PATTERN DISCOVERY FROM MULTIVARIATE TIME SERIES DATA

ZHOU PEIYUAN

Ph.D

The Hong Kong Polytechnic University

2016

The Hong Kong Polytechnic University
Department of Computing

Pattern Discovery from Multivariate Time Series Data

Zhou Peiyuan

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

June 2016

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

___________________ (Signed)

Zhou Peiyuan (Name of Student)

Abstract

A multivariate time series (MTS) is made up of data collected by monitoring the values of a set of temporarily related or interrelated variables over a period of time at successive instants spaced at uniform time intervals. Therefore it consists of a set of component univariate time series (CUTS) which corresponds to the series of values taken by a variable over the monitoring period of time. Given a set of MTS, the problem of classification or clustering such data is concerned with discovering inherent groupings of the data according to how similar or dissimilar the time series are to each other.

Two main challenges to processing MTS are multiple variables and high dimensions. Although existing feature extraction methods can effectively reduce the dimensions, the methods may lose certain important correlations among the variables while reducing high dimensions of MTS. In view of the growing need to deal with MTS in many application domains, we propose a generic and application-independent method capable of discovering, classifying and clustering phases associated with pattern discovery from data.

Firstly, a new feature extraction method resulting in a feature vector is proposed to reduce high dimensions of MTS. Next, with a view to reducing the dimensions of the feature vector, we propose an unsupervised feature selection method capable of reducing the computation time, improve classification performance, and facilitate a better understanding of datasets. Finally, the classifier and the clustering methods are applied. Specifically, the proposed algorithms address the following issues.

i) A general algorithm is proposed to discover the inter-temporal and intra-temporal patterns associated with an MTS.

ii) In order to discover patterns from the MTS, discretization is needed to transform numerical data into level value. Hence, a fuzzy approach is proposed to discover fuzzy temporal patterns using fuzzy membership functions.

iii) After discovering temporal patterns, a feature vector is constructed by combining diverse measurements of intra-/inter-temporal patterns for each MTS.

iv) Classify and cluster MTS using feature vectors after selecting the appropriate number of features using an unsupervised attribute clustering algorithm.

In addition, since MTS data are commonly found in business and finance, social and biological sciences, engineering and computing, medicine and healthcare, etc., effective classification of such data has many potential applications in a wide range of problem domains. The performance of the proposed algorithm has been tested using both synthetic and real-world datasets. It is also applied in several real case studies, viz. classification for single-trial EEG and association analysis, clustering and portfolio management for stock markets.

Both experimental results and practical solutions have shown that the proposed algorithm can be a promising algorithm for MTS analysis.

Publications arising from the thesis

1. Liu, Y., Liu, Y., Wang, C., Wang, X., **Zhou, P.Y.**, Yu, G., Chan, K.C.C., What Strikes the Strings of Your Heart?-Multi-Label Dimensionality Reduction for Music Emotion Analysis via Brain Imaging. IEEE Transactions on Autonomous Mental Development, 2015.
2. **Zhou, P.Y**, Chan, K.C.C., An Unsupervised Attribute Clustering Algorithm for Unsupervised Feature Selection. Data Science and Advanced Analytics (DSAA), 2015 IEEE International Conference on. IEEE, 2015.
3. **Zhou, P.Y**, Chan, K.C.C., A Feature Extraction Method for Multivariate Time Series Classification Using Temporal Patterns. Processing of Conference on PAKDD2015, 2015.
4. **Zhou, P.Y.,** Chan, K.C.C., A Model-based Multivariate Time Series Clustering Algorithm, Processing of Conference on PAKDD2014, Workshops on Pattern Mining and Application of Big Data (BigPMA), 2014
5. **Zhou, P.Y**, Chan, K.C.C., Fuzzy Clustering of Multivariate Time Series in the Application of Stock Analysis. IEEE Transaction of Systems, Man and Cybernetics: Systems. April, 2016 (under review)
6. **Zhou, P.Y**, Chan, K.C.C., A Fuzzy Approach to Discover Patterns for Multi-Channel EEG Classification. IEEE Transaction on Fuzzy Systems. March. 2016 (under review)
7. **Zhou, P.Y**, Chan, K.C.C., Discovering Fuzzy Temporal Association in Multivariate Time Series for Stock Analysis. IEEE Transaction on Fuzzy Systems. Nov. 2015 (major revision)
8. **Zhou, P.Y**, Ou, X.J.C., Chan, K.C.C., Corporate Communication Network and the Stock Price: Insights from the Algorithm. IEEE Transaction of Systems, Man and Cybernetics: Systems. 2015(major revision)

