaly score for subsequence $\boldsymbol{w}_j$ is calculated in the following form.

$$s_j = \left\| \boldsymbol{w}_j - \widehat{\boldsymbol{w}}_j \right\|^2 \tag{4.11}$$

4.1.5 Correlation coefficients representation of time series

When detecting amplitude anomalies is in concern, the original representation of time series can be considered, and the Euclidean distance function is a suitable measure for evaluating the similarity/dissimilarity of time series. On the other hand, for shape anomalies, since the subsequences might be impacted by time shifts, using the Euclidean distance function as a similarity measure in the original feature space is not a suitable choice.

Autocorrelation coefficients of time series have been considered to handle this issue. By representing subsequences in their autocorrelation coefficients feature space, one may remove time shifts and then the Euclidean distance can be used in this new feature space. As an example, let us consider the generated time series A, B, and C as shown in Figure 4.4. If we use Euclidean distance for the time series A, B, and C, it is obvious that A and C are more similar (exhibit lower values of the distance) because of the existing time shift between A and B. By considering the autocorrelation space of time series, the time shifts in time series are removed and then time series A and B are considered to be similar.

For a subsequence $\boldsymbol{w}_j$ of length q, its autocorrelation coefficients can be calculated in the following form

$$a_{j,e} = \frac{\sum_{t=e+1}^{q}\left(w_{j,t} - \overline{w}_j\right)\left(w_{j,t-e} - \overline{w}_j\right)}{\sum_{t=1}^{q}\left(w_{j,t} - \overline{w}_j\right)^2} \tag{4.12}$$

Where, $e = 1,2,\dots,q-1$ and $j = 1,2,\dots,N$. $\overline{w}_j$ is the mean of the time series.

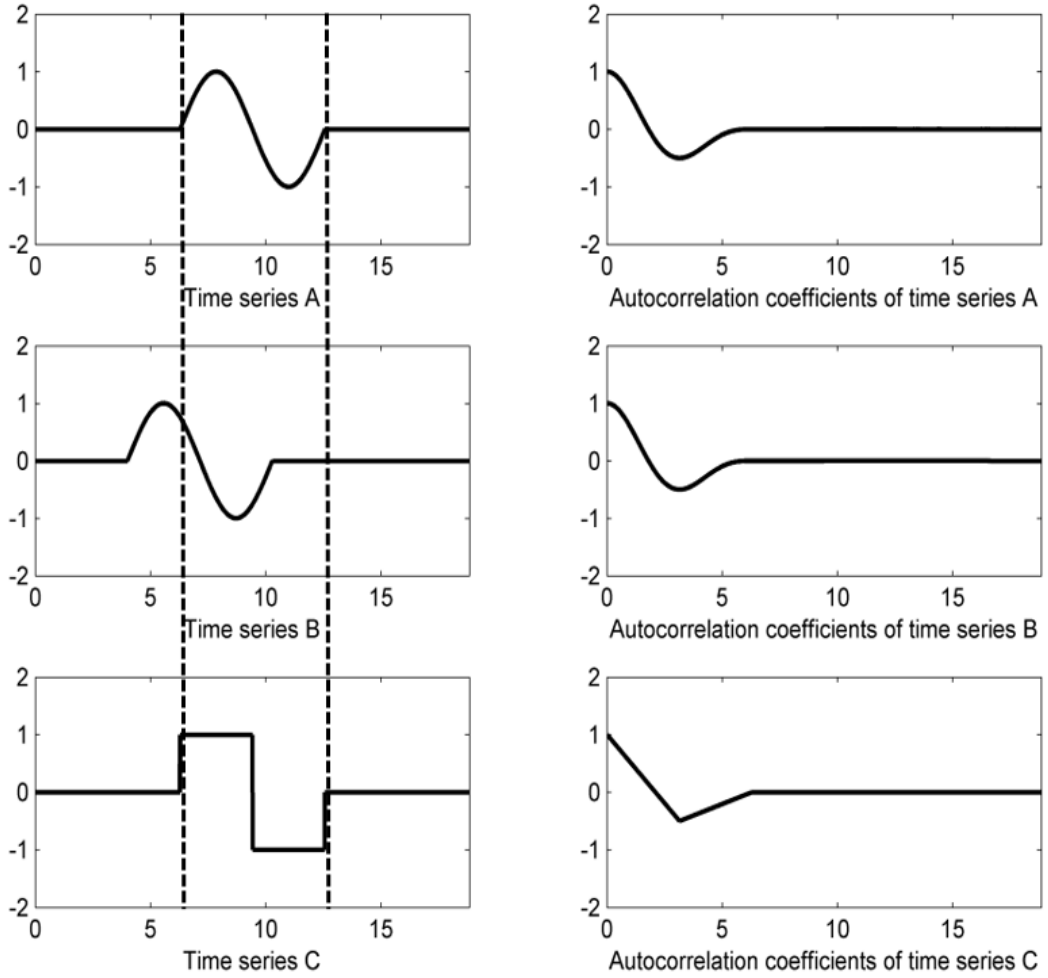


Figure 4.4 Comparison of subsequences A, B, and C along with their autocorrelation coefficients.

4.1.6 Parameter selection

Before running the proposed techniques, a Z-score normalization has been applied to each variable (time series) of the multivariate time series to remove the scaling effects among different variables. There are also some parameters such as fuzzy coefficient m, number of clusters c, the length of subsequences q, and the length of each step r whose values can impact the detection process. Comparing the score of anomalous subsequence with the average score of subsequences, a confidence index is considered to optimize those parameters. Figure 4.5 shows the idea behind this index when there is only one anomaly in time series.

The confidence index [182] is expressed using the following ratio

$$f = \frac{h_{anomaly}}{\bar{h}} \tag{4.13}$$

where h stands for the anomaly score for each subsequence, $\bar{h}$ is the average of anomaly scores and $h_{anomaly}$ anomaly score of anomalous subsequences. Encountering a higher value of f indicates assigning a higher anomaly score to anomalous part and lower scores to other parts. As the result, (13) can be used to find the optimal values of the parameters. The reason is that the higher value of f indicates that the difference between scores of normal subsequences and abnormal subsequences is bigger. When there is more than one anomaly in time series, $h_{anomaly}$ is the average of anomaly scores of anomalous subsequences.

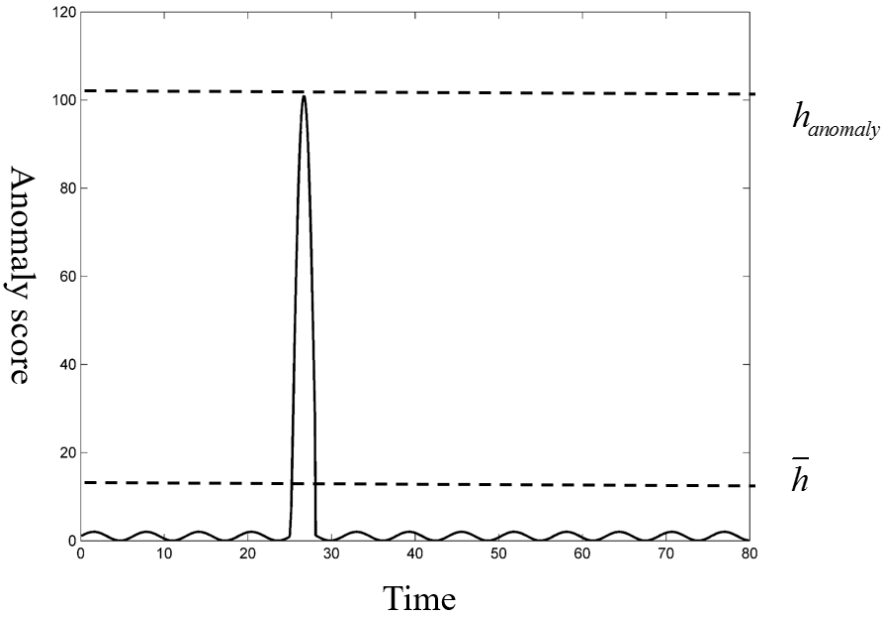


Figure 4.5 Confidence index (anomaly in the interval [26,29]).

4.2 Experimental Studies

In this section, the cluster-centric approach is evaluated over a set of synthetic and real-world multivariate time series.

4.2.1 Synthetic datasets

A synthetic multivariate time series is generated using some non-linear functions including log, sine, absolute values, cosine and exponential functions with an intent to test

the effectiveness of the proposed method. The length of the generated 5-dimensional synthetic time series is 4,000. The individual time series are as follows:

$$f_1(t) = sin(t)$$

$$f_2(t) = sin(t)^2$$

$$f_3(t) = sin(t) + sin(t)^2$$

$$f_4(t) = e^{(|sin(t)|+0.1)} + log(sin(t)^2 \times 0.0001)$$

$$f_5(t) = log(|sin(t)| + 0.1) + log(sin(t)^2 + 0.1)$$

In the series of experiments, the number of clusters was set from 2 to 6. The length of the subsequence is set to be in the interval [40, 400]. The translation of the sliding window is equal to 10% of the length of subsequences. The optimal of the number of clusters and the length of subsequence are determined by the confidence index described by (13). The fuzzification coefficient was set to 2.0.

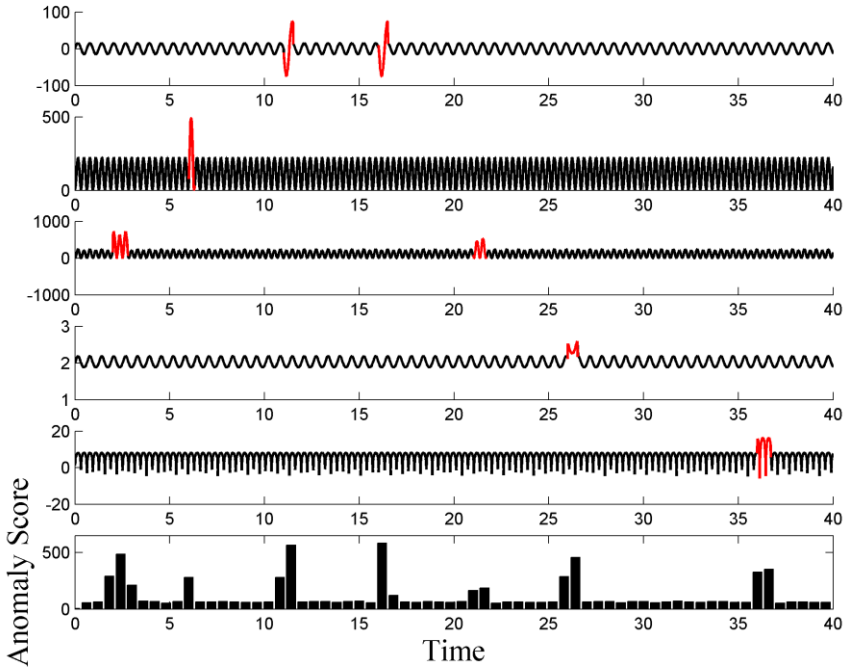
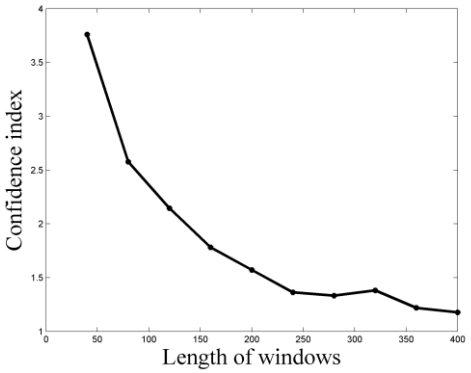


Figure 4.6 Multivariate time series with existing amplitude anomalies.

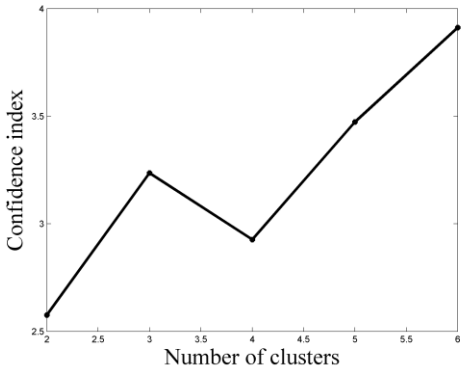
We inserted amplitude anomalies by randomly picking up some windows in different coordinates of the generated multivariate time series and then multiplying the values of picked windows with the random values of the interval [0, 5]. Figure 4.6 shows the multivariate time series with amplitude anomalies and the anomaly score of each

subsequence. It should be noted that this figure only shows the maximal anomaly score for each subsequence because of the overlap between subsequences. When the performance index (13) has been optimized, the length of windows and number of clusters are set to 40 and 6, respectively.

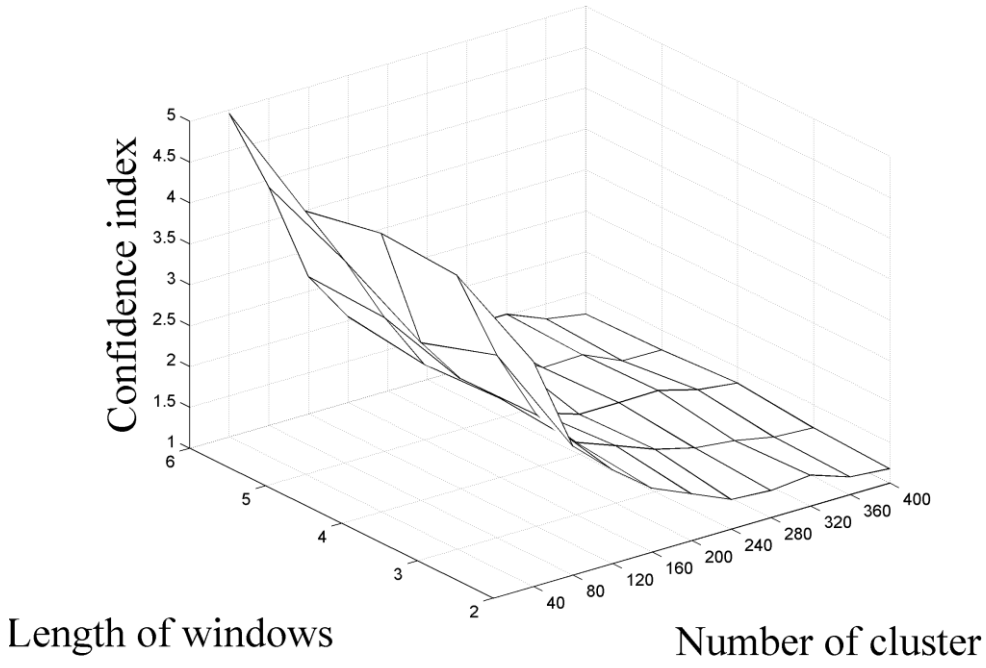
For the PSO, the number of particles is 100. The estimated optimal weights are $\lambda_1 = 0.05$, $\lambda_2 = 0.48, \lambda_3 = 0.03, \lambda_4 = 0.42$, and $\lambda_5 = 0.02$. As evidenced in Figure 4.7, the proposed method was able to detect all anomalies present in the multivariate time series.



(a)



(b)



(c)

Figure 4.7 (a) Different length of windows vs. confidence index (when number of clusters is 2); (b) Different number of clusters vs. confidence index (when length of windows is 80); (c) confidence index when length of sliding window and number of clusters take different values (amplitude anomaly).

Let us now evaluate the proposed method with regard to its abilities to detect and quantify shape anomalies. For this purpose, some shape anomalies have been injected into the generated multivariate time series used in the previous experiment. These shape anomalies are produced by picking up some windows randomly and changing the frequency (by multiplying the random value in the interval [1, 3]) of the signal within the windows because frequency change will lead shape change of the picked windows. Figure 4.8 illustrates this multivariate time series with shape anomalies and the obtained corresponding anomaly score for each subsequence determined by the proposed method. Again, the method was able to detect shape anomalies, see Figure 4.8.

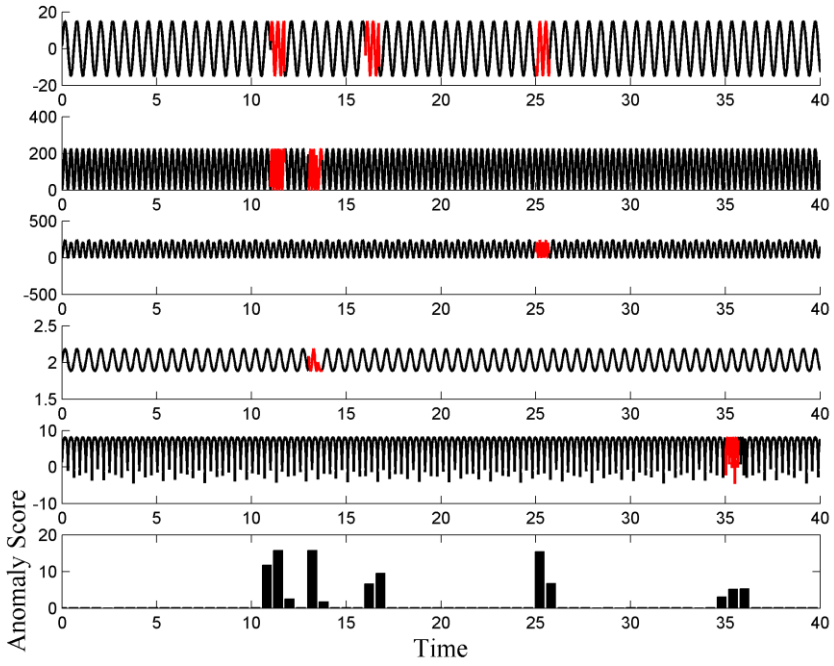
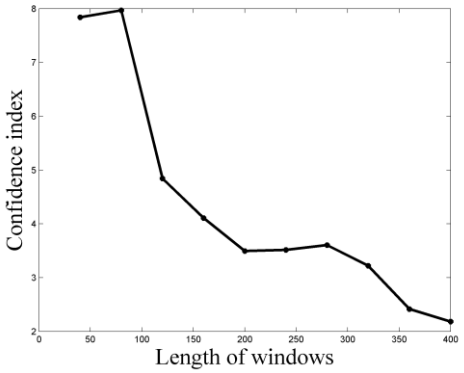


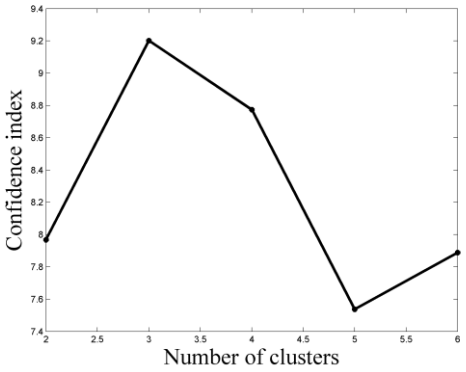
Figure 4.8 Multivariate time series with existing shape anomalies.

Figure 4.9 illustrates the values of confidence index produced for the different number of clusters and the length of subsequences. The best values of these two parameters are 40

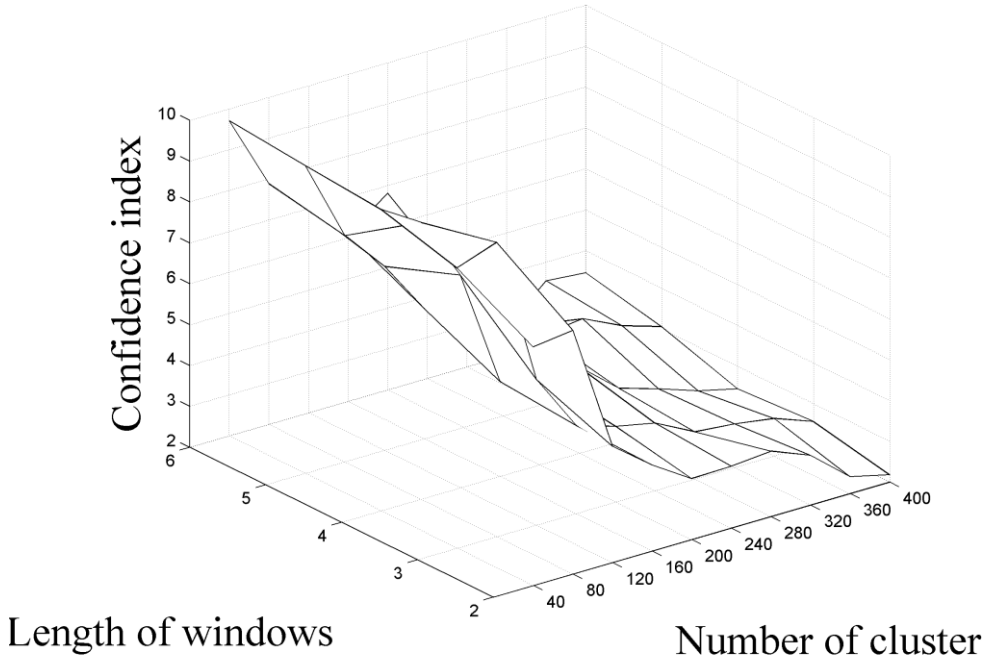
and 6 respectively. the estimated optimal values of the weights are $\lambda_1 = 0.25$, $\lambda_2 = 0.31, \lambda_3 = 0.18, \lambda_4 = 0.21$, and $\lambda_5 = 0.05$.