Acknowledgements

Firstly of all, I would like to express my sincere appreciation to my PhD supervisor, *Prof. Keith C.C. Chan,* for his gentle and patient supervision. His generous encouragements and support give me confidence to overcome difficulties during my PhD studies. I am deeply grateful for his insightful comments, carful revision for my research paper and this thesis, which inspire me to devote myself fully to the research work. I would say that I sincerely consider myself very fortunate to learn from him to become a critical thinker and an innovative problem solver.

I would like to thank Prof. Carol X.J. Ou, Department of Information Management, Tilburg University; Prof. Fiona Y. Liu, Department of Computing, The Hong Kong Polytechnic University for giving me many valuable suggestions and revision on my research paper revision. I also would like to thank Dr. Annie E.S. Lee, Dr. Andrew K.C. Wong. I have greatly enjoyed working with them.

And also, I would like to thank the university to provide vibrant environment for my PhD study. I also would like to thank my colleagues and friends in the Department of Computing for their friendship and encouragement.

Finally, I would like to thank my parents, their love support me throughout the time of my student life. I also would like to thank my husband, Li Jieming and my son, Li Sixuan. Without their love and support, I would not have completed this thesis. So, thanks go to my family, to whom I piously dedicate this thesis.

Table of Contents

CERTIFICATE OF ORIGINALITY II

Abstract III

Publications arising from the thesis V

Acknowledgements VI

List of Tables X

List of Figures XIII

Chapter 1.Introduction 1

1.1 Background and Research Motivations 2

1.2 The Problem 5

1.3 The Solutions 7

1.4 Organization of Thesis 9

Chapter 2.Related Work 11

2.1 Feature Extraction for MTS 11

2.2 Classification for MTS 13

2.3 Feature Selection 16

2.3.1 Attribute Clustering 17

2.3.2 Maximal Information Coefficient 18

Chapter 3.Feature Extraction and Classification for Multivariate Time Series 20

3.1 Notations and Definitions 21

3.2 Discretization for MTS 22

3.3 Discovering Intra-/Inter-CTP 23

3.3.1 Discovering Intra-CUTS Temporal Patterns (CTP) 23

3.3.2 Discovering Inter-CUTS Temporal Patterns (CTP) 26

3.4 Classification for MTS 28

3.5 Experimental Result 29

3.5.1 Evaluation Methods 29

3.5.2 Performance Evaluation of MTSC Using Synthetic Datasets 31

3.5.3 Further Experiments Using Real Data 36

3.5.4 Comparison Results 39

Chapter 4.A Fuzzy Approach for Discovering Patterns in Multivariate Time Series... 43

4.1 Fuzzification 44

4.2 Discovering Intra-Fuzzy Temporal Patterns (Intra-FTP) 46

4.3 Discovering Inter-Fuzzy Temporal Patterns (Inter-FTP) 49

4.4 Prediction Based on the Discovered Intra- and Inter-FTP 51

4.5 Illustrating Association Analysis using Synthetic Data Set 52

4.6 Classification and Clustering 55

4.7 Performance Evaluation 56

Chapter 5.Unsupervised Attribute Clustering Algorithm for Feature Selection 60

5.1 Proposed Algorithm 60

5.1.1 Attribute Interdependence Measure 61

5.1.2 Attribute Clustering Algorithm 63

5.3 Experimental Result on Synthetic Data Set 65

5.4 Experimental Result on Real Datasets 67

5.4.1 Comparison Result Between With and Without UACA 69

5.4.2 Comparison Result for Different Feature Selection Algorithms 70

Chapter 6.Experimental Result in Real Cases 74

6.1 Pattern Analysis and Prediction in the Application of Stock Market 74

6.1.1 Previous Work about Association Analysis for Stock Market 76

6.1.2 Datasets Description 79

6.1.3 Comparing with Association Analysis Algorithms 80

6.1.4 Analysis for Stock Market 88

6.2 Stock Clustering 98

6.2.1 Clustering Analysis for Stocks 98

6.2.2 Comparison Result for Stock Clustering 100

6.2.3 Creating the Efficient Portfolio for Stock Management 104

6.3 Multi-Channel EEG Classification 109

6.3.1 Previous Work about Classification Multi-Channel EEG Signals 111

6.3.3 Datasets Description 115

6.3.4 Performance Evaluation 117

6.3.5 Experimental Result 120

Chapter 7.Conclusion 128

7.1 Summary 128

7.2 Future Work 131

Reference 133

List of Tables

Table 3.1 Confusion Matrix for Evaluation 30

Table 3.2 Comparing the Results from the Synthetic Dataset.................................. 33

Table 3.3 Experimental Result Using MTSC on Real World Datasets 39

Table 3.4 Comparison Result on Real World Datasets... 40

Table 4.1 Synthetic Dataset for Illustration... 53

Table 4.2 Fuzzification Result.. 53

Table 4.3 Results Concerning Significant Discrepancy Measure 54

Table 4.4 Significant Associations Discovered in the Synthetic Dataset.................. 57

Table 5.1 Descriptions of Datasets in the Experiment 68

Table 5.2 Feature Selection Result Using UACA.. 69

Table 5.3 Classification Result without Feature Selection....................................... 70

Table 5.4 Classification Result Using Selected Features by UACA......................... 70

Table 5.5 Bayes Classification Accuracy Using Different Feature Selection Methods 71

Table 5.6 KNN Classification Accuracy Using Different Feature Selection Methods 71

Table 5.7 C5.0 Classification Accuracy Using Different Feature Selection Methods 72

Table 6.1(a) Prediction Results for the NYSE Data Set.. 82

Table 6.1(b) Prediction Result for the HKSE Data Set 83

Table 6.2(a) Comparison Result for the NYSE Data Set 85

Table 6.2 (b) Comparison Result for the HKSE Data Set 86

Table 6.3 Prediction Result for Stocks from the NYSE Using Different Time Periods 90

Table 6.4 The Most Significant Factors Affecting Movement in the NYSE 92

Table 6.5 Prediction Results for Stocks from HKSE Using Different Time Periods. 93

Table 6.6 The Most Significant Factors Affecting price movements in HKSE 95

Table 6.7 Prediction Results for Stocks from the NYSE Using Different Time Intervals 97

Table 6.8 Prediction Result for Stocks from the HKSE Using Different Time Intervals 97

Table 6.9 Comparing the Results from the Proposed Algorithm with Those from Other Approaches for the NYSE Dataset 102

Table 6.10 Comparing the Results from the Proposed Algorithm with Those from Other Approaches for the HKSE Dataset 102

Table 6.11 Weights for Stocks Chosen for Portfolio in the NYSE Dataset 105

Table 6.12 Weights for Stocks chosen for the Portfolios for the HKSE Dataset 106

Table 6.13 Comparison Results for Total Returns from Different NYSE Portfolios 108

Table 6.14 Comparison Results for Total Returns of Different HKSE Portfolios ... 108

Table 6.15 Classification Results from the GDD Dataset Collected From Different Combinations of Channels Located at the Back Part of the Brain 120

Table 6.16 Experimental Result with the GDD Dataset Using Channels Combinations from the Front Brain 122

Table 6.17 Experiments with the ALC Dataset Using Different Combinations of Data Drawn from Different Channels 123