(a)



(a)



(c)

Figure 4.9 (a) Different length of windows vs. confidence index (when the number of clusters is 2); (b) Different number of clusters vs. confidence index (when the length of windows is 80); (c) Confidence index when the length of sliding window and number of clusters take different values

In the next experiment, a multivariate time series of the length 80 is simulated and is shown in Figure 4.10. Let us consider we have two time series (two variables) that contain

6 types of different signals (shown in different colors). Each signal is repeated for some times but the number of repeated is unknown while the relationship (1-4, 2-5, 3-6) between different signals are known. Each number represents a specific type of signal. So multivariate time series without anomaly is expressed as follows:

Time series 1: 1 1 1 1 1 2 2 2 2 2 **2** 3 3 3 3 3

Time series 2: 4 4 4 4 4 5 5 5 5 5 **5** 6 6 6 6 6

For the case shown in Figure 4.10, multivariate time series with anomaly is shown as:

Time series 1: 1 1 1 1 1 2 2 2 2 2 **2** 3 3 3 3 3

Time series 2: 4 4 4 4 4 5 5 5 5 5 **6** 6 6 6 6 6

Each signal type has been repeated at different positions in each time series. As a result, there is no anomaly in each single time series. However, when considering both time series, there are two anomalies (highlighted) because signal 2 and signal 6 are coming together. Figure 4.10 shows the generated time series as well as the calculated anomaly scores.

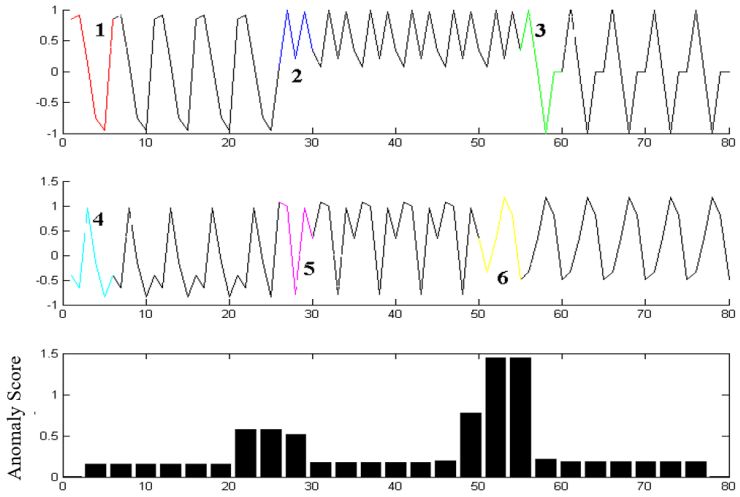


Figure 4.10 Experimental results of multivariate time series.

As shown in Figure 4.10, the proposed method is able to detect the anomalies caused by relations between time series in multivariate time series. The reason is that all univariate subsequences of multivariate subsequence are considered at the same time to determine an anomaly score of this multivariate subsequence.

In the next experiment, we provide a comparative illustration of the performance of the augmented FCM and the standard FCM in the proposed general framework of anomaly detection. In order to make all details easily visible, we consider a two-dimensional

multivariate time series which consists of four amplitude anomalies (marked by red) as shown in Figure 4.11 (top). The standard FCM and its augmented version were run on this multivariate time series with the same initialization and the following same parameters: the number of clusters = 2 and fuzzification coefficient =2.0. Their convergences are archived after a number of iterations. Figure 4.11 (middle) and (bottom) illustrates the obtained corresponding anomaly scores produced by the anomaly detectors with the augmented FCM (with the optimized weights $\lambda_1 = 0.2169$ and $\lambda_1 = 0.7831$) and the standard FCM. The detector with the augmented FCM can detect all anomalies while that with the standard FCM cannot. For an in-depth analysis, we map this multivariate time series onto 2-dimensional space and provide their clustering centers; see Figure 4.12 (a) and (b). They reveal two different data structures within data because an augmented version of the Euclidean distance function has been considered to control the impact of each variable in evaluating the similarity between multivariate time series. In addition, there is one error detection (shown in Figure 4.13 (a) and marked by blue diamond) produced by the detector with the standard FCM. On a basis of different clustering centers, we also provide the corresponding reconstruction version in Figure 4.13 (b). The distance (or the anomaly score) between actual pattern and its reconstruction version obtained by the augmented FCM is less than that obtained by the standard FCM.

As shown in Figure. 4.11-4.13, the augmented FCM is expected to reveal the available data structure for the subsequent anomaly detection because of the different impact of each variable in evaluating the similarity between multivariate time series. The detector with the augmented FCM can offer more accurate detection results.

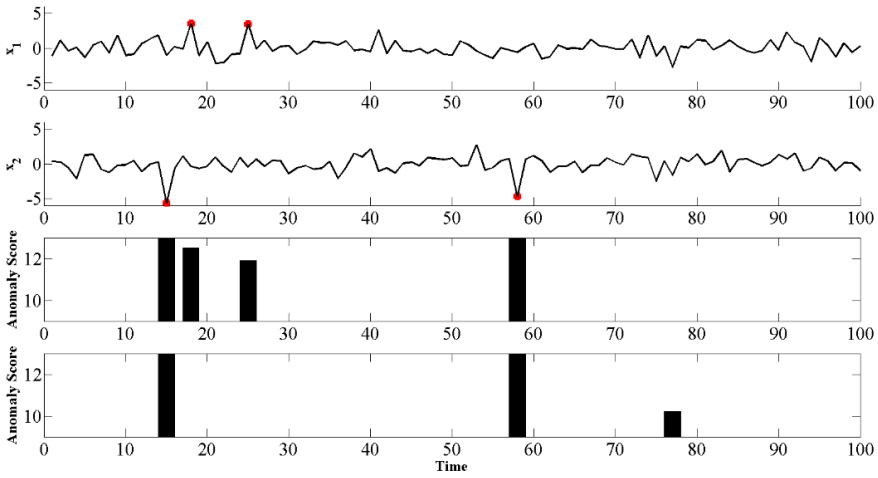
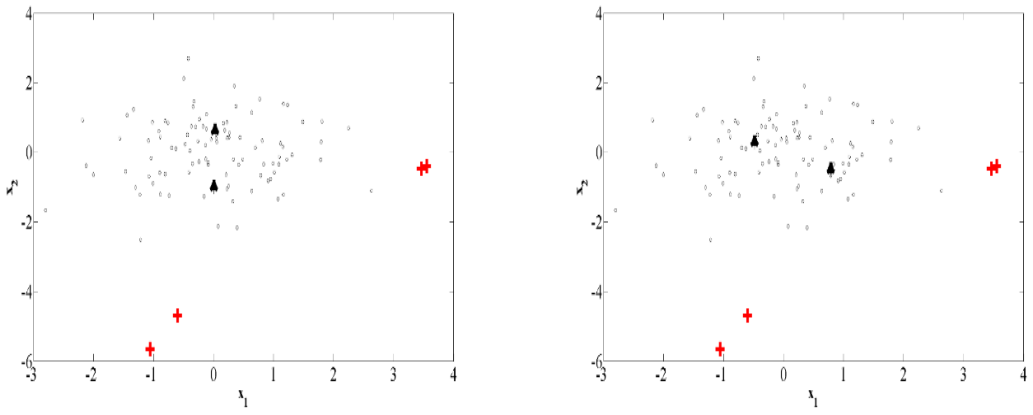
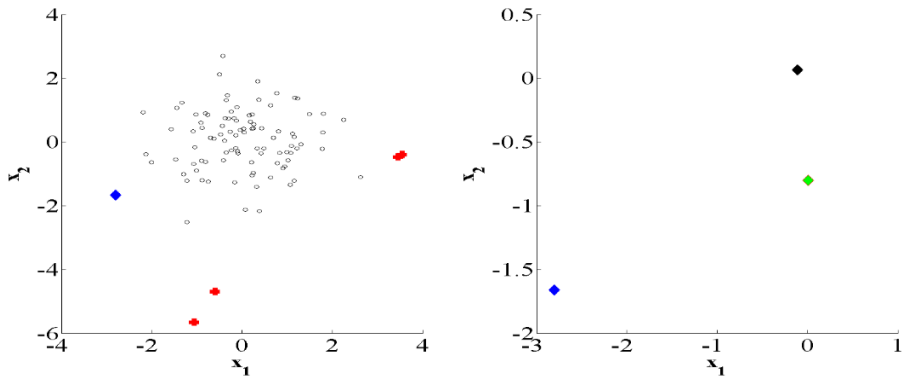


Figure 4.11 Top: a two-dimensional multivariate time which consists of four amplitude anomalies; Middle: experimental results of the augmented FCM; Bottom: experimental results of the standard FCM.



(a) (b)

Figure 4.12 (a) Clustering centers (marked by black triangles) obtained by the augmented FCM and four amplitude anomalies (marked by red pluses); (b) Clustering centers (marked by black triangles) obtained by the standard FCM and four amplitude anomalies (marked by red pluses);



(a) (b)

Figure 4.13 (a) anomalies (marked by red pluses) and error detection (marked by blue diamond); (b) error detection (marked by blue diamond) and its reconstruction versions based on the augmented FCM and the standard FCM respectively

4.2.2 Publicly available datasets

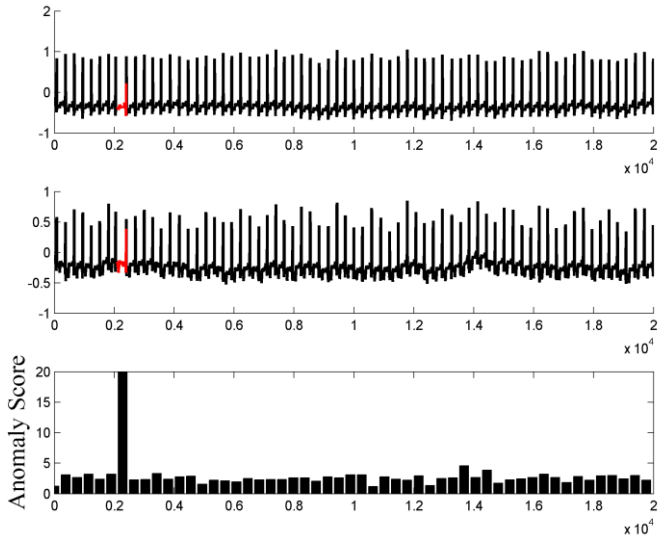
In this subsection, two real-world datasets, namely electrocardiogram data and climate change data have been studied for anomaly detection in shape and amplitude using the proposed method.

Electrocardiograms (ECG) data set

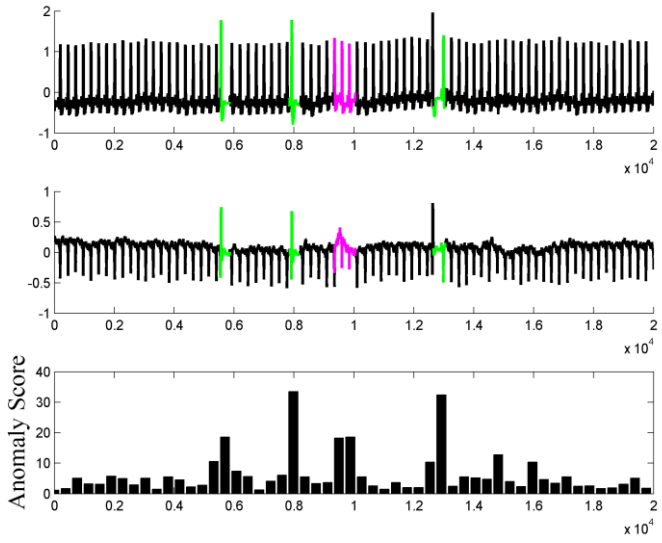
This dataset is obtained from the MIT-BIH Arrhythmia database [183, 184]. In the ECG dataset, each heartbeat is annotated by two cardiologists independently, so it can be used as a benchmark to evaluate the proposed method.

In this experiment, three ECG heartbeats with some annotated anomalies are considered. To consider all type of beats in a time window, the length of time window has been considered to be 1.2 times the average length of the RR interval [182]. The other parameters are selected in a similar fashion as used in the previous experiments. The optimal values of the number of clusters and the length of subsequence are determined by the confidence index (13). The fuzzification coefficient was set to 2.0 and the length of each slide (move) of the sliding window is set to 10% of the length of subsequences. Figure 4.14 illustrates the ECG signals and the corresponding anomaly scores generated by the proposed method. The anomalous part of each dataset is highlighted. The detected anomalous parts of data correspond to premature ventricular contraction (marked in red color) or Atrial premature contraction (marked in green). In all cases, the method can find

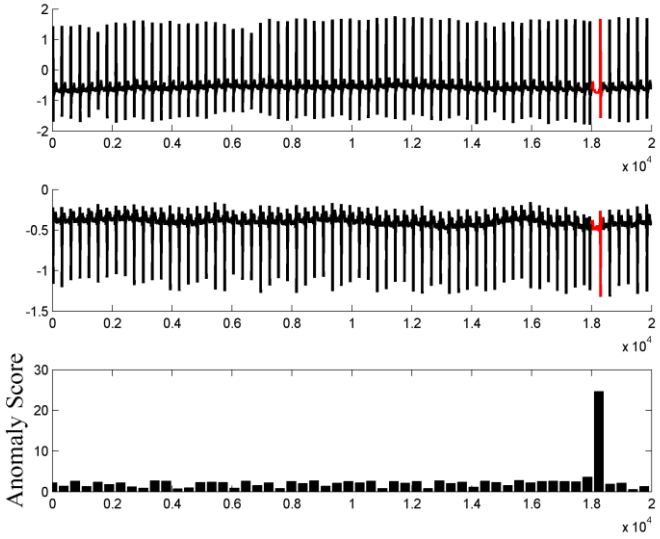
the abnormal parts of the time series. In Figure 4.14 (b), one of the detected anomalies is annotated as normal by cardiologists (highlighted in magenta), however this part of data has a visible difference in shape when compared with other normal portions of the series. As can be seen in Figure 4.14, shape information of anomalous parts (highlighted in different color) is different from that of normal parts of multivariate time series. The difference leads to different clustering results and yields a higher anomaly score of multivariate subsequence.



(a)



(b)



(c)

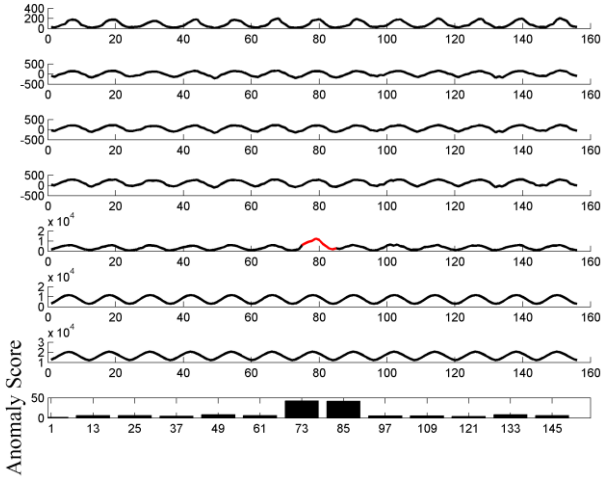
Figure 4.14 MIT-BIH arrhythmia data sets.

Climate change data set

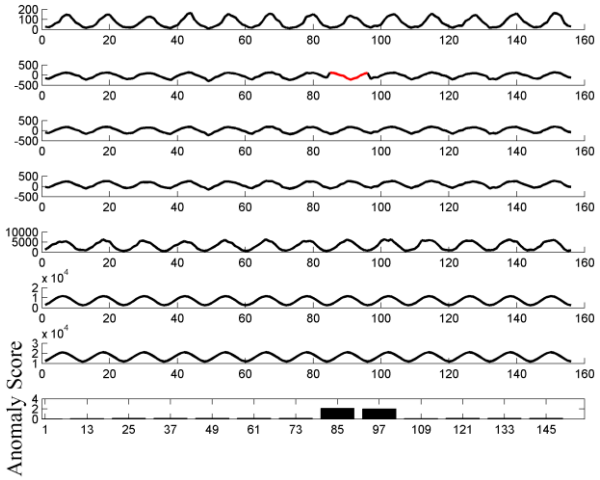
In this section, we consider the monthly measurements of climatological factors from January 1990 to December 2002, in three different places in USA. They contain vapor (VAP), temperature (TMP), temperature min (TMN), temperature max (TMX), global solar radiation (GLO), extraterrestrial radiation (ETR) and extraterrestrial normal radiation (ETRN) from CRU (http://www.cru.uea.ac.uk/cru/data), NOAA (http://www.esrl.noaa.gov/gmd/dv/ftpdata.html), and NCDC (http://rredc.nrel.gov/solar/old_data/nsrdb/).

Some simulated amplitude and shape anomalies are inserted into this multivariate time series to test our anomaly detection technique. For instance, the values of global solar radiation in the first place from April 1995 to November 1996 are doubled. The values of temperature recorded at the second place from January 1997 to December 1997 are replaced by that from July 1990 to June 1991. A sliding window of length 12 is considered to reflect successive months. The other parameter settings are similar to those used in the previous experiments. The fuzzification coefficient was set to 2.0 and the number of clusters was varied from 2 to 6. The length of each movement of the sliding window is equal to 10% of the length of the subsequences. The optimal values of weights are shown

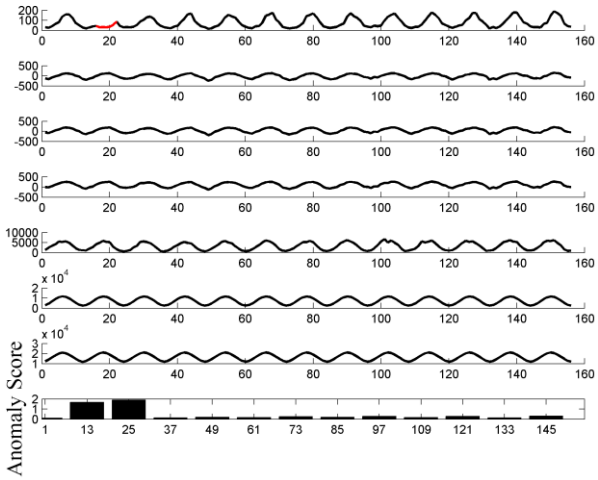
in Table 4-1. Figure 4.15 illustrates the experimental results showing the detected anomalous parts.



(a)



(b)



(c)

Figure 4.15 Climate change data sets

From the above table, we can infer that the impact of different variables could be different in the clustering process even if similar datasets are used. The higher the weights are, the more important the variables are in the clustering process.

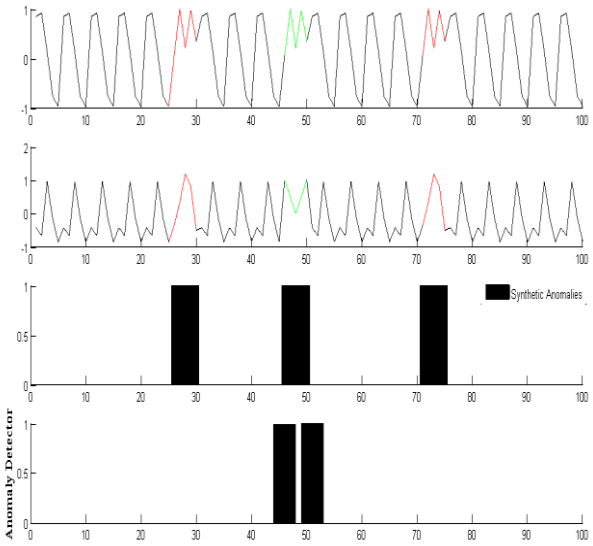
Table 4-1 Optimal values of weights

climate change data	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7
(a)	0.02	0.20	0.26	0.04	0.23	0.03	0.22
(b)	0.12	0.16	0.08	0.15	0.16	0.16	0.17
(c)	0.14	0.08	0.01	0.16	0.00	0.32	0.29

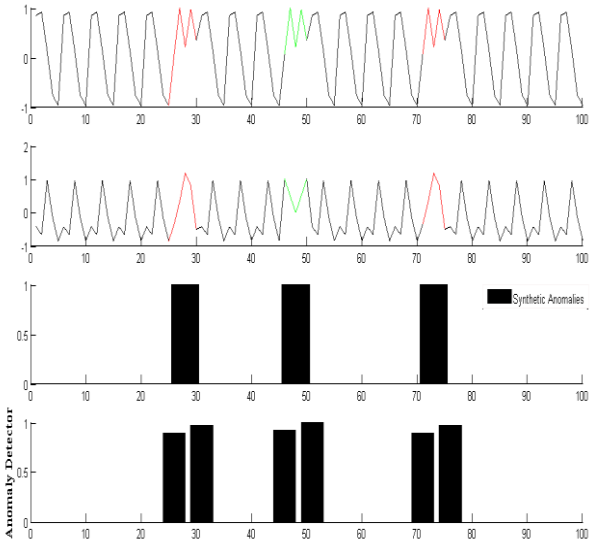
A comparison with distance-based methods

In a general framework of distance-based anomaly detection techniques one calculates the similarity of each subsequence with all other subsequences in the data set and selects the subsequences with lowest similarity (highest distance) to other subsequences as anomalous parts of data. A 1-NN technique has been used for this purpose. The main drawback of 1-NN based anomaly detection methods is that if the two similar anomalies happen in a time series, the technique is not able to detect them. To resolve this problem, one may consider the use of the *K*-NN classifier. However, finding the parameter *K* is a challenging problem. On the other hand, in clustering-based methods, by considering the number of clusters equal to 2 or 3 (or using a cluster validity index approach) one may detect anomalies even if some anomalies are similar.

In the next experiment, we compare the proposed clustering-based technique with the one discussed in [185]. A two-dimensional multivariate time series is constructed and shown in Figure 4.16. Two anomalies are inserted into this multivariate time series. It is noticeable that the distance-based method is not able to detect all anomalous parts of data.