Table 6.18 Experiments with the ALC Dataset Using Other Channel Combination 123

Table 6.19 Experiments with the EPL Dataset Using the Fz and Pz Channels 124

Table 6.20 Comparison of Experimental Results from Different Algorithms 124

List of Figures

Figure 3.1 The Description of the ith MTS ($\boldsymbol{Ti}$) 21

Figure 3.2 The Pseudo Code for Pattern Discovery from MTS Data 27

Figure 3.3 Patterns embedded in synthetic data 32

Figure 3.4 Example of Missing or Unevenly Sampling Data 34

Figure 3.5 Results for Synthetic Data with Missing Values 35

Figure 3.6 Performance of the Different Algorithms on Real World Datasets 42

Figure 4.1 Membership functions for One CUTS 45

Figure 4.2 Degree of Membership 46

Figure 4.3 Pseudo Code for Process of Discovering Intra-/Inter-FTP 50

Figure 4.4 Patterns Embedded in Synthetic Dataset 57

Figure 4.5 Example of Missing or Unevenly Sampling Data 58

Figure 4.6 Comparison Results for Synthetic Data with Missing Values 59

Figure 5.1 The pseudo code of unsupervised attribute clustering algorithm 64

Figure 5.2 Attribute value of A1 and A2 in the tuples in the synthetic dataset 66

Figure 5.3 The value of SMIC over all the clusters found in the synthetic dataset ... 67

Figure 5.4 Average classification accuracy using different feature selection methods 73

Figure 6.1 Performances of Different Algorithms on NYSE and HKSE data 88

Figure 6.2 Different Time Intervals for Prediction 96

Figure 6.3 Comparison of Clustering Result for the Two Datasets........................104

Figure 6.4 Plots of Returns from Different NYSE Portfolios................................107

Figure 6.5 Plots of Returns from Different HKSE Portfolios................................107

Figure 6.6 Method to Collect EEG Data..110

Figure 6.7 Process of Raw-Data Based Classification...112

Figure 6.8 Process of Feature-Based Classification...113

Figure 6.9 (a) Notable Channels for GDD Data Set (Inoue, Takuya. Et al. 2001)...118

Figure 6.10 Combinations of Channels in the Front Area of Brain for Gender EEG Data...121

Figure 6.11 Other Channels Combinations for ALC Data Set................................123

Figure 6.12 Comparison of Average Classification Accuracy from Different Classification Methods...126

Chapter 1.

Introduction

Time-series data consist of a sequence of values obtained over a period of time at successive instants spaced at uniform time intervals (Lee, S. et al. 2004; Dunham, M.H. 2002). In a *univariate time series* (UTS) the values are made up of single real numbers corresponding to different time points while a *multivariate time series* (MTS) consists of numerous observations corresponding to each time point (Yang, K. et al, 2004). Any MTS therefore consists of a set of component univariate time series (CUTS) each of which corresponds to the series of values taken by a variable over the given period of monitoring time.

Time series classification has already received much attention, so a number of different approaches have been developed to tackle the problem. Hence, in this thesis, I focus on introducing the proposed algorithm for processing multivariate time series and applying it to analyze a range of real-world cases.

In this chapter, the background and research motivations are introduced first. Next, in section 1.2, the major challenges involved in processing MTS are summarized. The problem and the solutions are summarized in Sections 1.3 and 1.4, respectively. Finally, the organization of the rest of the thesis is presented in Section 1.5.

1.1 Background and Research Motivations

A multivariate time series (MTS) is made up of a collection of data values resulting from monitoring, over a period of time, a set of temporally interrelated variables at successive time instants spaced at uniform time intervals (Yang, K. et al, 2004). While considerable methods have been developed for discovering patterns in an individual time series, relatively little work on discovering patterns in multivariate time series has been reported (Zhuang, D.E.H., 2014). Temporal pattern discovery is a subtask of time series mining (Laxman, S. et al. 2006; Fu, T. 2011) and the objective of it is to discover interesting patterns hidden in the time series. And then by using the discovered patterns, we can do further analysis, such as classification or clustering, for MTS.