(a)



(b)

Figure 4.16 The proposed method vs. a 1-NN technique: (a) experimental result of the 1-NN method; (b) experimental result of the proposed method.

4.3 Summary

In this study, we introduced the method for detecting amplitude and shape anomalies in multivariate time series. The modified Fuzzy C-Means clustering was used to capture the structure of multivariate time series. A reconstruction error serves as the fitness function of the PSO algorithm and also has been considered as the level of anomaly detected in each subsequence. We conducted experiments using both synthetic and real-world datasets. Experimental results demonstrate the effectiveness of the proposed methods to detect amplitude and shape anomalies in multivariate time series. Although the proposed method shows good performance, it comes with some limitations. One of them is an intensive computing overhead resulting from the fact that the Euclidean distance function calls for a substantial level of computing. Considering the computing overhead associated with the analysis of different combinations of the entries of λ, one may consider a simplified version of the method with some predetermined values of the entries of the vector of these parameters.

Chapter 5

Time Series Reconstruction and Classification: A Comprehensive Comparative Study[c]

After introducing supervised and unsupervised time series anomaly detection, let us switch to the second time series mining task, namely time series representation/approximation. In this chapter, we carry out a comprehensive analysis of relationships between reconstruction error and classification performance when dealing with various representation (approximation) mechanisms of time series. Typically, time series approximation leads to the representation of original time series in the space of lower dimensionality in comparison with the dimensionality of the original signals. We reveal, quantify, and visualize the relationships between the reconstruction error and classification error (classification rate) for a number of commonly encountered representation methods. Through carefully structured experiments completed for sixteen publicly available datasets, we demonstrate experimentally that the classification error obtained for time series in the developed representation space becomes smaller than when dealing with original time series. It has been also observed that the reconstruction error decreases when increasing the dimensionality of the representation space. Experimental results report the performance of classification and reconstruction results when dealing with clustering methods.

5.1 Reconstruction aspects associated with the FCM method

As mentioned earlier, we focus on the relationship between the classification error and the reconstruction error of various time series representation methods. Once the FCM clustering algorithm (mentioned in Chapter 4) has been completed, by invoking cluster centers and partition matrix, one reconstructs the original time series by minimizing the following function [186].

$$F = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{r} d^2 \left\| \cdot_j \quad {}_{\iota} \right\| \tag{5.1}$$

Here $\widehat{\boldsymbol{x}}_j'$ is the reconstructed version of $\boldsymbol{x}_j$. By zeroing the gradient of F with respect to $\widehat{\boldsymbol{x}}_j'$, we have

$$\sum_{i=1}^{m} u_{ij}^{r} \boldsymbol{v}_i \qquad (5.2)$$

In most cases, time series representation is performed as preprocessing of time series mining while time series clustering is presented its the following step. The main reason behind this behavior is that clustering itself can be time consuming [187]. Compared with the other time series approximation methods, there is one more parameter that is the fuzzification coefficient and impacts its reconstruction errors and classification errors.

5.2 Time series Classification

Once the approximations of time series normalized has been completed, there are a number of classification methods, like Naive Bayes Classifier, C4.5, and nearest neighbor classifier, to conduct the classification task. In this study, we use a nearest neighbor (*NN*) classifier because of its effectiveness, simplicity and parameter- free nature of the classification scheme. A testing sample without the label is assigned the class label of the most similar training sample. Normally, let's consider that X, Y and $\boldsymbol{x}$ are training set, class labels of the training set and the testing sample. Here the weighted Euclidean distance between one training sample $\boldsymbol{p}$ and one testing sample $\boldsymbol{q}$ is used.

$$Dist(\boldsymbol{p},\boldsymbol{q}) = \sum_{i=1}^{m} \frac{(p_i - q_i)^2}{\sigma_i^2} \qquad (5.3)$$

σ_i is the standard deviation of the i^{th} essential feature. Then assign the label of the sample (in the training set) which is the nearest neighbor to this testing sample $\boldsymbol{x}$.

5.3 Experimental Results

We completed a comparative study on these time series representation methods over a variety of datasets from the UCR time series databases [188], with the goal of evaluating the impact on classification accuracy of the one Nearest Neighbor (1-NN) classifier equipped with the weighted Euclidean distance. Table 5-1 summarizes the characteristics of the datasets used in the experiments. The z-normalization was applied to all datasets. In order to complete a fair comparison, before performing time series representation methods, padding with zeros to the datasets, whose length are not a power of 2, is necessary.

Table 5-1 Characteristics of publicly available datasets

Dataset	# of Class	Train	Test	Length
CBF	3	30	900	128
ProximalPhalanxOutlineAgeGroup	3	400	205	80
BeetleFly	2	20	20	512
BirdChicken	2	20	20	512
Wine	2	57	54	234
ECG200	2	100	100	96
ToeSegmentation1	2	40	228	277
ArrowHead	3	36	175	251
Beef	5	30	30	470
Trace	4	100	100	275
FaceFour	4	24	88	350
ProximalPhalanxOutlineCorrect	2	600	291	80
Gun-Point	2	50	150	150
Synthetic Control	6	300	300	60
Lighting-7	7	70	73	319
ToeSegmentation2	2	36	130	343

Except for the number of essential features that are necessary for all approximation methods, there are one important parameter for FCM, namely the fuzzification coefficient. Here we vary the values of m ranging from 2 to the length of time series with each step of 1. The fuzzification coefficient r of FCM ranges from 1.1 to 2.9 with the step of 0.1. For a clear comparison between the original time series and other time series approximations, the classification results (of original time series) obtained 1-NN classifier are also provided in Figure 5.3. Based on the results shown in Figures, the experimental results can be concisely summarized in the following way.

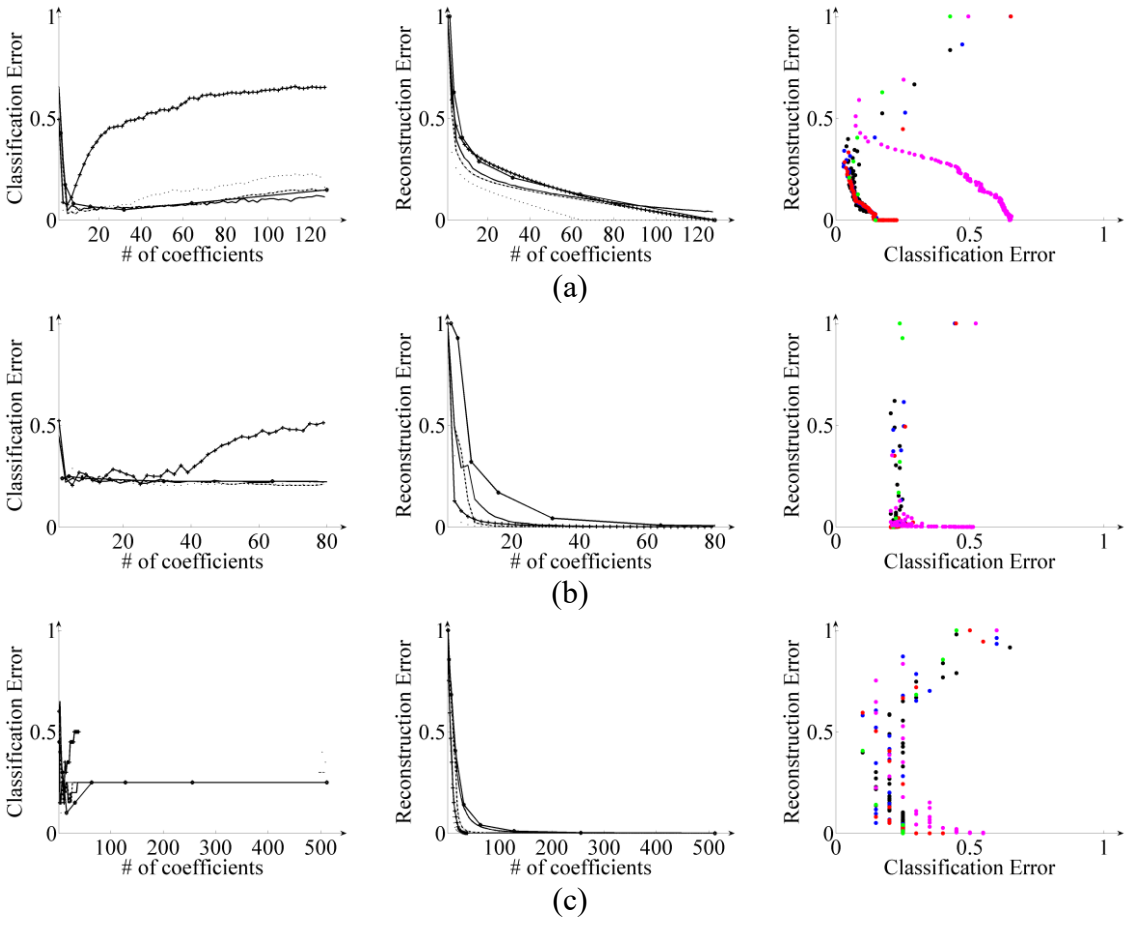
All dataset cannot be treated in the same way by performing the same approximation methods and classifications methods because the discriminative information between samples with different labels differs with the objective of the applications. For instance, the video surveillance application generates the Gun-Point dataset which identifies whether a person draws a gun or not. The time series record the position of the right hand. Only a small peak in the middle of the time series reflects whether there is the gun. If the more detailed information is preserved during the approximation process, the classification performance will be improved.

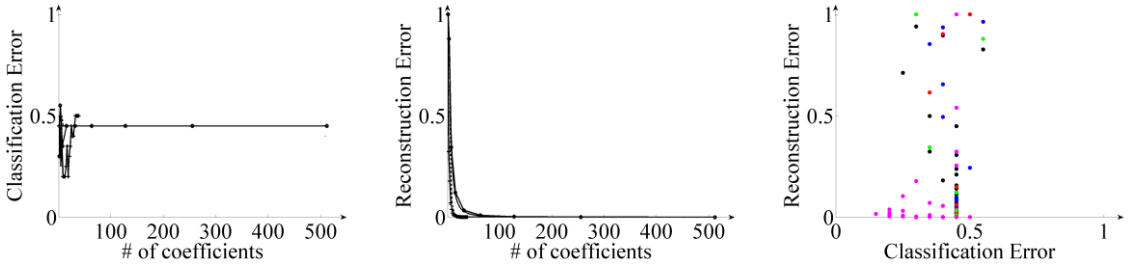
We have demonstrated experimentally that the classification error will be smaller than that of the original time series when taking suitable approximation methods and parameters

setting (shown in Figure 5.3). The approximation methods can eliminate some useless information (e.g. noise) of time series to archive better classification results. In some sense, it can illustrate why the classification error goes down and then up with the increase of the number of coefficients (shown in Figure 5.1). And we also provide the comparison (of the classification error) between original time series and their different representations after the tuning of their parameters in Figure 5.2.

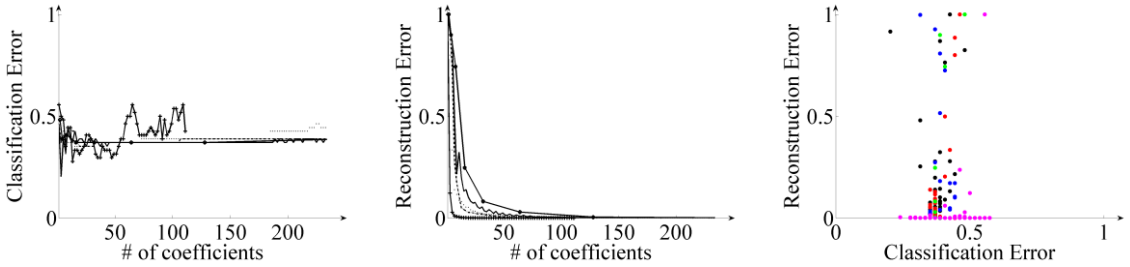
One can observe that the reconstruction error will decrease towards zero with increasing the number of coefficients in a number of cases.

Experimental results reported that the fuzzification coefficient of the FCM method also exhibits a significant impact on the performance of classification and reconstruction.

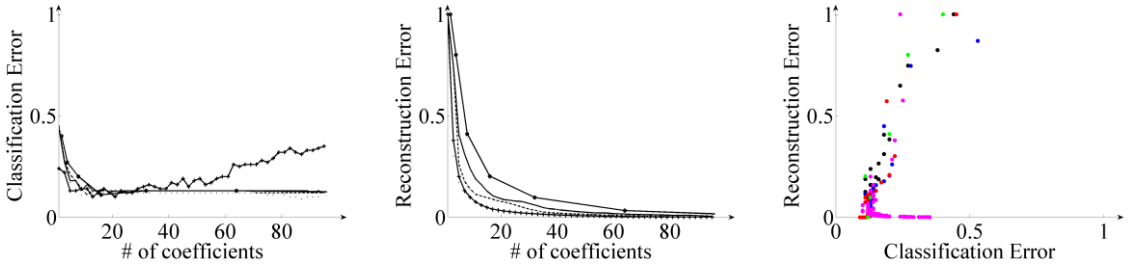



Classification Error
# of coefficients
Reconstruction Error
# of coefficients
Reconstruction Error
Classification Error

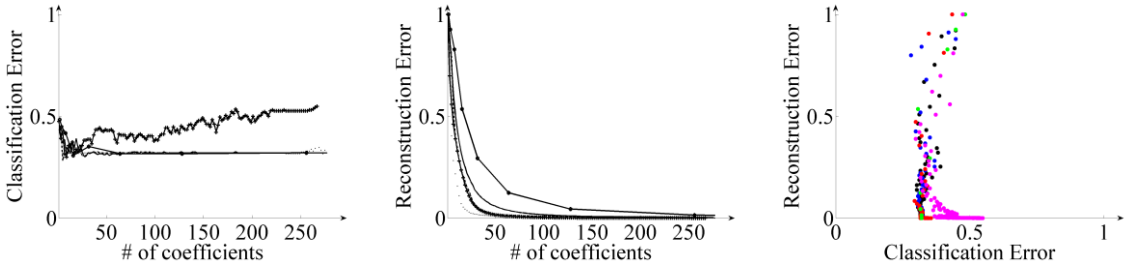
(d)


Classification Error
# of coefficients
Reconstruction Error
# of coefficients
Reconstruction Error
Classification Error

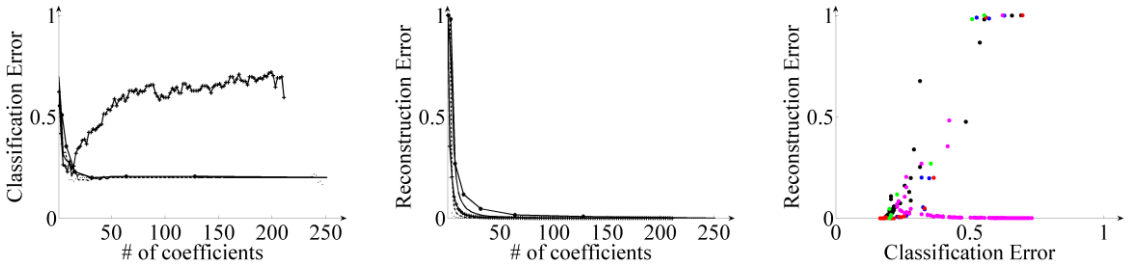
(e)


Classification Error
# of coefficients
Reconstruction Error
# of coefficients
Reconstruction Error
Classification Error

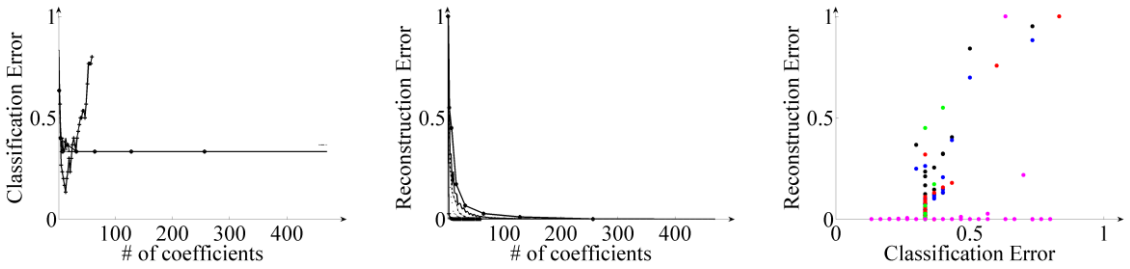
(f)


Classification Error
# of coefficients
Reconstruction Error
# of coefficients
Reconstruction Error
Classification Error

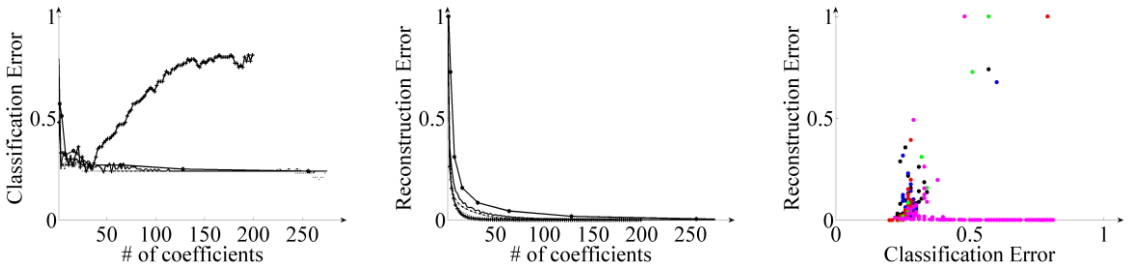
(g)


Classification Error
# of coefficients
Reconstruction Error
# of coefficients
Reconstruction Error
Classification Error

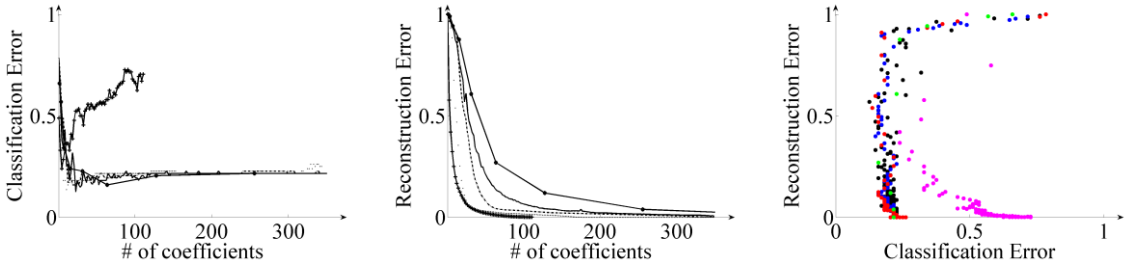
(h)


Classification Error
# of coefficients
Reconstruction Error
# of coefficients
Reconstruction Error
Classification Error

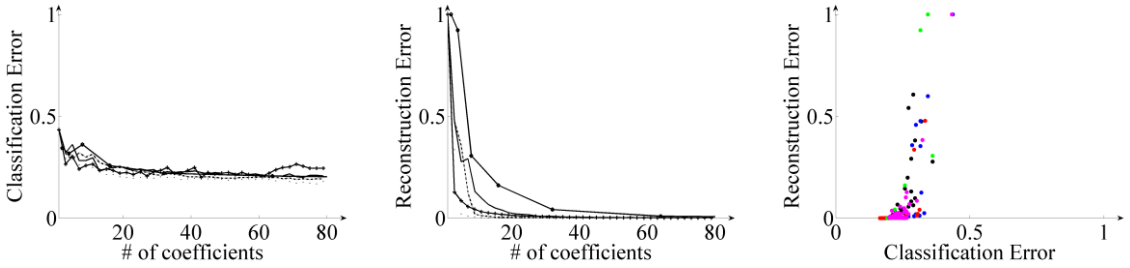
(i)


Classification Error
# of coefficients
Reconstruction Error
# of coefficients
Reconstruction Error
Classification Error

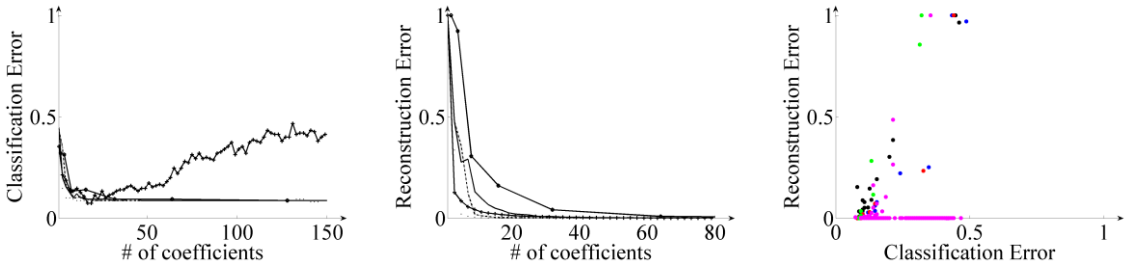
(j)


Classification Error
# of coefficients
Reconstruction Error
# of coefficients
Reconstruction Error
Classification Error

(k)


Classification Error
# of coefficients
Reconstruction Error
# of coefficients
Reconstruction Error
Classification Error

(l)


Classification Error
# of coefficients
Reconstruction Error
# of coefficients
Reconstruction Error
Classification Error

(m)