MTS classification is a supervised learning problem aimed at labelling a set of MTS (Esmael, B. et al. 2012), which are commonplace in business and finance, social and biological sciences, engineering and computing, medicine and healthcare, and so on. Effective classification of such data can have many applications in many problem domains (Liao, T.W. 2005; Deshpande, M. et al. 2002; Wei, L. et al. 2006; Duskin, O. 2009; Batal, I. et al. 2009). For example, in genomic research, Deshpande et al. (Deshpande, M. et al. 2002) classified protein sequences into existing categories to learn the functions of a new protein. In health-informatics, Wei et al. (Wei, L. et al. 2006) classified ECG time series data to determine whether the data had been collected from a healthy person or a patient with heart disease. In medicine and healthcare, the data collected from the monitoring of many variables that represent different signs and symptoms constitute an MTS. Classifying of such MTS for a group of patients may allow people suffering from similar diseases to be identified and grouped for appropriate prognosis, diagnosis and medical treatment. There have also been some interesting applications using MTS classification. For example, Duskin et al. (Duskin, O.et al. 2009) studied the activity patterns along multiple dimensions and built a classifier capable of distinguishing web-robots from human users.

In general, feature extraction is performed for MTS before classifying them. The traditional classification algorithms, such as Support Vector Machines (SVM) and Artificial Neural Networks (ANN), are used to classify the feature vectors (Larsen, B. et al. 1999; Ding, S. et al. 2012; Zhu, D. et al. 1999) after feature extraction.

The classification approaches developed so far can be categorized into two types depending on whether raw data are used directly or indirectly in the classification process. For those using raw data directly, existing algorithms such as the Support Vector Machine (SVM), K-Nearest Neighbors (K-NN), typically use distance function to measure the similarity between a pair of times series. This determines the quality of the classification significantly, and is called the *Raw-data based* approach. Such approaches work best when the data involved are relatively low in noise content and are collected at relatively low sampling rates (Xing, Z. et al. 2010).

Time series classification algorithms that do not use raw-data directly can be classified into two types depending on whether they take a *feature-based* or *model-based approach* (Zhu, D. et al. 1999; Liao, T.W. 2005). *Feature-based* classification performs their tasks by first extracting features from the raw data to form feature vectors and these feature vectors, which are usually of lower dimensions than the original data, are used instead, for classification. Hence, feature extraction plays an important role in this kind of methods and the effectiveness of such feature-based approaches depends very much on how well the features can be identified (Zhu, D. et al. 1999; Liao, T.W. 2005). The more a problem domain is known, the better one can identify relevant features and the more accurately the classification task can be executed (Wang, X.Z. et al. 2006). If the features are not known well enough, one can consider model-based approaches to time series classification.

Algorithms adopting a *model-based* approach assume that each time series can be approximated by a known mathematical model definable by a set of parameters and that these parameters can be estimated relatively accurately (Liao, T.W. 2005). They also assume that time series that are similar to each other can be approximated by similar models. Under these assumptions, a pair of time series that are similar to each other can be approximated by correspondingly similar models characterized by similar model parameters. As in the case of feature-based approaches, the accuracies of model-based approaches depend on how much we know about the models of the time series.

In addition, the dimensionality of the dataset becomes larger and larger due to a tremendous growth in the volume of dataset available on the internet, digital libraries and news sources (Zhao X. et al. 2013). In order to increase the performance while mining large datasets, it is important to select a subset of original features, both in terms of dimension and size, to mine knowledge from large datasets (Fayyad, U. et al. 1996). Although the dimensions have been reduced effectively as compared with original MTS, a feature vector extracted from original multivariate time series can still contain many features. Hence, feature selection is important in the data mining field, which can select a subset of representative features from the input and can efficiently describe the input data while reducing effects from noise or irrelevant variables while still providing good prediction results (Mitra, P. et al. 2002; Guyon, I. et al. 2003). In order to improve classification performance, feature selection can be applied to continue optimizing the classification result. Feature selection has been applied in several domains, such as plant disease image recognition (Au, W.H. et al. 2005), gene expression understanding (Au, W.H. et al. 2005), and financial statement fraud detection (Ravisankar, P.et al. 2011).