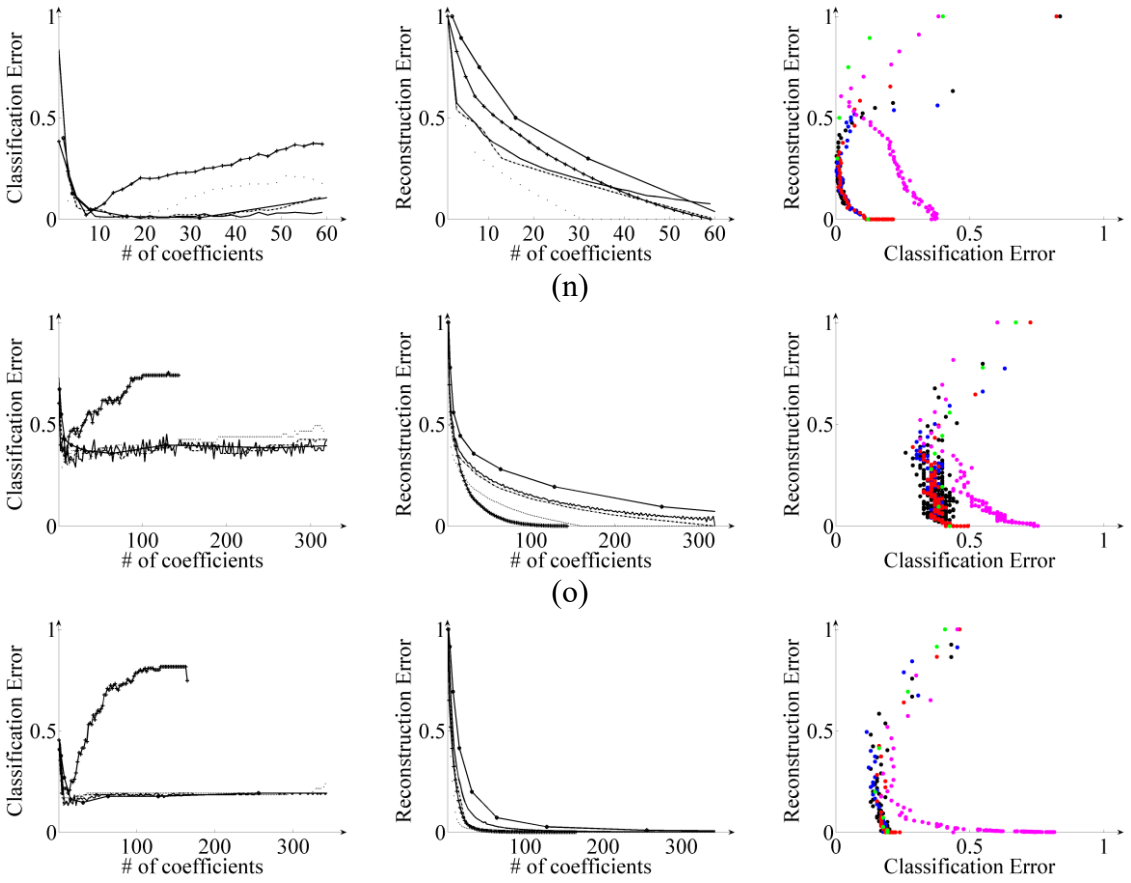
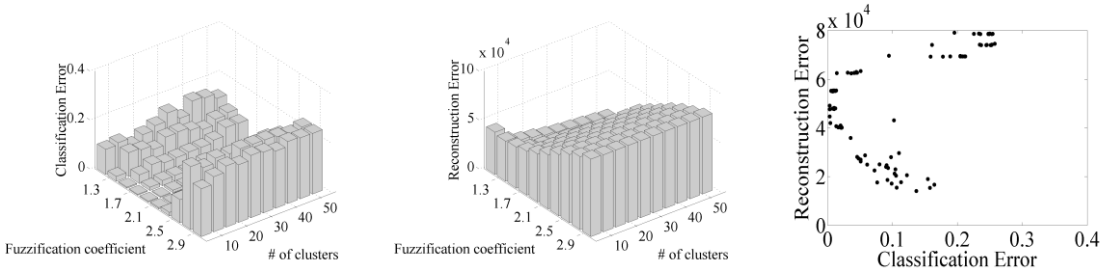
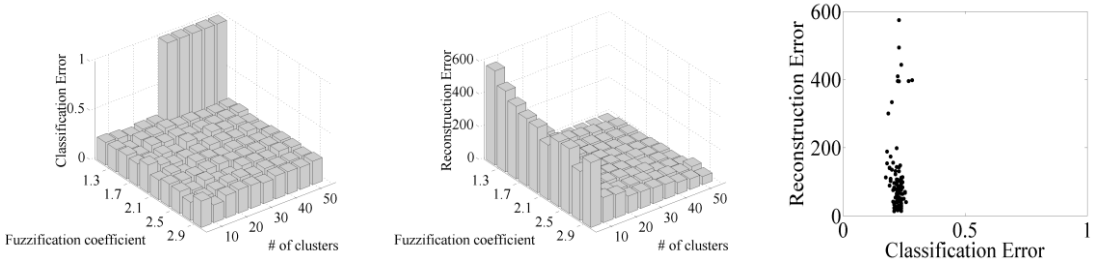


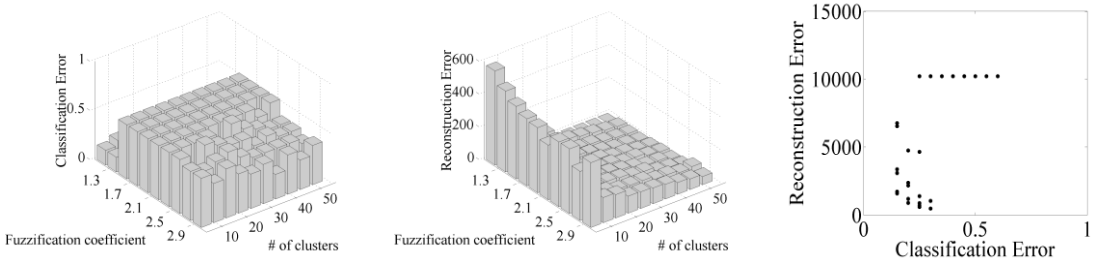
Figure 5.1 Classification error (first column) and Reconstruction error (second column) of PAA(solid), DCT(dashdot), DFT(dotted), DWT(plus) and SVD(hexagram). Comparison (third column) between classification error and reconstruction of PAA(black), DCT(blue), DFT(red), DWT(green) and SVD(magenta). (a) CBF; (b) ProximalPhalanxOutlineAgeGroup; (c) BeetleFly; (d) BirdChicken; (e) Wine; (f) ECG200; (g) ToeSegmentation1; (h) ArrowHead; (i) Beef; (j) Trace; (k) FaceFour; (l) ProximalPhalanxOutlineCorrect; (m) Gun-Point; (n) Synthetic Control; (o) Lighting-7; (p) ToeSegmentation2;


Classification Error
Fuzzification coefficient
# of clusters
Reconstruction Error
x 10^4
Classification Error

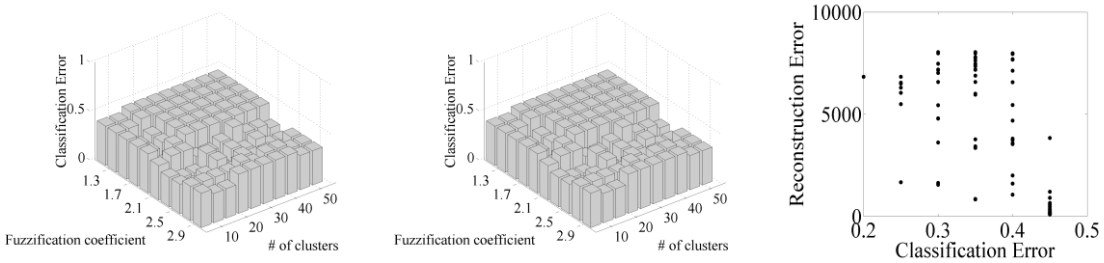
(a)


Classification Error
Fuzzification coefficient
# of clusters
Reconstruction Error
Classification Error

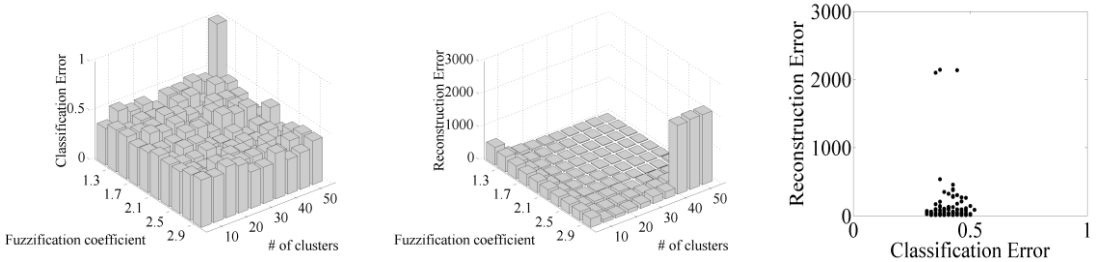
(b)


Classification Error
Fuzzification coefficient
# of clusters
Reconstruction Error
Classification Error

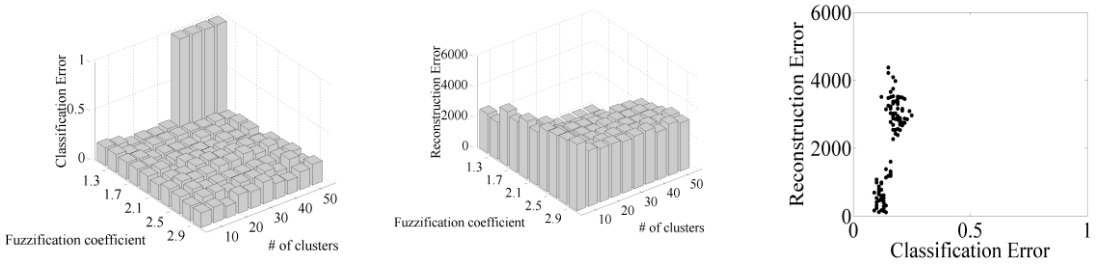
(c)


Classification Error
Fuzzification coefficient
# of clusters
Reconstruction Error
Classification Error

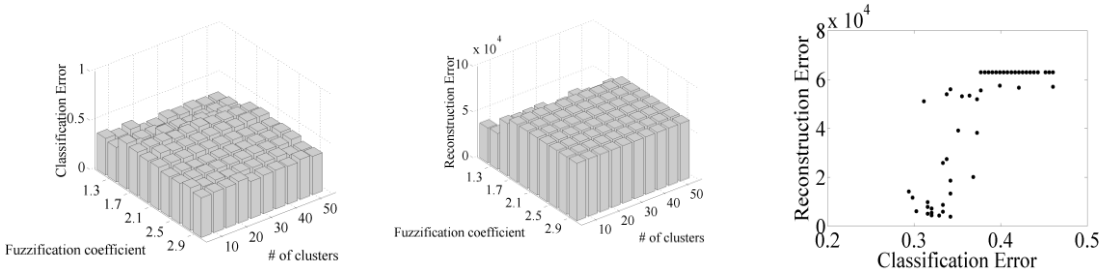
(d)


Classification Error
Fuzzification coefficient
# of clusters
Reconstruction Error
Classification Error

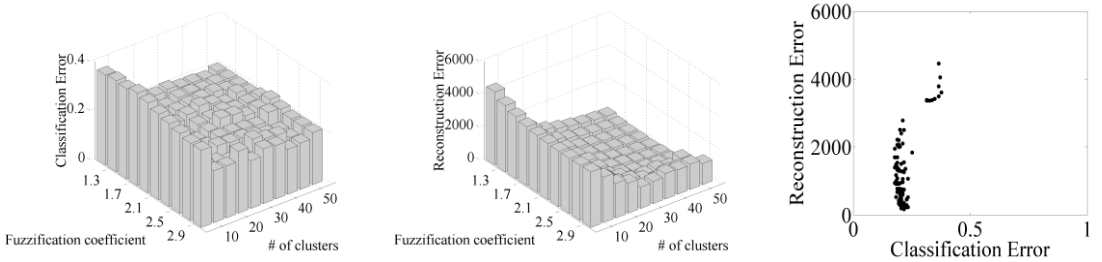
(e)


Classification Error
Fuzzification coefficient
# of clusters
Reconstruction Error
Fuzzification coefficient
# of clusters
Reconstruction Error
Classification Error

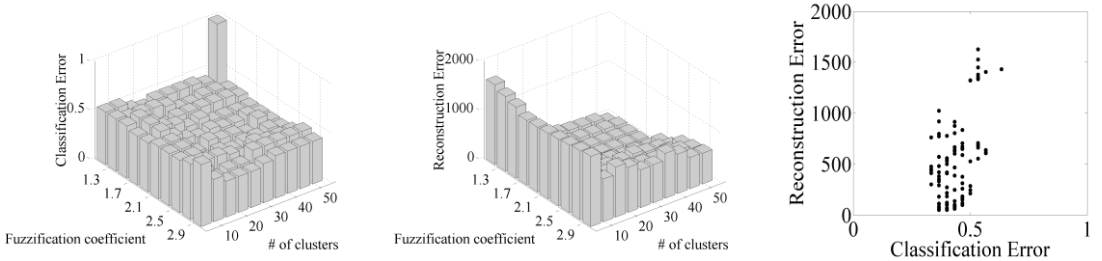
(f)


Classification Error
Fuzzification coefficient
# of clusters
x 10^4
Reconstruction Error
Fuzzification coefficient
# of clusters
x 10^4
Reconstruction Error
Classification Error

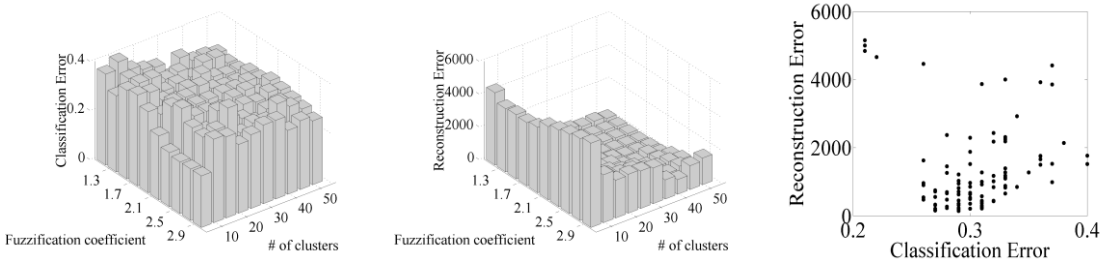
(g)


Classification Error
Fuzzification coefficient
# of clusters
Reconstruction Error
Fuzzification coefficient
# of clusters
Reconstruction Error
Classification Error

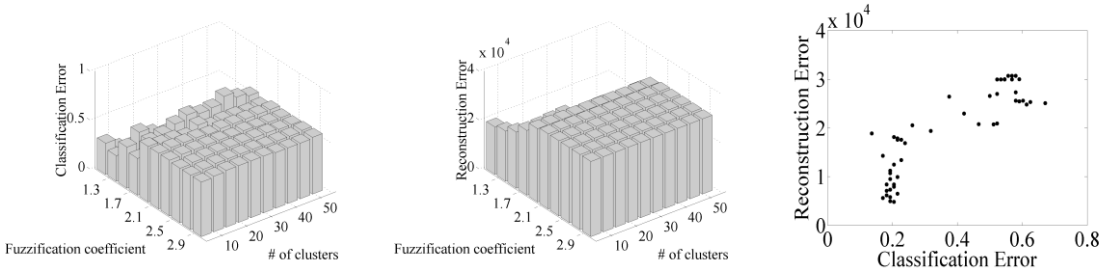
(h)


Classification Error
Fuzzification coefficient
# of clusters
Reconstruction Error
Fuzzification coefficient
# of clusters
Reconstruction Error
Classification Error

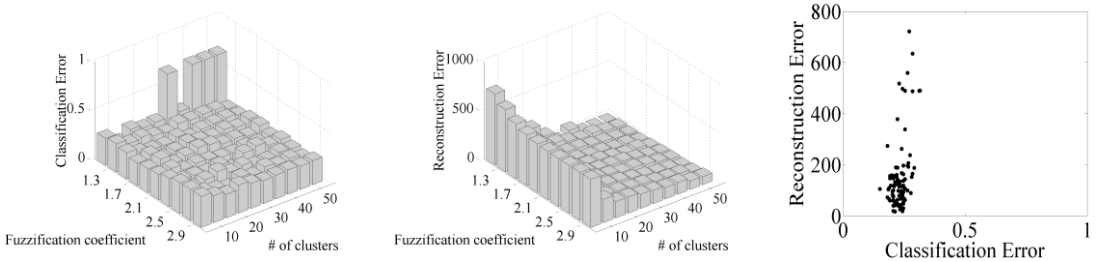
(i)


Classification Error
Fuzzification coefficient
# of clusters
Reconstruction Error
Fuzzification coefficient
# of clusters
Reconstruction Error
Classification Error

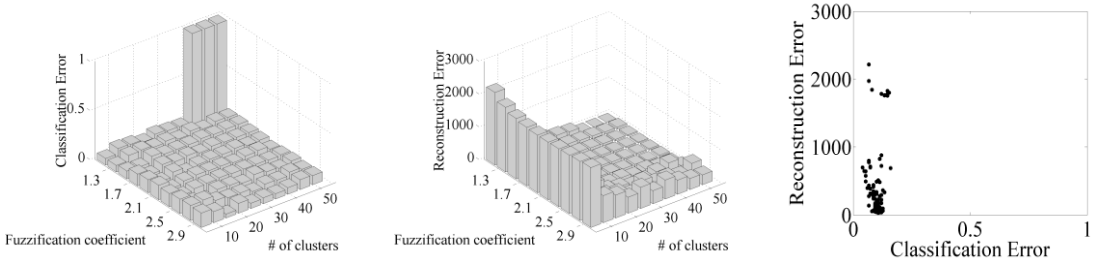
(j)


Classification Error
Fuzzification coefficient
# of clusters
Reconstruction Error
x 10^4
Reconstruction Error
Classification Error

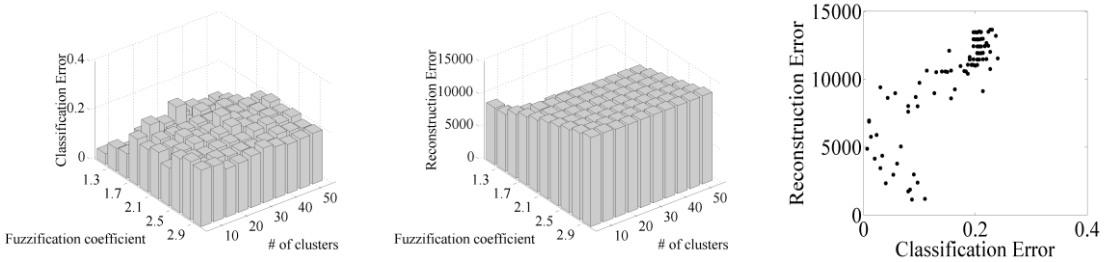
(k)


Classification Error
Fuzzification coefficient
# of clusters
Reconstruction Error
Reconstruction Error
Classification Error

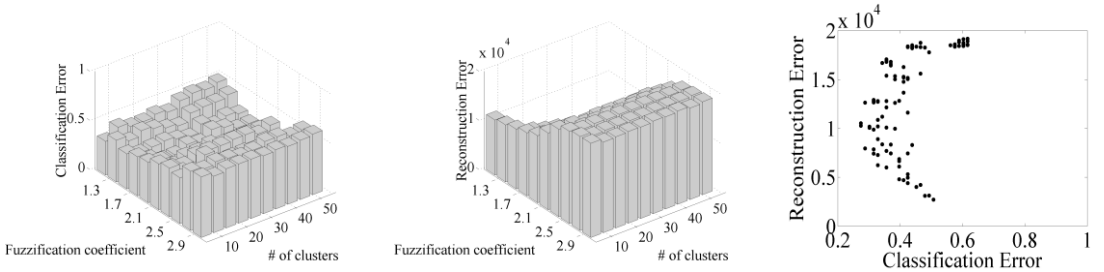
(l)


Classification Error
Fuzzification coefficient
# of clusters
Reconstruction Error
Reconstruction Error
Classification Error

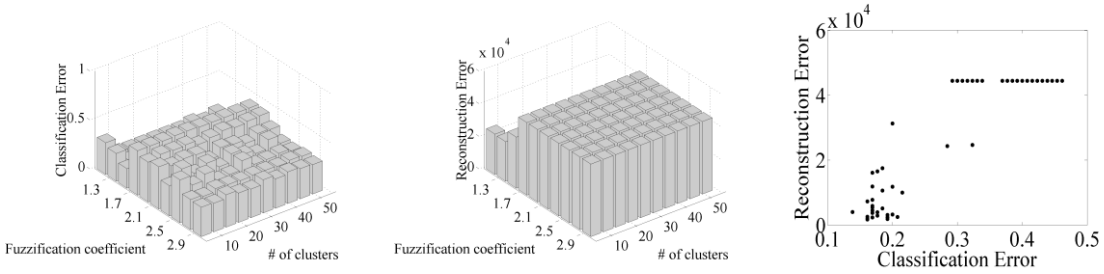
(m)


Classification Error
Fuzzification coefficient
# of clusters
Reconstruction Error
Reconstruction Error
Classification Error

(n)


Classification Error
Fuzzification coefficient
# of clusters
Reconstruction Error
x 10^4
Reconstruction Error
Classification Error

(o)



(p)

Figure 5.2 Classification error and reconstruction error of FCM based time series representation, and Comparison between classification error rate and reconstruction when the number of clusters and fuzzification coefficient take different values. (a) CBF dataset, (b) ProximalPhalanxOutlineAgeGroup dataset, (c) BeetleFly dataset, (d) BirdChicken dataset, (e) Wine dataset, (f) ECG200 dataset, (g) ToeSegmentation1 dataset, (h) ArrowHead dataset, (i) Beef dataset, (j) Trace dataset, (k) FaceFour dataset, (l) ProximalPhalanxOutlineCorrect dataset, (m) Gun-Point dataset, (n) Synthetic Control dataset, (o) Lighting-7 dataset, and (p) ToeSegmentation2 dataset.

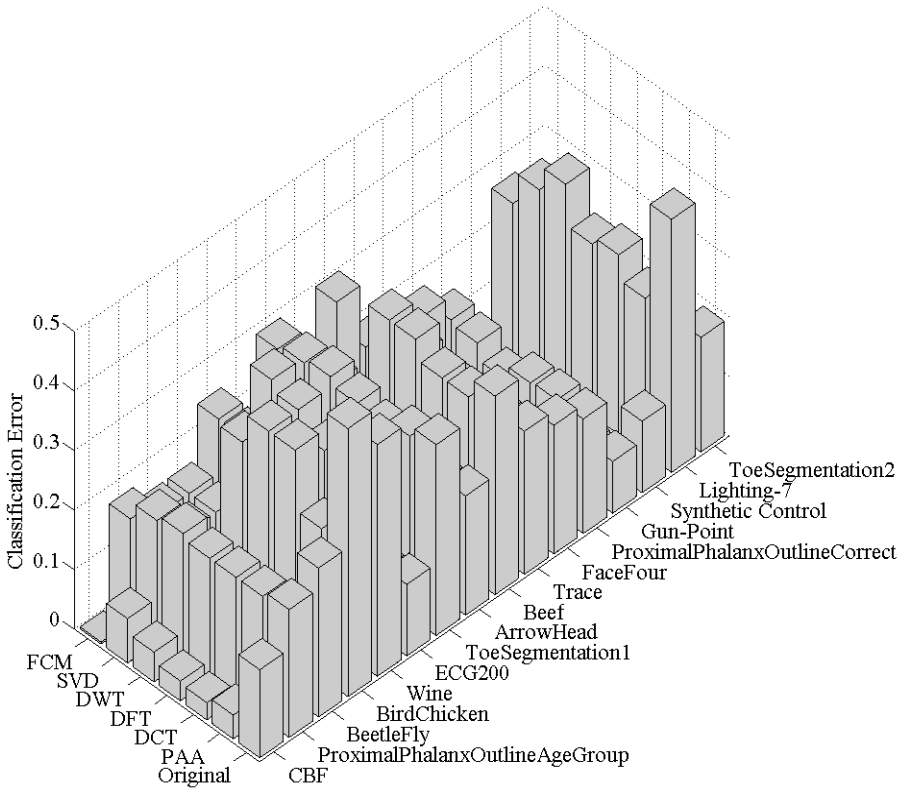


Figure 5.3 Classification error of original time series (no representation), PAA, DCT, DFT, DWT, SVD, FCM after the tuning their parameters (number of coefficients, fuzzification coefficient)

5.4 Summary

The time series approximation methods can represent the original time series in lower dimensional space, remove noise, speedup of classification process and capture its essential characteristic to improve classification performance; however, they also may eliminate some discriminative information that is essential for classification and reconstruction purposes. The thoughtful analysis of the datasets is expected before applying pertinent approximation and classification methods. Experimental results reported in this study demonstrate that the use of carefully selected approximation methods and parameters setting yields a smaller classification error rate than the one which results from the classification scheme applied to raw data.

Although there are a variety of time series approximation approaches, there is no one method that can claim to be better than other approaches because there are different characteristics and structures in time series from different domains. On the basis of

structure information and domain information, users can select the suitable time series approximation method. More information about the performance of the different experimented approaches refers to [51, 57, 80, 84].

The classifier being used in the study is a generic one and its consideration here was intentional to avoid possible bias associated with more advanced classification schemes. Some future enhancements of 1-NN classifier worth studying could be to involve applying different similarity measure such as dynamic warping time distance.

Chapter 6

A Hidden Markov Model-Based Fuzzy Modeling of Multivariate Time Series[d]

Time series modeling is a challenge and interesting research field of time series mining, which is also the third task archived in this dissertation. In this chapter, we start to elaborate on a novel Hidden Markov Model (HMM)-based fuzzy model for time series prediction. Fuzzy rules (rule-based models) are employed to describe and quantify the relationship between the input and output time series while the HMM is regarded as a vehicle for capturing efficiently the temporal behavior or changes of the multivariate time series. Essentially, the proposed strategies control the contribution of different fuzzy rules so that the proposed model can well model the dynamic behavior of time series. The use of Fuzzy C-Means clustering technique is an alternative way to construct fuzzy rules. Particle Swarm Optimization (PSO) serves as a tool to optimize the parameters of the model (e.g., the transition matrix and the emission matrix).

6.1 An Overview of Multiple Fuzzy Rule-based Model

The primary objective is to design a Hidden Markov Model based fuzzy model for multivariate time series modeling. A multivariate time series consists of more than a single univariate/individual time series which are correlated with each other. Fuzzy rule-based models can describe and reflect the relationship between the input and output time series. The underlying idea is to model the non-linear characteristics by using a set of fuzzy rules. Each rule can capture some local information about the relationships (characteristics) of multivariate time series so that the aggregation of multiple fuzzy rules results in the complex and non-linear model. In addition, the building and aggregating the rules ("if – then" statement) make the model transparent yet accurate and enhance the readability of the obtained model. The temporal correlation is another essential issue, which has to be considered in modeling multivariate time series. The HMM can be efficiently employed for portraying the temporal behavior or changes of the multivariate

time series. The design strategies differ in the type of visible state and in this study are used to control the contribution of different fuzzy rules so that the proposed model can well model the dynamic behavior of time series.

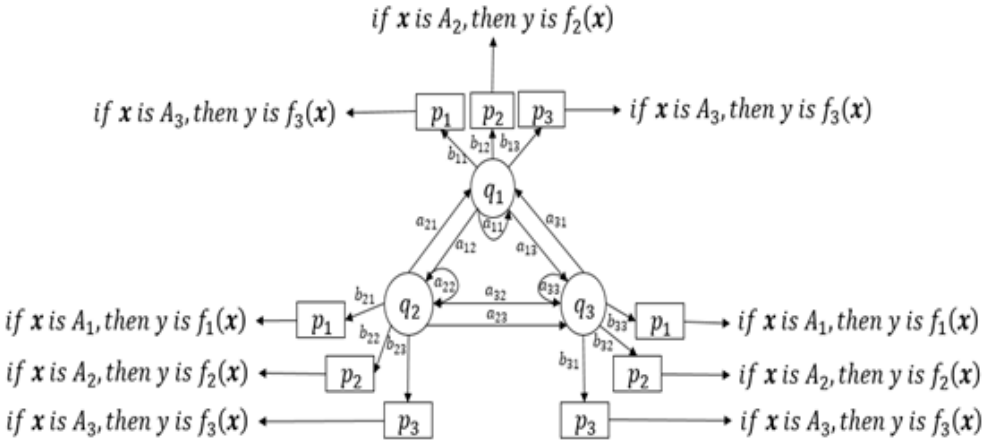


Figure 6.1 Overall scheme of the proposed time series model.

As shown in Figure 6.1, the architecture of the proposed model dwells upon the temporal dependency and relationship characteristics of multivariate time series. The process of building the overall model consists of two main steps (shown in Figure 6.2): construction of fuzzy rule-based models and the estimation of the Hidden Markov Model. The identification of the antecedent and consequent parts of the fuzzy rule-based models is realized through the usage of Fuzzy C-Means clustering approach. We perform the fuzzy clustering in the input-output space to reveal the available structure within a multivariate time series. The obtained prototypes could be sought as a structural setup for the ensuing modeling the temporal characteristic. Then we can construct an HMM on a basis of multiple fuzzy rules to describe the dynamic behavior of time series over time. In a nutshell, in order to mimic the properties of the multivariate time series, the contribution of different fuzzy rules can change with time by the HMM.

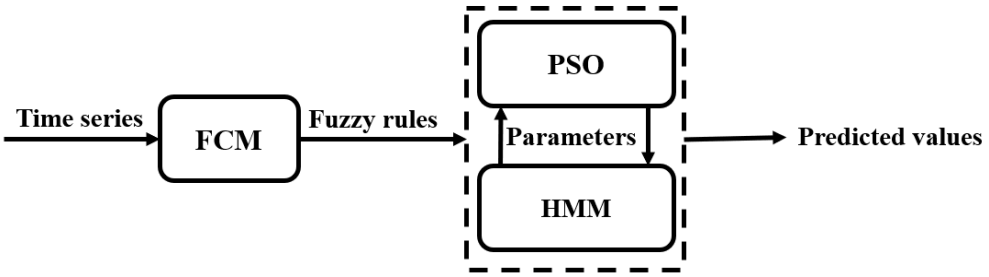


Figure 6.2 The workflow of the approach

6.2 Fundamental Development Phases

Fuzzy rule-based model:

Fuzzy rule-based models have been suggested to have better prediction performance, since its ability in describing complex, highly nonlinear models from input-output data [189]. Here, we consider the commonly encountered Takagi–Sugeno (TS) type fuzzy model [190] which has the rules in the following forms:

$$\text{if} \quad \boldsymbol{x} \quad \text{is} \quad A_i, \quad \text{then} \quad y \quad \text{is} \quad f\left(\boldsymbol{x}\right), i = 1,2,...,c \tag{6.1}$$

Where A_i denotes a fuzzy set of the i^{th} rule formed in the input space and $f_i\left(\boldsymbol{x}\right)$ is a function as a consequence of the fuzzy model. $\boldsymbol{x}$ and y are the input and output variable. c and $f_i\left(.\right)$ stand for the number of rules and the function of input variables. Through aggregating each fuzzy rule, one has the output y expressed as follows:

$$y = \frac{\sum_{i=1}^{c} A_i f_i\left(\boldsymbol{x}\right)}{\sum_{i=1}^{c} A_i} \tag{6.2}$$

There are a significant number of variants or extensions to the basic TS fuzzy model [191]. The design of these variants can be categorized into two main directions: (i) formation of premise parts (conditions) of these rules and (ii) determination of their conclusion parts. Commonly encountered fuzzy clustering can serve as a vehicle to construct conditions of the rules while linear regression stands in their corresponding conclusion parts. For more detailed information of its variants consider [191].

Prediction:

The output time series prediction is implemented based on the constructed fuzzy rules and the optimized HMM. Given the estimated transition and emission matrix, the probability of each fuzzy rule can be calculated by the following formulas:

$t = 1$:

$$\alpha_1\left(M_k\right) = \sum_{i=1}^{L} \pi_i b_{iM_k}, k = 1,2,...,c \tag{6.3}$$

$t > 1$:

$$\beta_t\left(q_j\right) = \sum_{i=1}^{L} \beta_{t-1}\left(q_i\right) a_{iq_j}, j = 1,2,...,L \tag{6.4}$$

$$\alpha_t(M_k) = \sum_{i=1}^{L} \beta_t(q_j) b_{iM_k}, k = 1, 2, ..., c \tag{6.5}$$

Here $\beta_t(q_j)$ and $\alpha_t(M_k)$ denote the probability of q_j and the probability of k^{th} fuzzy rule at the i^{th} time point.

Then, after having the prototypes $\boldsymbol{v}_k = (\boldsymbol{v}_{kx_t}, \boldsymbol{v}_{ky_t})$ generated by FCM, we can obtain the fuzzy set A_i of each fuzzy rule in the form.

$$A_k = \frac{1}{\sum_{j=1}^{c} \left(\frac{\left\| \boldsymbol{v}_{kx_t} - \boldsymbol{x}_t \right\|^2}{\left\| \boldsymbol{v}_{jx_t} - \boldsymbol{x}_t \right\|^2} \right)^{1/(m-1)}} \tag{6.6}$$

Where $\boldsymbol{v}_{kx_t}$ and $\boldsymbol{v}_{ky_t}$ are the k^{th} prototype of the input and output pair $(\boldsymbol{x}_t, y_t)$ respectively.

Once the probabilities and fuzzy sets of each fuzzy rule have been obtained, the final value of $\hat{y}_t$ is calculated as.

$$\hat{y}_t = \frac{\sum_{k=1}^{c} A_k \alpha_t(M_k) \boldsymbol{v}_{ky_t}}{\sum_{k=1}^{c} A_k \alpha_t(M_k)} \tag{6.7}$$

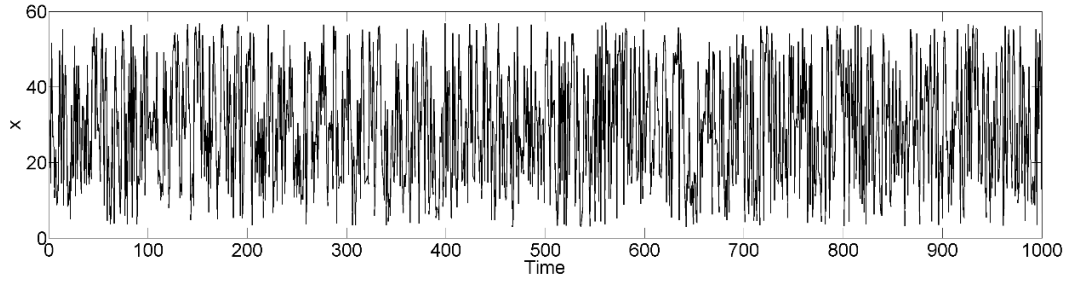
The above aggregation result $\hat{y}_t$ is regarded as the estimated value of the output time series.

6.3 Experiment and Case Studies

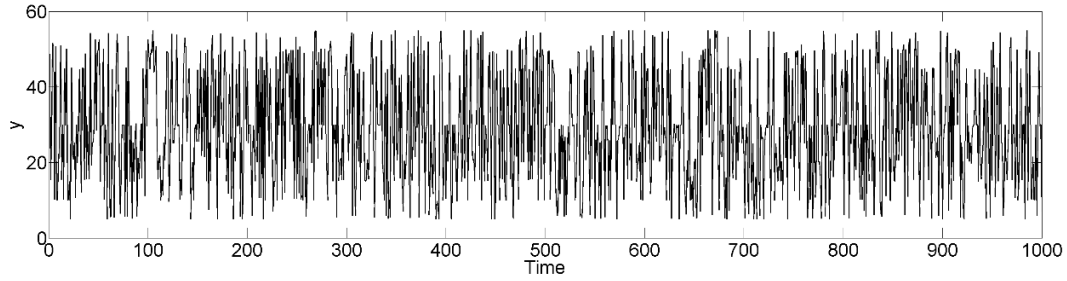
. In what follows, to demonstrate the effectiveness of the proposed model, we report experimental results for different data sets involving synthetic multivariate time series as well as real-world multivariate time series from the Data Market repository (https://datamarket.com/), the Yahoo Finance website (https://finance.yahoo.com/) and the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets.html). In particular, we are concerned whether the fuzzy rule-based model can generate the more accurate prediction with the help of the Hidden Markov Model.

6.3.1 Synthetic multivariate time series

For the clarity of explanation, a two-dimensional time series is generated with the following clustering centers and uniform distribution. Totally, there are three large clusters (denoted here as Bclsuter-1, Bcluster-2, …, Bcluster-3) and these clusters consist of 4, 3 and 3 small clusters (denoted here Sclsuter-1, Scluster-2, …, Scluster-10) respectively. Since there are 100 patterns (data) in each small cluster, the length of this multivariate time series is composed of 1,000 samples. In this study, the time series $\boldsymbol{x}$ and y denote the input and corresponding output time series respectively, as shown in Figure 6.3. And they are divided into the training set (700 time points from 1 to 700, 70%) and testing test (300 time points from 701 to 1000, 30%), which are applied to construct the models and evaluate their performance. For comparison, the fuzzy rule-based model without HMM is also exploited to estimate the value of the corresponding time series. In addition, in this study, particle swarm optimization is exploited to estimate the parameters of HMM. The number of particles and iterations are set to 200 and 300, respectively.



(a)



(b)

Figure 6.3 Two-dimensional time series (a) input time series (b) corresponding time series

Bcluster-1:

Sclsuter-1: $[v_1, w_1] = [5,5]$

Sclsuter-2: $[v_2, w_2] = [10,10]$

Sclsuter-3: $[v_3, w_3] = [15,15]$

Sclsuter-4: $[v_4, w_4] = [16,22]$

Bcluster-2:

Sclsuter-5: $[v_5, w_5] = [25,55]$

Sclsuter-6: $[v_6, w_6] = [30,50]$

Sclsuter-7: $[v_7, w_7] = [35,45]$

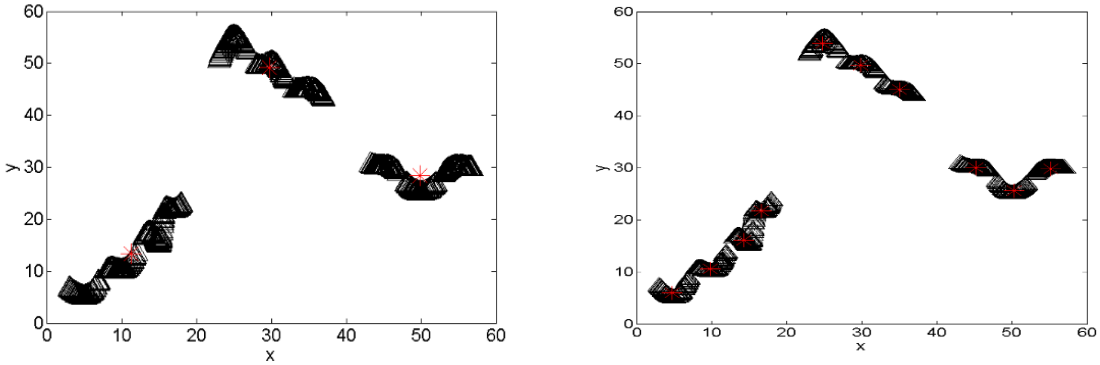
Bcluster-3:

Sclsuter-8: $[v_8, w_8] = [45,30]$

Sclsuter-9: $[v_9, w_9] = [50,25]$

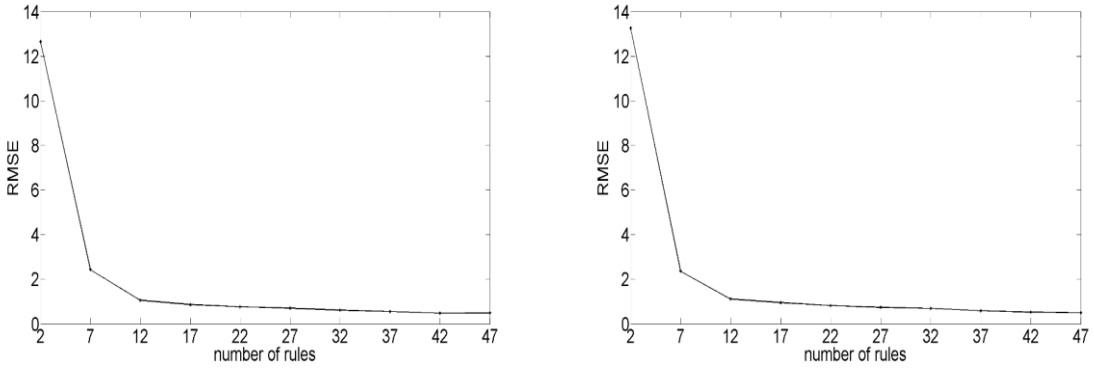
Sclsuter-10: $[v_{10}, w_{10}] = [55,30]$

There are three parameters which may influence the accuracy of estimation results in our model. Here we are mainly concerned with the impacts of the number of rules (of FCM) and number of hidden states (of HMM) because the number of visible states is equal to the number of rules. Figure 6.4. (a) and (b) show the clustering results generated by FCM when the number of prototypes is 3 and 10, respectively. As visible, FCM has been used as a vehicle to capture the structure in the data through a series of numeric prototypes.



(a) (b)

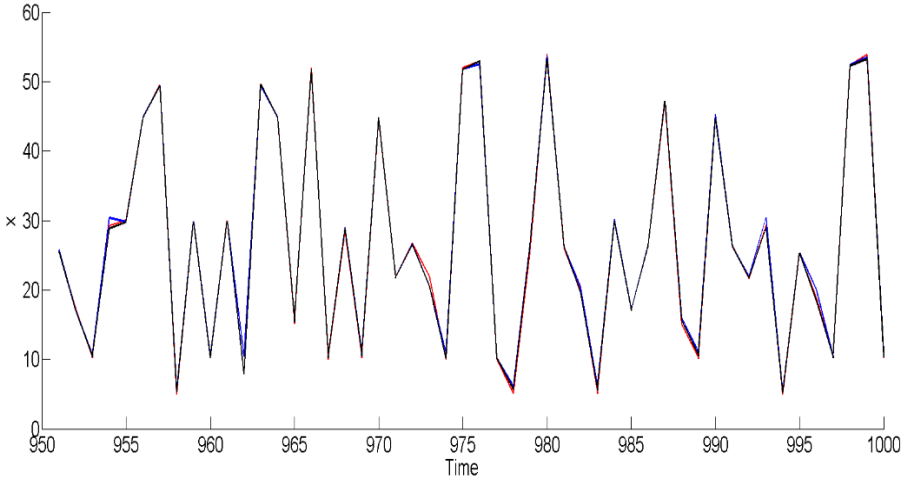
Figure 6.4 Clustering results generated by FCM when the number of prototypes is 3 and 10



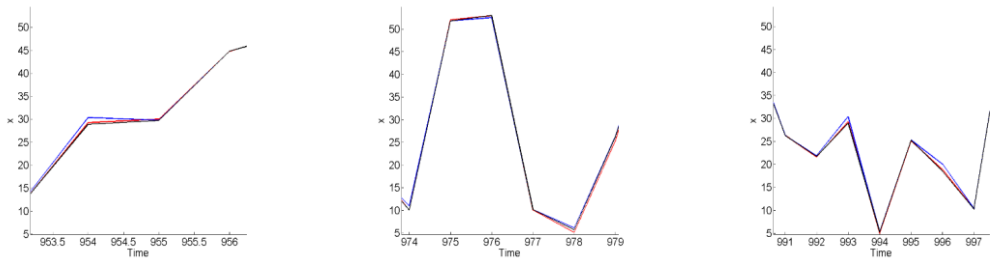
(a) (b)

Figure 6.5 Experimental results generated by the fuzzy rule-based model without HMM (a) training set (b) testing set, when the number of rules varies from 2 to 47

For all following experiments, the fuzzification coefficient was fixed at 2.0 to make the comparison with the fuzzy rules-based model (without the aid of HMM) easier and clearer. The number of rules comes from the set. the number of hidden states was ranging from 2 to 10 with a step of 1. Figure 6.5 provides the experimental results generated by the fuzzy rule-based model obtained without involving HMM.



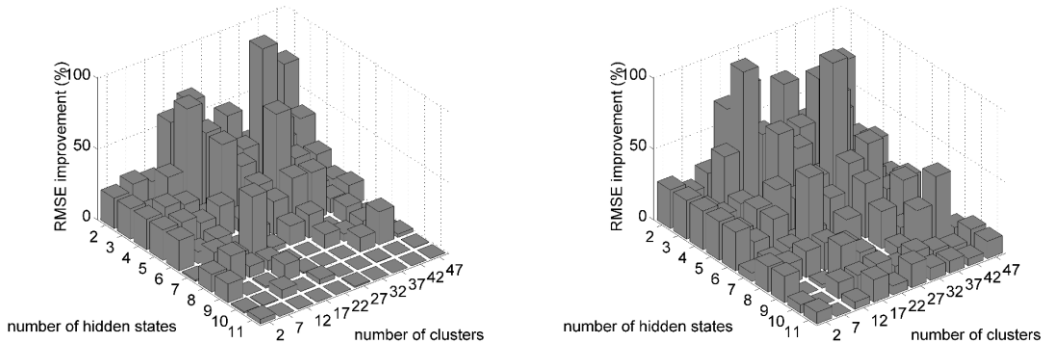
(a)



(b) (c) (d)

Figure 6.6 Experimental results of testing time series (from 951 to 1000) when the number of rules and hidden states are 32 and 4. (a) actual values (red); estimated values by the fuzzy rule-based model (blue); estimated values by HMM based fuzzy model (black); (b) enlargement of first part of (a); (c) enlargement of second part of (a); (d) enlargement of third part of (a);

There is a visible trend that the performance of the model will improve with the increase of the number of rules. The reason behind that is that more details from the time series can be captured through exacting more fuzzy rules. Figure 6.6. (a-d) provides the experimental results of testing sets from the fuzzy rule-based model and the proposed model. It is evident that the estimated values by the proposed model are closer to the real values. In contrast to the fuzzy rule-based model only, the proposed model can archive more than 90% (best) prediction performance improvement. Figure 6.7. summarized the performance improvement results when the number of hidden states and rules varies over some range.



(a) (b)

Figure 6.7 Experimental results generated by the fuzzy rule-based model with HMM (a) training set (b) testing set, when the number of rules varies from 2 to 47 and the number of hidden states varies from 2 to 100. (c) RMSE improvement of the training set. (d) RMSE improvement of the testing set.

6.3.2 Real-world multivariate time series

Here we also perform both the proposed model and the fuzzy rule-based model on the several real-world multivariate time series (from different publicly available repositories) to investigate their performance. The main characteristics of these data sets are summarized in Table 6-1.

Table 6-1 Basic characteristics of real-world datasets

Dataset	From	To	Features	Training	Testing
Precipitation of Fisher River near Dallas	Jan 1, 1988	Dec 31, 1991	2	1023	438
Rainfall in Melbourne, Australia	Jan 1, 1981	Dec 30, 1984	2	1022	438
PM 2.5	Jun 23, 2010 10:00	Aug 13, 2010 19:00	3	864	370
Istanbul Stock Exchange	Jan 5, 2009	Feb 22, 2011	7	375	161
S&P 500	Feb 12, 2013	Feb 9, 2018	2	881	378
US Dollar Exchange Rate	Mar 6, 1973	Mar 1, 1977	2	700	300
Temperature change of Oldman River near Brocket	Jan 1, 1988	Dec 31, 1991	2	1023	438
Precipitation change of	Jan 1,	Nov 1,	2	980	420

Saugeen River near Port Elgin	1988	1991

We present comparative results of these two models in Figure 6.8, Figure 6.9 and Table 6-2. Through the experimental results, we arrive at some interesting and important findings:

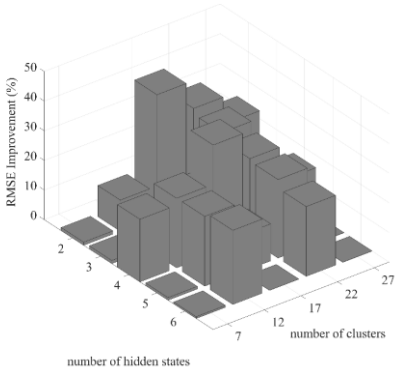
1. Compared with the fuzzy model without the aid of HMM, the HMM based fuzzy model can produce better prediction results producing lower values of the corresponding criterion (RMSE), which results in that HMM can model the temporal changes of multivariate time series.

2. Increasing the number of fuzzy rules will achieve more accurate prediction with lower RMSE value in most cases. The reason behind the performance improvement may be linked with the fuzzy clustering in the input-output space. Obviously, the growth of clustering centers (or prototypes) means that more structured or relevant information is captured by FCM clustering. It will provide more effective assistance for the ensuing construction of fuzzy rules and fuzzy sets, which is very important for modeling multivariate time series.

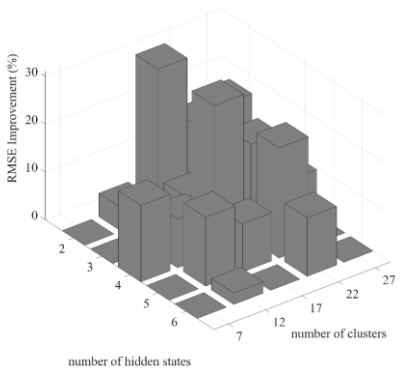
3. Increasing the number of hidden or unknown states might produce a limited improvement of the prediction in some cases. This is because there are too many potential solutions that it is difficult for PSO to obtain the global optimal solution with the relatively low computation cost.

4. The experimental results also indicated that the proposed model showed more performance improvement on some datasets while in others it seems to have relatively limited improvement. In practice, little real-world multivariate time series can meet the assumptions (mentioned in Section 3) of HMM. The weakness of HMM limits the predictive ability of the proposed model.

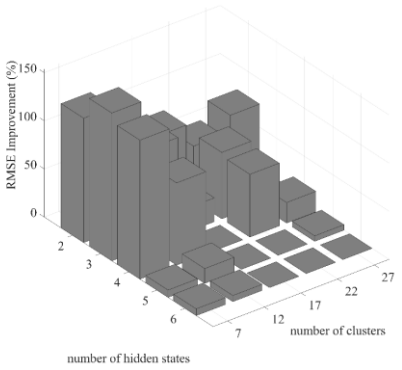
To sum up, the above findings suggest that the proposed HMM based fuzzy model is much more promising and effective alternative than the fuzzy rule-based model. The performance improvement results in that the temporal changes are captured by HMM. We also demonstrate that the proposed approach works in practice by performing it on different real-world time series.


RMSE Improvement (%)
50
40
30
20
10
0
2
3
4
5
6
number of hidden states
7
12
17
22
27
number of clusters

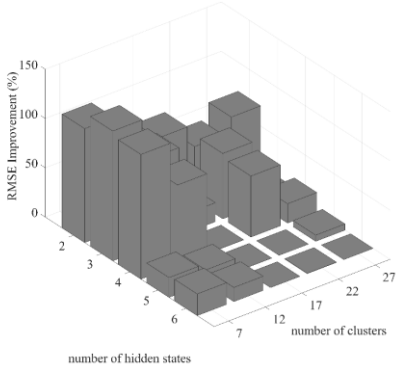
(a)


RMSE Improvement (%)
30
20
10
0
2
3
4
5
6
number of hidden states
7
12
17
22
27
number of clusters

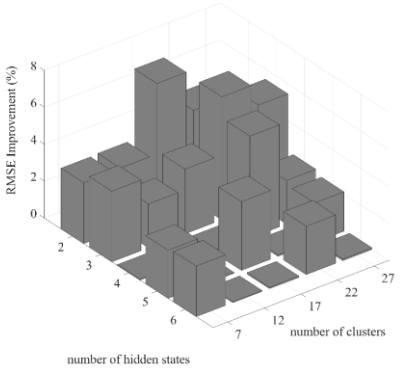
(b)


RMSE Improvement (%)
150
100
50
0
2
3
4
5
6
number of hidden states
7
12
17
22
27
number of clusters

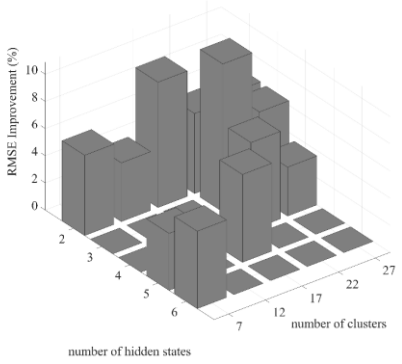
(c)


RMSE Improvement (%)
150
100
50
0
2
3
4
5
6
number of hidden states
7
12
17
22
27
number of clusters

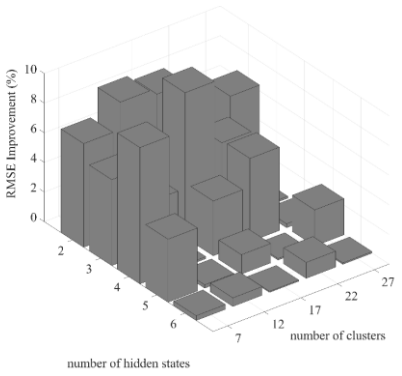
(d)


RMSE Improvement (%)
8
6
4
2
0
2
3
4
5
6
number of hidden states
7
12
17
22
27
number of clusters

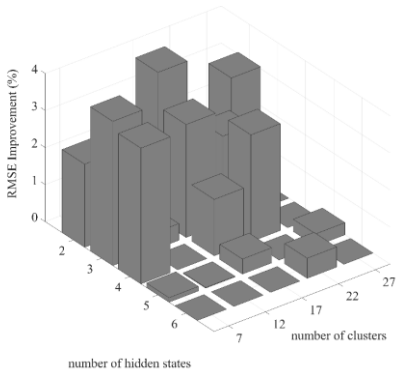
(e)


RMSE Improvement (%)
10
8
6
4
2
0
2
3
4
5
6
number of hidden states
7
12
17
22
27
number of clusters

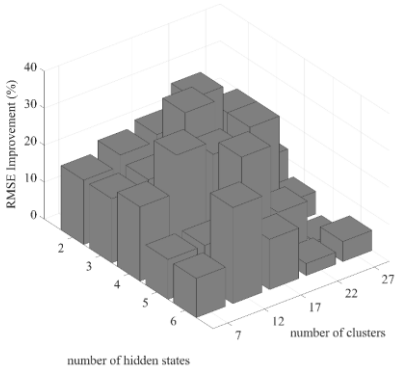
(f)


RMSE Improvement (%)
number of hidden states
number of clusters

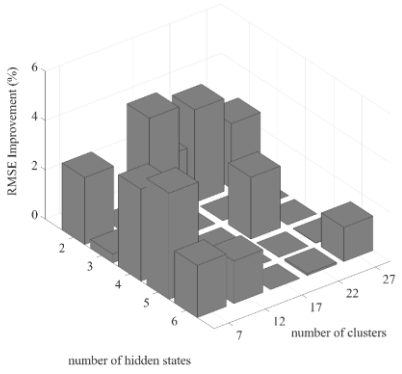
(g)


RMSE Improvement (%)
number of hidden states
number of clusters

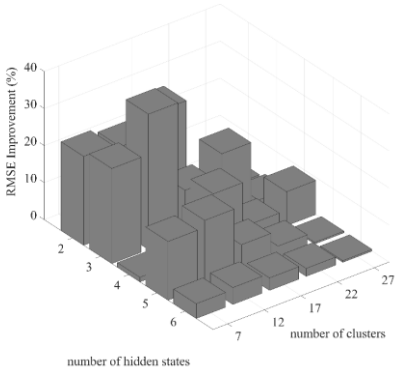
(h)


RMSE Improvement (%)
number of hidden states
number of clusters

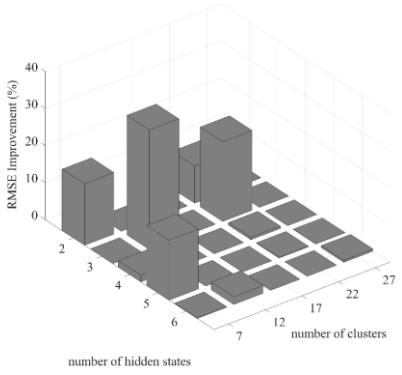
(i)


RMSE Improvement (%)
number of hidden states
number of clusters

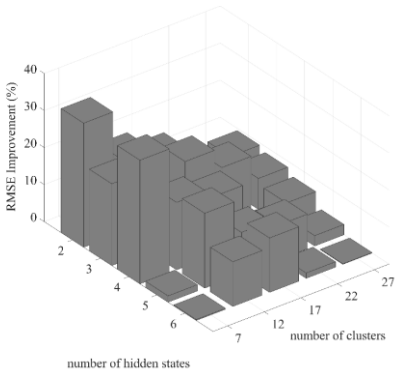
(j)


RMSE Improvement (%)
number of hidden states
number of clusters

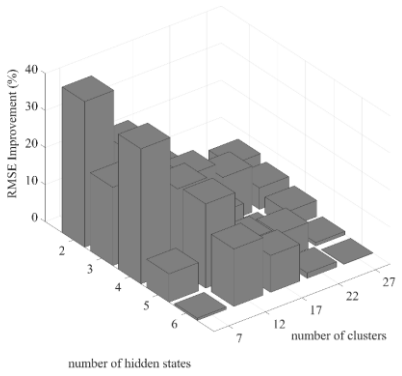
(k)


RMSE Improvement (%)
number of hidden states
number of clusters

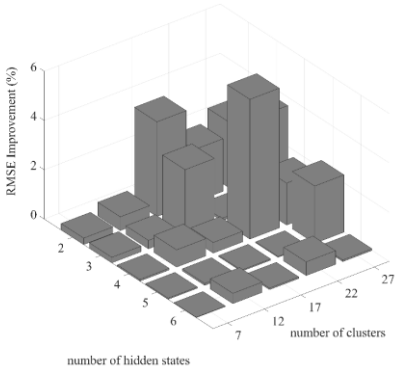
(l)



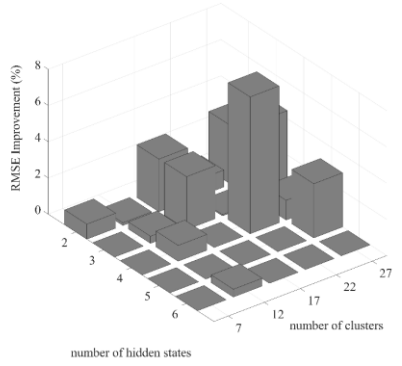
(m)



(n)



(o)



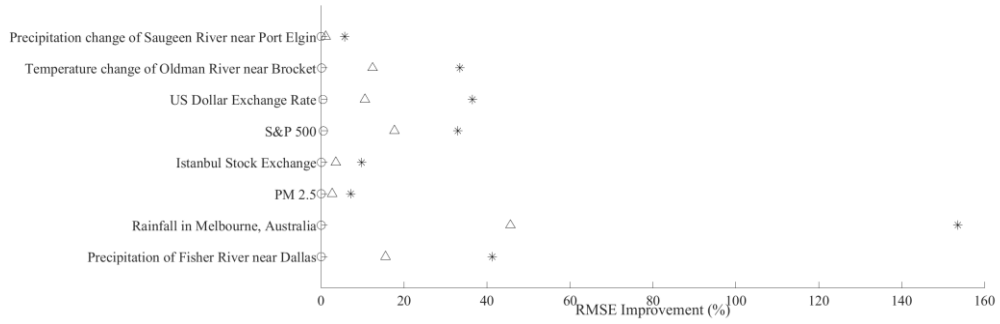
(p)

Figure 6.8 Experimental results (RMSE improvement) of real-world time series (a) training set (b) testing set.

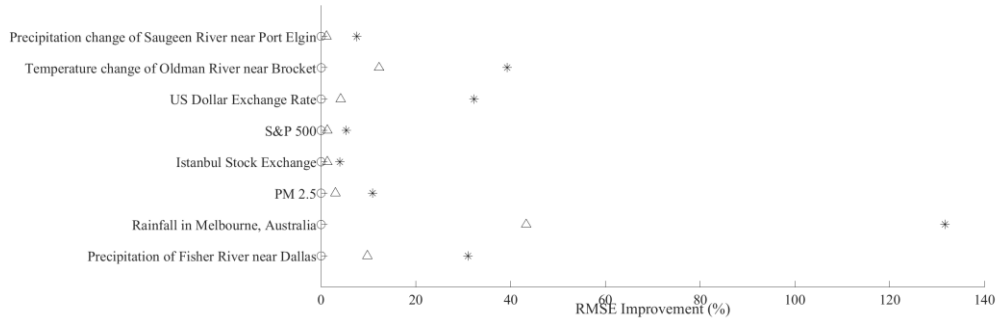
Table 6-2 The improvement (best) of RMSE from training sets and testing sets

Datasets	Training set (%)	Testing set (%)
Precipitation of Fisher River near Dallas	41.3069	31.0057
Rainfall in Melbourne, Australia	153.5939	131.6728
PM 2.5	7.1465	10.8829
Istanbul Stock Exchange	9.7674	3.9625
S&P 500	33.0264	5.2978
US Dollar Exchange Rate	36.5345	32.2558
Temperature change of Oldman	33.4962	39.3174

River near Brocket		
Precipitation change of Saugeen River near Port Elgin	5.6700	7.5545



(a)



(b)

Figure 6.9 Improvement of prediction performance of real-world data sets: (a) training set; (b) testing set (maximum, average and minimum values marked by star, triangle and circle).

6.4 Summary

In this study, we have introduced the HMM-based fuzzy model that models the multivariate time series with the aid of fuzzy rules and HMM for prediction. Fuzzy C-means algorithm, which reveals the available structure in the input-output space, is used to construct the fuzzy rules. A set of fuzzy rules describes the correlation between the input and output time series. The temporal dependency of time points is captured with the use of

HMM. Afterwards, parametric learning of the HMM is realized through PSO algorithm. Several experiments involving synthetic and real-world multivariate time series are completed to demonstrate that the proposed model outperforms the fuzzy rule-based model. We also have noticed that the limitation results in the assumptions of HMM. As a future work, high order HMM or other techniques which can model the temporal characteristics might be worth investigating.

Chapter 7

Conclusions and Future Studies

This dissertation is concerned with three core time series mining tasks, which are time series anomaly detection, approximation/representation, and modeling respectively. We construct and investigate the performance of the proposed models by using a series of synthetic and publicly available multivariate time series. Overall, the approaches introduced in this thesis exhibit some impressive highlights.

7.1 Major conclusions

Time series anomaly detection

Compared with the traditional FCM, the augmented FCM can reveal the available data structure more efficiently for the subsequent anomaly detection because of the different impact of each variable in evaluating the similarity between multivariate time series. The proposed detector with the augmented FCM can offer more accurate detection results.

The proposed cluster-centric time series anomaly detection frameworks are able to detect the anomalies caused by relations between time series in multivariate time series. The reason is that all univariate subsequences of multivariate subsequence are considered at the same time to determine an anomaly score of this multivariate subsequence.

As for the proposed HMM-based anomaly detectors with different transformation methods, the accuracy improvement of FCM, Sugeno fuzzy integral, Choquet fuzzy integral based detectors vis-a-vis the generic PCA-based detector is quite apparent (around 7-9 % improvement). This is related to the fact that more useful information is contained in the transformation to an observed sequence.

Additionally, experimental results suggest that the fuzzification coefficient and the number of clusters (or observed states) are also associated with the performance of these detectors. Specifically, the increase in the number of clusters will affect the performance of the detector significantly. The fuzzification coefficient exhibits some impact on the performance of the detector.

Time series approximation/representation

The time series approximation methods can represent the original time series in lower dimensional space, remove noise, speedup of classification process and capture its essential characteristic to improve classification performance; however, they also may eliminate some discriminative information that is essential for classification and reconstruction purposes. In spite of the diversity of the existing methods used in time series representation and classification, it is quite uncommon to encounter studies that report on the relationships between the classification error and the representation properties (e.g., reconstruction quality) of different time series representations. We have demonstrated experimentally that the classification error will be smaller than that of original time series when taking suitable approximation methods and parameters setting. The approximation methods can eliminate some useless information (e.g. noise) of time series to archive better classification results. In some sense, it can illustrate why the classification error goes down and then up with the increase of the number of coefficients. The reconstruction error will decrease towards zero with increasing the number of coefficients in a number of cases. As for FCM based approximation method, the fuzzification coefficient exhibits a significant impact on the performance of classification and reconstruction.

Time series modeling

Compared with the fuzzy model without the aid of HMM, the HMM-based fuzzy model can produce better prediction results producing lower values of the corresponding criterion (RMSE), which results in that HMM can model the temporal changes of multivariate time series. The growth of clustering centers (or prototypes) means that more structured or relevant information is captured by FCM clustering. It will provide more effective assistance for the ensuing construction of fuzzy rules and fuzzy sets, which is very important for modeling multivariate time series. The proposed HMM based fuzzy model is much more promising and effective alternative than the fuzzy rule-based model.

Increasing the number of fuzzy rules will achieve more accurate prediction with lower RMSE value in most cases. The reason behind the performance improvement may be linked with the fuzzy clustering in the input-output space. Obviously, the growth of clustering centers (or prototypes) means that more structured or relevant information is captured by FCM clustering. It will provide more effective assistance for the ensuing

construction of fuzzy rules and fuzzy sets, which is very important for modeling multivariate time series.

7.2 Future Studies

The proposed frameworks and techniques could be developed for the future extension as follows:

Time series anomaly detection

Although the proposed cluster-centric time series anomaly detection frameworks show good performance, it comes with some limitations. One of them is an intensive computing overhead resulting from the fact that the Euclidean distance function calls for a substantial level of computing. Considering the computing overhead associated with the analysis of different combinations of the entries of λ, one may consider a simplified version of the method with some predetermined values of the entries of the vector of these parameters. The other alternative is to try more up to date objective optimization techniques to obtain the different weight of each component in multivariate time series.

As for the HMM detectors, intensive computing, especially in case of fuzzy integral based detectors, is also a significant limitation. To overcome this problem, a certain alternative would be to engage experts in specifying some initial values of degrees of importance. Additionally, in this study, the proposed approach has only concentrated on amplitude anomalies in multivariate time series. Therefore, detecting other types of anomalies (e.g., shape anomalies) for larger datasets is a useful further direction. Another pursuit worth investigating is to quantify information loss when transforming from multivariate time series to univariate time series. A criterion or index which is used to measure the information loss is expected.

Time series approximation/representation

The classifier being used in the study is a generic 1-NN classifier and its consideration here was intentional to avoid possible bias associated with more advanced classification schemes. Some future enhancements of 1-NN classifier worth studying could be to involve applying different similarity measure such as dynamic warping time distance. Additionally, one of the primary objectives of the approximation methods is to eliminate some useless information (e.g. noise) of time series and retain the useful information for subsequent

classifiers. Motivated by this, it would be an interest is to further identify useful information of time series by investigating their shape and aptitude.

Time series modeling

The experimental results indicated that the proposed model showed more performance improvement on some datasets while in others it seems to have relatively limited improvement. In practice, little real-world multivariate time series can meet the assumptions (mentioned in Section 6) of HMM. The weakness of HMM limits the predictive ability of the proposed model. High order HMM or other techniques (e.g. Conditional random field) which can also model the temporal characteristics might be worth investigating.

Bibliography

[1] C.-C. Wang, Y. Kang, P.-C. Shen, Y.-P. Chang, and Y.-L. Chung, "Applications of fault diagnosis in rotating machinery by using time series analysis with neural network," *Expert Systems with Applications,* vol. 37, no. 2, pp. 1696-1702, 2010.

[2] M. H. Amini, A. Kargarian, and O. Karabasoglu, "ARIMA-based decoupled time series forecasting of electric vehicle charging demand for stochastic power system operation," *Electric Power Systems Research,* vol. 140, pp. 378-390, 2016.

[3] J. Mason, N. Freemantle, and G. Browning, "Impact of Effective Health Care bulletin on treatment of persistent glue ear in children: time series analysis," *Bmj,* vol. 323, no. 7321, pp. 1096-1097, 2001.

[4] M. J. Woods, C. J. Russell, R. J. Davy, and P. A. Coppin, "Simulation of wind power at several locations using a measured time-series of wind speed," *IEEE Transactions on Power Systems,* vol. 28, no. 1, pp. 219-226, 2013.

[5] R. Ak, O. Fink, and E. Zio, "Two machine learning approaches for short-term wind speed time-series prediction," *IEEE transactions on neural networks and learning systems,* vol. 27, no. 8, pp. 1734-1747, 2016.

[6] K. Yunus, T. Thiringer, and P. Chen, "ARIMA-based frequency-decomposed modeling of wind speed time series," *IEEE Transactions on Power Systems,* vol. 31, no. 4, pp. 2546-2556, 2016.

[7] K. Xie, Q. Liao, H.-M. Tai, and B. Hu, "Non-Homogeneous Markov Wind Speed Time Series Model Considering Daily and Seasonal Variation Characteristics," *IEEE Transactions on Sustainable Energy,* vol. 8, no. 3, pp. 1281-1290, 2017.

[8] M. Chaouch, "Clustering-based improvement of nonparametric functional time series forecasting: Application to intra-day household-level load curves," *IEEE Transactions on Smart Grid,* vol. 5, no. 1, pp. 411-419, 2014.

[9] P.-F. Marteau and S. Gibet, "On recursive edit distance kernels with application to time series classification," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 26, no. 6, pp. 1121-1133, 2015.

[10] J. P. González, A. M. San Roque, and E. A. Pérez, "Forecasting functional time series with a new Hilbertian ARMAX model: Application to electricity price forecasting," *IEEE Transactions on Power Systems,* vol. 33, no. 1, pp. 545-556, 2018.

[11] C. Zhang, Y. Yan, A. Narayan, and H. Yu, "Practically Oriented Finite-Time Control Design and Implementation: Application to a Series Elastic Actuator," *IEEE Transactions on Industrial Electronics,* vol. 65, no. 5, pp. 4166-4176, 2018.

[12] J. K. Pant and S. Krishnan, "Compressive sensing of foot gait signals and its application for the estimation of clinically relevant time series," *IEEE Transactions on Biomedical Engineering,* vol. 63, no. 7, pp. 1401-1415, 2016.

[13] K. Yan, Z. Ji, H. Lu, J. Huang, W. Shen, and Y. Xue, "Fast and Accurate Classification of Time Series Data Using Extended ELM: Application in Fault Diagnosis of Air Handling Units," *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* 2017.

[14] A. M. Atto, E. Trouvé, J.-M. Nicolas, and T. T. Lê, "Wavelet Operators and Multiplicative Observation Models—Application to SAR Image Time-Series Analysis," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 54, no. 11, pp. 6606-6624, 2016.

[15] J. Zhang, Z. Wei, Z. Yan, M. Zhou, and A. Pani, "Online Change-Point Detection in Sparse Time Series With Application to Online Advertising," *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* 2017.

[16] Z. Malkin, "Application of the allan variance to time series analysis in astrometry and geodesy: A review," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control,* vol. 63, no. 4, pp. 582-589, 2016.

[17] S. Yang and J. Liu, "Time Series Forecasting based on High-Order Fuzzy Cognitive Maps and Wavelet Transform," *IEEE Transactions on Fuzzy Systems,* 2018.

[18] O. Yazdanbakhsh and S. Dick, "Forecasting of Multivariate Time Series via Complex Fuzzy Logic," *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* vol. 47, no. 8, pp. 2160-2171, 2017.

[19] S. Jahandari, A. Kalhor, and B. N. Araabi, "Online Forecasting of Synchronous Time Series Based on Evolving Linear Models," *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* no. 99, pp. 1-12, 2018.

[20] J. Mei, Y. De Castro, Y. Goude, J.-M. Azaïs, and G. Hébrail, "Nonnegative matrix factorization with side information for time series recovery and prediction," *IEEE Transactions on Knowledge and Data Engineering,* 2018.

[21] H. Izakian and W. Pedrycz, "Anomaly detection and characterization in spatial time series data: A cluster-centric approach," *IEEE Transactions on Fuzzy Systems,* vol. 22, no. 6, pp. 1612-1624, 2014.

[22] C. Huang, G. Min, Y. Wu, Y. Ying, K. Pei, and Z. Xiang, "Time Series Anomaly Detection for Trustworthy Services in Cloud Computing Systems," *IEEE Transactions on Big Data,* 2017.

[23] S. Jin, Z. Zhang, K. Chakrabarty, and X. Gu, "Towards Predictive Fault Tolerance in a Core-Router System: Anomaly Detection Using Correlation-Based Time-Series Analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* 2017.

[24] H. Wang, M. Tang, Y. Park, and C. E. Priebe, "Locality statistics for anomaly detection in time series of graphs," *IEEE Transactions on Signal Processing,* vol. 62, no. 3, pp. 703-717, 2014.

[25] F. Rasheed and R. Alhajj, "A framework for periodic outlier pattern detection in time-series sequences," *IEEE transactions on cybernetics,* vol. 44, no. 5, pp. 569-582, 2014.

[26] P. Galeano, D. Peña, and R. S. Tsay, "Outlier detection in multivariate time series by projection pursuit," *Journal of the American Statistical Association,* vol. 101, no. 474, pp. 654-669, 2006.

[27] G. M. Ljung, "On outlier detection in time series," *Journal of the Royal Statistical Society. Series B (Methodological),* pp. 559-567, 1993.

[28] S. C. Chin, A. Ray, and V. Rajagopalan, "Symbolic time series analysis for anomaly detection: a comparative evaluation," *Signal Processing,* vol. 85, no. 9, pp. 1859-1868, 2005.

[29] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR),* vol. 41, no. 3, p. 15, 2009.

[30] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection for discrete sequences: A survey," *IEEE Transactions on Knowledge and Data Engineering,* vol. 24, no. 5, pp. 823-839, 2012.

[31] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: a survey," *Data Mining and Knowledge Discovery,* vol. 29, no. 3, pp. 626-688, 2015.

[32] S. Kumar and S. S. Gangwar, "Intuitionistic Fuzzy Time Series: An Approach for Handling Nondeterminism in Time Series Forecasting," *IEEE Transactions on Fuzzy Systems,* vol. 24, no. 6, pp. 1270-1281, 2016.

[33] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE transactions on neural networks,* vol. 5, no. 2, pp. 240-254, 1994.

[34] R. Chandra, "Competition and collaboration in cooperative coevolution of Elman recurrent neural networks for time-series prediction," *IEEE transactions on neural networks and learning systems,* vol. 26, no. 12, pp. 3123-3136, 2015.

[35] D. T. Mirikitani and N. Nikolaev, "Recursive bayesian recurrent neural networks for time-series modeling," *IEEE Transactions on Neural Networks,* vol. 21, no. 2, pp. 262-274, 2010.

[36] K. Lukoseviciute and M. Ragulskis, "Evolutionary algorithms for the selection of time lags for time series forecasting by fuzzy inference systems," *Neurocomputing,* vol. 73, no. 10-12, pp. 2077-2088, 2010.

[37] V. Ravi, D. Pradeepkumar, and K. Deb, "Financial time series prediction using hybrids of chaos theory, multi-layer perceptron and multi-objective evolutionary algorithms," *Swarm and Evolutionary Computation,* vol. 36, pp. 136-149, 2017.

[38] N. I. Sapankevych and R. Sankar, "Time series prediction using support vector machines: a survey," *IEEE Computational Intelligence Magazine,* vol. 4, no. 2, 2009.

[39] A. S. Weigend, *Time series prediction: forecasting the future and understanding the past*. Routledge, 2018.

[40] P. J. Brockwell and R. A. Davis, *Introduction to time series and forecasting*. springer, 2016.

[41] A. Grinsted, J. C. Moore, and S. Jevrejeva, "Application of the cross wavelet transform and wavelet coherence to geophysical time series," *Nonlinear processes in geophysics,* vol. 11, no. 5/6, pp. 561-566, 2004.

[42] G. Sugihara and R. M. May, "Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series," *Nature,* vol. 344, no. 6268, p. 734, 1990.

[43] P. Protopapas, J. Giammarco, L. Faccioli, M. Struble, R. Dave, and C. Alcock, "Finding outlier light curves in catalogues of periodic variable stars," *Monthly Notices of the Royal Astronomical Society,* vol. 369, no. 2, pp. 677-696, 2006.

[44] M. Das and S. Parthasarathy, "Anomaly detection and spatio-temporal analysis of global climate system," in *Proceedings of the third international workshop on knowledge discovery from sensor data*, 2009, pp. 142-150: ACM.

[45] H. Izakian and W. Pedrycz, "Anomaly detection in time series data using a fuzzy c-means clustering," in *IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint*, 2013, pp. 1513-1518: IEEE.

[46] V. Chandola, V. Mithal, and V. Kumar, "Comparative evaluation of anomaly detection techniques for sequence data," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, 2008, pp. 743-748: IEEE.

[47] D. Gao, Y. Kinouchi, K. Ito, and X. Zhao, "Neural networks for event extraction from time series: a back propagation algorithm approach," *Future Generation Computer Systems,* vol. 21, no. 7, pp. 1096-1105, 2005.

[48] J. Ma and S. Perkins, "Time-series novelty detection using one-class support vector machines," in *Neural Networks, 2003. Proceedings of the International Joint Conference on*, 2003, vol. 3, pp. 1741-1745: IEEE.

[49] S. Staniford, J. A. Hoagland, and J. M. McAlerney, "Practical automated detection of stealthy portscans," *Journal of Computer Security,* vol. 10, no. 1-2, pp. 105-136, 2002.

[50] H. N. Akouemo and R. J. Povinelli, "Probabilistic anomaly detection in natural gas time series data," *International Journal of Forecasting,* vol. 32, no. 3, pp. 948-956, 2016.

[51] 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*, 2004, pp. 206-215: ACM.

[52] D. Rafiei and A. Mendelzon, "Similarity-based queries for time series data," in *ACM SIGMOD Record*, 1997, vol. 26, no. 2, pp. 13-25: ACM.

[53] E. Frentzos, K. Gratsias, and Y. Theodoridis, "Index-based most similar trajectory search," in *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, 2007, pp. 816-825: IEEE.

[54] M. López-García, R. Ramos-Lara, O. Miguel-Hurtado, and E. Cantó-Navarro, "Embedded system for biometric online signature verification," *IEEE Transactions on industrial informatics,* vol. 10, no. 1, pp. 491-501, 2014.

[55] M. A. Simão, P. Neto, and O. Gibaru, "Unsupervised gesture segmentation by motion detection of a real-time data stream," *IEEE Transactions on Industrial Informatics,* vol. 13, no. 2, pp. 473-481, 2017.

[56] N. Takeishi and T. Yairi, "Anomaly detection from multivariate time-series with sparse representation," in *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on*, 2014, pp. 2651-2656: IEEE.

[57] D. Yankov, E. Keogh, and U. Rebbapragada, "Disk aware discord discovery: Finding unusual time series in terabyte sized datasets," *Knowledge and Information Systems,* vol. 17, no. 2, pp. 241-262, 2008.

[58] C. C. Aggarwal, "Outlier analysis," in *Data mining*, 2015, pp. 237-263: Springer.

[59] T. W. Liao, "Clustering of time series data—a survey," *Pattern recognition,* vol. 38, no. 11, pp. 1857-1874, 2005.

[60] H. Izakian, W. Pedrycz, and I. Jamal, "Clustering spatiotemporal data: An augmented fuzzy c-means," *IEEE transactions on fuzzy systems,* vol. 21, no. 5, pp. 855-868, 2013.

[61] J. Ren, H. Li, C. Hu, and H. He, "ODMC: Outlier Detection on Multivariate Time Series Data based on Clustering," *Journal of Convergence Information Technology,* vol. 6, no. 2, pp. 70-77, 2011.

[62] Z. Qiao, J. He, J. Cao, G. Huang, and P. Zhang, "Multiple time series anomaly detection based on compression and correlation analysis: a medical surveillance case study," in *Asia-Pacific Web Conference*, 2012, pp. 294-305: Springer.

[63] X. Wang, "Two-phase outlier detection in multivariate time series," in *Fuzzy Systems and Knowledge Discovery (FSKD), 2011 Eighth International Conference on*, 2011, vol. 3, pp. 1555-1559: IEEE.

[64] J. Li, W. Pedrycz, and I. Jamal, "Multivariate time series anomaly detection: A framework of Hidden Markov Models," *Applied Soft Computing,* vol. 60, pp. 229-240, 2017.

[65] H. Cheng, P.-N. Tan, C. Potter, and S. Klooster, "Detection and characterization of anomalies in multivariate time series," in *Proceedings of the 2009 SIAM International Conference on Data Mining*, 2009, pp. 413-424: SIAM.

[66] M. E. O. S. Parthasarathy, "A dissimilarity measure for comparing subsets of data: Application to multivariate time series," *Temporal data mining: algorithms, theory and applications (TDM 2005),* p. 101, 2005.

[67] S. Bay, K. Saito, N. Ueda, and P. Langley, "A framework for discovering anomalous regimes in multivariate time-series data with local models," in *Symposium on Machine Learning for Anomaly Detection, Stanford, USA*, 2004.

[68] H. Qiu, Y. Liu, N. A. Subrahmanya, and W. Li, "Granger causality for time-series anomaly detection," in *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, 2012, pp. 1074-1079: IEEE.

[69] F. Tu, S. S. Ge, Y. Tang, and C. C. Hang, "Robust Visual Tracking via Collaborative Motion and Appearance Model," *IEEE Transactions on Industrial Informatics,* vol. 13, no. 5, pp. 2251-2259, 2017.

[70] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[71] L.-J. Cao and F. E. H. Tay, "Support vector machine with adaptive parameters in financial time series forecasting," *IEEE Transactions on neural networks,* vol. 14, no. 6, pp. 1506-1518, 2003.

[72] L. Mascolo, J. M. Lopez-Sanchez, F. Vicente-Guijalba, F. Nunziata, M. Migliaccio, and G. Mazzarella, "A complete procedure for crop phenology estimation with PolSAR data based on the complex Wishart classifier," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 54, no. 11, pp. 6505-6515, 2016.

[73] S. K. Yoo, S. L. Cotton, and W. G. Scanlon, "Switched Diversity Techniques for Indoor Off-Body Communication Channels: An Experimental Analysis and Modeling," *IEEE Transactions on Antennas and Propagation,* vol. 64, no. 7, pp. 3201-3206, 2016.

[74] Q. Gao, T. C. Lee, and C. Y. Yau, "Nonparametric modeling and break point detection for time series signal of counts," *Signal Processing,* vol. 138, pp. 307-312, 2017.

[75] W. Lian, R. Talmon, H. Zaveri, L. Carin, and R. Coifman, "Multivariate time-series analysis and diffusion maps," *Signal Processing,* vol. 116, pp. 13-28, 2015.

[76] D. Huang, H. Zareipour, W. D. Rosehart, and N. Amjady, "Data mining for electricity price classification and the application to demand-side management," *IEEE Transactions on Smart Grid,* vol. 3, no. 2, pp. 808-817, 2012.

[77] X. Huang, L. Shi, and J. A. Suykens, "Support vector machine classifier with pinball loss," *IEEE transactions on pattern analysis and machine intelligence,* vol. 36, no. 5, pp. 984-997, 2014.

[78] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE transactions on systems, man, and cybernetics,* vol. 21, no. 3, pp. 660-674, 1991.

[79] A. Saxena and A. Saad, "Evolving an artificial neural network classifier for condition monitoring of rotating mechanical systems," *Applied Soft Computing,* vol. 7, no. 1, pp. 441-454, 2007.

[80] X. Xi, E. Keogh, C. Shelton, L. Wei, and C. A. Ratanamahatana, "Fast time series classification using numerosity reduction," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 1033-1040: ACM.

[81] J. J. Rajan and P. J. Rayner, "Unsupervised time series classification," *Signal processing,* vol. 46, no. 1, pp. 57-74, 1995.

[82] J. Sulam, Y. Romano, and R. Talmon, "Dynamical system classification with diffusion embedding for ECG-based person identification," *Signal Processing,* vol. 130, pp. 403-411, 2017.

[83] Y. Wen, K. Mukherjee, and A. Ray, "Adaptive pattern classification for symbolic dynamic systems," *Signal Processing,* vol. 93, no. 1, pp. 252-260, 2013.

[84] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, 2003, pp. 2-11: ACM.

[85] S. Papadopoulos, L. Wang, Y. Yang, D. Papadias, and P. Karras, "Authenticated multistep nearest neighbor search," *IEEE Transactions on Knowledge and Data Engineering,* vol. 23, no. 5, pp. 641-654, 2011.

[86] O. Edfors, M. Sandell, J.-J. Van de Beek, S. K. Wilson, and P. O. Borjesson, "OFDM channel estimation by singular value decomposition," *IEEE Transactions on communications,* vol. 46, no. 7, pp. 931-939, 1998.

[87] K. Ben-Kilani and M. Elleuch, "Structural analysis of voltage stability in power systems integrating wind power," *IEEE Transactions on Power Systems,* vol. 28, no. 4, pp. 3785-3794, 2013.

[88] J. J. Dongarra, C. B. Moler, J. R. Bunch, and G. W. Stewart, *LINPACK users' guide*. SIAM, 1979.

[89] M. Safari, F. A. Davani, H. Afarideh, S. Jamili, and E. Bayat, "Discrete Fourier Transform Method for Discrimination of Digital Scintillation Pulses in Mixed Neutron-Gamma Fields," *IEEE Transactions on Nuclear Science,* vol. 63, no. 1, pp. 325-332, 2016.

[90] M. A. Moussa, M. Boucherma, and A. Khezzar, "A Detection Method for Induction Motor Bar Fault Using Sidelobes Leakage Phenomenon of the Sliding Discrete Fourier Transform," *IEEE Transactions on Power Electronics,* vol. 32, no. 7, pp. 5560-5572, 2017.

[91] C.-S. Park, "2D discrete Fourier transform on sliding windows," *IEEE Transactions on Image Processing,* vol. 24, no. 3, pp. 901-907, 2015.

[92] S. Chen, Z. Peng, Y. Yang, X. Dong, and W. Zhang, "Intrinsic chirp component decomposition by using Fourier Series representation," *Signal Processing,* vol. 137, pp. 319-327, 2017.

[93] M. J. Shensa, "The discrete wavelet transform: wedding the a trous and Mallat algorithms," *IEEE Transactions on signal processing,* vol. 40, no. 10, pp. 2464-2482, 1992.

[94] A. K. Jain, *Fundamentals of digital image processing*. Englewood Cliffs, NJ: Prentice Hall, 1989.

[95] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still image data compression standard*. Springer Science & Business Media, 1992.

[96] F. Korn, H. V. Jagadish, and C. Faloutsos, "Efficiently supporting ad hoc queries in large datasets of time sequences," in *Acm Sigmod Record*, 1997, vol. 26, no. 2, pp. 289-300: ACM.

[97] L. Chen and G. Chen, "Fuzzy modeling, prediction, and control of uncertain chaotic systems based on time series," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications,* vol. 47, no. 10, pp. 1527-1531, 2000.

[98] J. L. Aznarte and J. M. Benítez, "Equivalences between neural-autoregressive time series models and fuzzy systems," *IEEE transactions on neural networks,* vol. 21, no. 9, pp. 1434-1444, 2010.

[99] J. Casillas, O. Cordón, F. H. Triguero, and L. Magdalena, *Interpretability issues in fuzzy modeling*. Springer, 2013.

[100] S.-M. Chen, "Forecasting enrollments based on fuzzy time series," *Fuzzy sets and systems,* vol. 81, no. 3, pp. 311-319, 1996.

[101] D. T. Ho and J. M. Garibaldi, "Context-dependent fuzzy systems with application to time-series prediction," *IEEE Transactions on Fuzzy Systems,* vol. 22, no. 4, pp. 778-790, 2014.

[102] S. Alizadeh, A. Kalhor, H. Jamalabadi, B. N. Araabi, and M. N. Ahmadabadi, "Online Local Input Selection Through Evolving Heterogeneous Fuzzy Inference System," *IEEE Transactions on Fuzzy Systems,* vol. 24, no. 6, pp. 1364-1377, 2016.

[103] R. Bao, H. Rong, P. P. Angelov, B. Chen, and P.-k. Wong, "Correntropy-based evolving fuzzy neural system," *IEEE Transactions on Fuzzy Systems,* 2017.

[104] H. Rong, P. P. Angelov, X. Gu, and J. Bai, "Stability of Evolving Fuzzy Systems based on Data Clouds," *IEEE Transactions on Fuzzy Systems,* 2018.

[105] X. Xie, L. Lin, and S. Zhong, "Process Takagi–Sugeno model: A novel approach for handling continuous input and output functions and its application to time series prediction," *Knowledge-Based Systems,* vol. 63, pp. 46-58, 2014.

[106] Q. Gan and C. J. Harris, "A hybrid learning scheme combining EM and MASMOD algorithms for fuzzy local linearization modeling," *IEEE transactions on neural networks,* vol. 12, no. 1, pp. 43-53, 2001.

[107] W. Huang, S.-K. Oh, and W. Pedrycz, "Hybrid Fuzzy Wavelet Neural Networks Architecture Based on Polynomial Neural Networks and Fuzzy Set/Relation Inference-Based Wavelet Neurons," *IEEE transactions on neural networks and learning systems,* 2017.

[108] W. Huang, S.-K. Oh, and W. Pedrycz, "Fuzzy wavelet polynomial neural networks: analysis and design," *IEEE Transactions on Fuzzy Systems,* vol. 25, no. 5, pp. 1329-1341, 2017.

[109] C. Li and T.-W. Chiang, "Complex neurofuzzy ARIMA forecasting—a new approach using complex fuzzy sets," *IEEE Transactions on Fuzzy Systems,* vol. 21, no. 3, pp. 567-584, 2013.

[110] M. Kumar, R. Stoll, and N. Stoll, "Deterministic approach to robust adaptive learning of fuzzy models," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* vol. 36, no. 4, pp. 767-780, 2006.

[111] P. A. Mastorocostas and J. B. Theocharis, "A recurrent fuzzy-neural model for dynamic system identification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* vol. 32, no. 2, pp. 176-190, 2002.

[112] N. Baklouti, A. Abraham, and A. Alimi, "A Beta basis function Interval Type-2 Fuzzy Neural Network for time series applications," *Engineering Applications of Artificial Intelligence,* vol. 71, pp. 259-274, 2018.

[113] Z. Lv, J. Zhao, Y. Zhai, and W. Wang, "Non-iterative T–S fuzzy modeling with random hidden-layer structure for BFG pipeline pressure prediction," *Control Engineering Practice,* vol. 76, pp. 96-103, 2018.

[114] T. Dam and A. K. Deb, "A clustering algorithm based TS fuzzy model for tracking dynamical system data," *Journal of the Franklin Institute,* vol. 354, no. 13, pp. 5617-5645, 2017.

[115] M. M. Ebadzadeh and A. Salimi-Badr, "CFNN: Correlated fuzzy neural network," *Neurocomputing,* vol. 148, pp. 430-444, 2015.

[116] R. Dash and P. Dash, "Efficient stock price prediction using a self evolving recurrent neuro-fuzzy inference system optimized through a modified differential harmony search technique," *Expert Systems with Applications,* vol. 52, pp. 75-90, 2016.

[117] D. Graves and W. Pedrycz, "Fuzzy prediction architecture using recurrent neural networks," *Neurocomputing,* vol. 72, no. 7-9, pp. 1668-1678, 2009.

[118] H. Gu and H. Wang, "Fuzzy prediction of chaotic time series based on singular value decomposition," *Applied Mathematics and Computation,* vol. 185, no. 2, pp. 1171-1185, 2007.

[119] S. Hassan, M. A. Khanesar, J. Jaafar, and A. Khosravi, "Comparative analysis of three approaches of antecedent part generation for an IT2 TSK FLS," *Applied Soft Computing,* vol. 51, pp. 130-144, 2017.

[120] J. M. Mendel, *Uncertain rule-based fuzzy logic systems: introduction and new directions*. Prentice Hall PTR Upper Saddle River, 2001.

[121] A. Fiordaliso, "Autostructuration of fuzzy systems by rules sensitivity analysis," *Fuzzy Sets and Systems,* vol. 118, no. 2, pp. 281-296, 2001.

[122] S. K. Oh, W. Pedrycz, and B. J. Park, "Hybrid identification of fuzzy rule-based models," *International journal of intelligent systems,* vol. 17, no. 1, pp. 77-103, 2002.

[123] Z. Michalewicz and S. J. Hartley, "Genetic algorithms+ data structures= evolution programs," *Mathematical Intelligencer,* vol. 18, no. 3, p. 71, 1996.

[124] W. Pedrycz and F. Gomide, *Fuzzy systems engineering: toward human-centric computing*. John Wiley & Sons, 2007.

[125] M. Männle, "FTSM—Fast Takagi-Sugeno Fuzzy Modeling," *IFAC Proceedings Volumes,* vol. 33, no. 11, pp. 651-656, 2000.

[126] M. Mannle, "Parameter optimization for Takagi-Sugeno fuzzy models-lessons learnt," in *Systems, Man, and Cybernetics, 2001 IEEE International Conference on*, 2001, vol. 1, pp. 111-116: IEEE.

[127] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The RPROP algorithm," in *Neural Networks, 1993., IEEE International Conference on*, 1993, pp. 586-591: IEEE.

[128] R. Yusof, R. Z. A. Rahman, M. Khalid, and M. F. Ibrahim, "Optimization of fuzzy model using genetic algorithm for process control application," *Journal of the Franklin Institute,* vol. 348, no. 7, pp. 1717-1737, 2011.

[129] T. L. Seng, M. B. Khalid, and R. Yusof, "Tuning of a neuro-fuzzy controller by genetic algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* vol. 29, no. 2, pp. 226-236, 1999.

[130] O. Cord, *Genetic fuzzy systems: evolutionary tuning and learning of fuzzy knowledge bases*. World Scientific, 2001.

[131] A. Trabelsi, F. Lafont, M. Kamoun, and G. Enea, "Fuzzy identification of a greenhouse," *Applied Soft Computing,* vol. 7, no. 3, pp. 1092-1101, 2007.

[132] R. Babuška and H. B. Verbruggen, "An overview of fuzzy modeling for control," *Control Engineering Practice,* vol. 4, no. 11, pp. 1593-1606, 1996.

[133] W.-D. Kim, S.-K. Oh, K.-S. Seo, and W. Pedrycz, "Growing rule-based fuzzy model developed with the aid of fuzzy clustering," in *IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint*, 2013, pp. 573-578: IEEE.

[134] J. Yen, L. Wang, and C. W. Gillespie, "Improving the interpretability of TSK fuzzy models by combining global learning and local learning," *IEEE Transactions on fuzzy Systems,* vol. 6, no. 4, pp. 530-537, 1998.

[135] C. Li, J. Zhou, X. Xiang, Q. Li, and X. An, "T–S fuzzy model identification based on a novel fuzzy c-regression model clustering algorithm," *Engineering Applications of Artificial Intelligence,* vol. 22, no. 4-5, pp. 646-653, 2009.

[136] L.-X. Wang and J. M. Mendel, "Fuzzy basis functions, universal approximation, and orthogonal least-squares learning," *IEEE transactions on Neural Networks,* vol. 3, no. 5, pp. 807-814, 1992.

[137] C. Li, J. Zhou, B. Fu, P. Kou, and J. Xiao, "T–S fuzzy model identification with a gravitational search-based hyperplane clustering algorithm," *IEEE Transactions on Fuzzy Systems,* vol. 20, no. 2, pp. 305-317, 2012.

[138] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Information sciences,* vol. 179, no. 13, pp. 2232-2248, 2009.

[139] C. Li and J. Zhou, "Parameters identification of hydraulic turbine governing system using improved gravitational search algorithm," *Energy Conversion and Management,* vol. 52, no. 1, pp. 374-381, 2011.

[140] B.-J. Park, W. Pedrycz, and S.-K. Oh, "Identification of fuzzy models with the aid of evolutionary data granulation," *IEE Proceedings-Control Theory and Applications,* vol. 148, no. 5, pp. 406-418, 2001.

[141] M. Soltani, B. Aissaoui, A. Chaari, F. B. Hmida, and M. Gossa, "A modified fuzzy c-regression model clustering algorithm for TS fuzzy model identification," in

Systems, Signals and Devices (SSD), 2011 8th International Multi-Conference on, 2011, pp. 1-6: IEEE.

[142] X.-F. Wu, Z.-Q. Lang, and S. A. Billings, "An orthogonal least squares based approach to FIR designs," *International Journal of Automation and Computing,* vol. 2, no. 2, pp. 163-170, 2005.

[143] E. Lughofer, *Evolving fuzzy systems-methodologies, advanced concepts and applications*. Springer, 2011.

[144] R. R. Yager and D. P. Fileu, "Learning of fuzzy rules by mountain clustering," in *Applications of Fuzzy Logic Technology*, 1993, vol. 2061, pp. 246-255: International Society for Optics and Photonics.

[145] S. L. Chiu, "Fuzzy model identification based on cluster estimation," *Journal of Intelligent & fuzzy systems,* vol. 2, no. 3, pp. 267-278, 1994.

[146] J.-S. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE transactions on systems, man, and cybernetics,* vol. 23, no. 3, pp. 665-685, 1993.

[147] J.-S. R. Jang, "Fuzzy Modeling Using Generalized Neural Networks and Kalman Filter Algorithm," in *AAAI*, 1991, vol. 91, pp. 762-767.

[148] E. D. Lughofer, "FLEXFIS: A robust incremental learning approach for evolving Takagi–Sugeno fuzzy models," *IEEE Transactions on fuzzy systems,* vol. 16, no. 6, pp. 1393-1410, 2008.

[149] A. Gersho and R. M. Gray, *Vector quantization and signal compression*. Springer Science & Business Media, 2012.

[150] W. Li, H. H. Yue, S. Valle-Cervantes, and S. J. Qin, "Recursive PCA for adaptive process monitoring," *Journal of process control,* vol. 10, no. 5, pp. 471-486, 2000.

[151] P. P. Angelov and D. P. Filev, "An approach to online identification of Takagi-Sugeno fuzzy models," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* vol. 34, no. 1, pp. 484-498, 2004.

[152] M. Pratama, S. G. Anavatti, P. P. Angelov, and E. Lughofer, "PANFIS: A novel incremental learning machine," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 25, no. 1, pp. 55-68, 2014.

[153] B. Vigdor and B. Lerner, "The bayesian artmap," *IEEE Transactions on Neural Networks,* vol. 18, no. 6, pp. 1628-1644, 2007.

[154] E. Lughofer, J.-L. Bouchot, and A. Shaker, "On-line elimination of local redundancies in evolving fuzzy systems," *Evolving Systems,* vol. 2, no. 3, pp. 165-187, 2011.

[155] E. Lughofer and S. Kindermann, "SparseFIS: Data-driven learning of fuzzy systems with sparsity constraints," *IEEE Transactions on Fuzzy Systems,* vol. 18, no. 2, pp. 396-411, 2010.

[156] T. Bonesky, K. Bredies, D. A. Lorenz, and P. Maass, "A generalized conditional gradient method for nonlinear operator equations with sparsity constraints," *Inverse Problems,* vol. 23, no. 5, p. 2041, 2007.

[157] E. Lughofer, "Extensions of vector quantization for incremental clustering," *Pattern Recognition,* vol. 41, no. 3, pp. 995-1011, 2008.

[158] R. R. Yager and D. P. Filev, "Generation of fuzzy rules by mountain clustering," *Journal of Intelligent & Fuzzy Systems,* vol. 2, no. 3, pp. 209-219, 1994.

[159] S.-T. Li and Y.-C. Cheng, "A stochastic HMM-based forecasting model for fuzzy time series," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* vol. 40, no. 5, pp. 1255-1266, 2010.

[160] Y.-C. Cheng and S.-T. Li, "Fuzzy time series forecasting with a probabilistic smoothing hidden markov model," *IEEE Transactions on Fuzzy Systems,* vol. 20, no. 2, pp. 291-304, 2012.

[161] M. R. Hassan, K. Ramamohanarao, J. Kamruzzaman, M. Rahman, and M. M. Hossain, "A HMM-based adaptive fuzzy inference system for stock market forecasting," *Neurocomputing,* vol. 104, pp. 10-25, 2013.

[162] M. R. Hassan, "A combination of hidden Markov model and fuzzy model for stock market forecasting," *Neurocomputing,* vol. 72, no. 16-18, pp. 3439-3446, 2009.

[163] M. R. Hassan, B. Nath, M. Kirley, and J. Kamruzzaman, "A hybrid of multiobjective Evolutionary Algorithm and HMM-Fuzzy model for time series prediction," *Neurocomputing,* vol. 81, pp. 1-11, 2012.

[164] M. A. Mohamed and P. Gader, "Generalized hidden Markov models. I. Theoretical frameworks," *IEEE Transactions on fuzzy systems,* vol. 8, no. 1, pp. 67-81, 2000.

[165] M. A. Mohamed and P. Gader, "Generalized hidden markov models. ii. application to handwritten word recognition," *IEEE transactions on fuzzy systems,* vol. 8, no. 1, pp. 82-94, 2000.

[166] N. K. Verma and M. Hanmandlu, "Additive and nonadditive fuzzy hidden Markov models," *IEEE Transactions on Fuzzy Systems,* vol. 18, no. 1, pp. 40-56, 2010.

[167] H.-C. Wang, C.-S. Lee, and T.-H. Ho, "Combining subjective and objective QoS factors for personalized web service selection," *Expert Systems with Applications,* vol. 32, no. 2, pp. 571-584, 2007.

[168] G. D. Forney, "The Viterbi algorithm," *Proceedings of the IEEE,* vol. 61, no. 3, pp. 268-278, 1973.

[169] N. R. Pal and J. C. Bezdek, "On cluster validity for the Fuzzy C-means model," *IEEE Transactions on Fuzzy systems,* vol. 3, no. 3, pp. 370-379, 1995.

[170] J. Zhou, C. P. Chen, L. Chen, and H.-X. Li, "A collaborative fuzzy clustering algorithm in distributed network environments," *IEEE Transactions on Fuzzy Systems,* vol. 22, no. 6, pp. 1443-1456, 2014.

[171] H. Tahani and J. M. Keller, "Information fusion in computer vision using the fuzzy integral," *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 20, no. 3, pp. 733-741, 1990.

[172] J. Wu, F. Chen, C. Nie, and Q. Zhang, "Intuitionistic fuzzy-valued Choquet integral and its application in multicriteria decision making," *Information Sciences,* vol. 222, pp. 509-527, 2013.

[173] C.-M. Hwang, M.-S. Yang, W.-L. Hung, and M.-G. Lee, "A similarity measure of intuitionistic fuzzy sets based on the Sugeno integral with its application to pattern recognition," *Information Sciences,* vol. 189, pp. 93-109, 2012.

[174] J. J. Liou, Y.-C. Chuang, and G.-H. Tzeng, "A fuzzy integral-based model for supplier evaluation and improvement," *Information Sciences,* vol. 266, pp. 199-217, 2014.

[175] B.-S. Yoo and J.-H. Kim, "Fuzzy integral-based gaze control of a robotic head for human robot interaction," *IEEE Transactions on Cybernetics* vol. 45, no. 9, pp. 1769-1783, 2015.

[176] A. C. B. Abdallah, H. Frigui, and P. Gader, "Adaptive local fusion with fuzzy integrals," *IEEE Transactions on Fuzzy Systems,* vol. 20, no. 5, pp. 849-864, 2012.

[177] T. Murofushi and M. Sugeno, "An interpretation of fuzzy measures and the Choquet integral as an integral with respect to a fuzzy measure," *Fuzzy sets and Systems,* vol. 29, no. 2, pp. 201-227, 1989.

[178] T. Onisawa, M. Sugeno, Y. Nishiwaki, H. Kawai, and Y. Harima, "Fuzzy measure analysis of public attitude towards the use of nuclear energy," *Fuzzy sets and systems,* vol. 20, no. 3, pp. 259-289, 1986.

[179] Y. Chi and T. Zhang, "Study on optimum fusion algorithms of IKONOS high spatial resolution remote sensing image," in *In Proceedings of the International Conference on Multimedia Technology (ICMT)*, Hangzhou, China, 2011, pp. 761-764: IEEE.

[180] V. Roberge, M. Tarbouchi, and G. Labonté, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning," *IEEE Transactions on Industrial Informatics,* vol. 9, no. 1, pp. 132-141, 2013.

[181] W. Elsayed, Y. Hegazy, M. El-bages, and F. Bendary, "Improved Random Drift Particle Swarm Optimization with Self-Adaptive Mechanism for Solving the Power Economic Dispatch Problem," *IEEE Transactions on Industrial Informatics,* vol. 13, no. 3, pp. 1017 - 1026, 2017.

[182] H. Izakian and W. Pedrycz, "Anomaly detection in time series data using a fuzzy c-means clustering," in *Proceedings of the Joint IFSA World Congr. and NAFIPS Annual Meeting*, 2013, pp. 1513-1518: IEEE.

[183] R. Mark, P. Schluter, G. Moody, P. Devlin, and D. Chernoff, "An annotated ECG database for evaluating arrhythmia detectors," *IEEE Trans. Biomed. Eng.,* vol. 29, no. 8, pp. 600-600, 1982.

[184] G. B. Moody and R. G. Mark, "The MIT-BIH arrhythmia database on CD-ROM and software for use with it," in *Computers in Cardiology 1990, Proceedings.*, 1990, pp. 185-188: IEEE.

[185] C. C. Aggarwal, *Outlier analysis*. New York, NY, USA: Springer 2013.

[186] W. Pedrycz and J. V. de Oliveira, "A development of fuzzy encoding and decoding through fuzzy clustering," *IEEE Transactions on Instrumentation and Measurement,* vol. 57, no. 4, pp. 829-837, 2008.

[187] B. Jeon and D. A. Landgrebe, "Fast Parzen density estimation using clustering-based branch and bound," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 16, no. 9, pp. 950-954, 1994.

[188] Y. Chen *et al.*, "The ucr time series classification archive," *URL www. cs. ucr. edu/~ eamonn/time_series_data,* 2015.

[189] Y.-C. Lin, H.-H. Lai, and C.-H. Yeh, "Consumer-oriented product form design based on fuzzy logic: A case study of mobile phones," *International Journal of Industrial Ergonomics,* vol. 37, no. 6, pp. 531-543, 2007.

[190] P. Angelov and R. Buswell, "Identification of evolving fuzzy rule-based models," *IEEE Transactions on Fuzzy Systems,* vol. 10, no. 5, pp. 667-677, 2002.

[191] Y.-W. Chen, J.-B. Yang, C.-C. Pan, D.-L. Xu, and Z.-J. Zhou, "Identification of uncertain nonlinear systems: constructing belief rule-based models," *Knowledge-Based Systems,* vol. 73, pp. 124-133, 2015.