1.2 The Problem

Although time series classification may have received a great deal of attention in the past, it also brings some new challenges to the data mining and machine learning community (Spiegel, S. et al. 2010). For example, a variety of approaches have been proposed for univariate time series classification (Zhang, H. et al. 2005; Rodrgueza, J.J. et al. 2005; Hayashi, A. et al. 2005). However, few papers about *multivariate time series* classification are found in the literature (Weng, X. et al. 2008).

The challenges associated with the classification process can be summarized as follows (Xing, Z. et al. 2010; Li, C. et al. 2007).

i. The algorithm should be able to handle high dimensionality of MTS (Esmael, B. et al. 2012).
ii. Traditional classifiers can only take input feature vector as input data, but no explicit features can be discovered from time series data.
iii. Subsequent to feature extraction, the dimensionality of feature space can be high and therefore the computation be costly.
iv. Important correlation information among variables may be lost if the value of one variable is broken into an MTS or each is processed separately.
v. It is difficult to extract features from an MTS with a different length using traditional feature extraction methods.

It is apparent that, for a time series classification algorithm to be useful for more complex tasks, it has to be able to tackle unequal length MTS. It is also required to be application-independent and be able to perform its tasks without any domain knowledge about relevant features or any assumption about underlying data models.

In addition, as Fulcher et al. (Fulcher, B.D. et al, 2014) have proposed, there are two main challenges associated with time-series classification: I) selecting an appropriate representation of the time series, and II) selecting a suitable measure for the dissimilarity or distance between time series.

In view of the research motivations and challenges mentioned above and to meet such requirements as much as possible, a multivariate time series classification algorithm needs to be developed to select I) an appropriate representation of the time series, and II) a suitable measure of dissimilarity of distance between time series.

Besides, there is a need to apply feature selection in order to improve the performance of classification. The main aim of feature selection is to select the appropriate number of attributes to avoid information loss and decrease time and space complexity (Zhao X. et al. 2013). When all the features are used for classification or clustering, the entire information contained in the data are retained, thus guaranteeing the accuracy of classification or clustering. However, within the feature space of high dimensional data, there can exist a large number of redundant features and noise characteristics. The features can greatly increase space and time complexity in the learning and training phases and therefore reduce the accuracy of classification (Zhao X. et al. 2013). Hence, in order to overcome problems arising from high dimensional data, it becomes necessary to use a feature selection algorithm capable of finding a feature subspace with good separability that can reduce the high dimensions of data and reduce time and space complexity for analysis (Han, J. et al. 2007).

1.3 The Solutions

In order to satisfy the requirements mentioned above, the proposed solves the problem as follows:

i) because of the multiple variables and high dimensions, the original MTS needs to be transformed into a feature vector using feature extraction methods or generative models;

ii) the proposed algorithm should retain information concerning correlations among variables;

iii) the proposed algorithm can handle processing MTS of different lengths;

iv) the application-independent algorithm should be able to classify MTS data that can have wide applications in many problem domains; and

v) the algorithm has to be able to carry out its tasks without having to make any assumptions about data and data models and it has to work well even in a relatively noisy environment where data might be missing or inconsistent.

Hence, we propose a new feature extraction algorithm combined with traditional classifier that provides a general classification strategy, called as Multivariate Time Series Classifier (MTSC).

Given a set of MTS, each consisting of a set of CUTS, MTSC classifies the MTS in question by discovering the temporal patterns implicit within each MTS. The temporal patterns that can be discovered in each MTS consist of both inter- and intra- temporal patterns. To discover and make use of these patterns for clustering, MTSC executes the following steps: (i) preprocessing data, (ii) discovering intra-temporal patterns in each CUTS within each MTS, (iii) discovering inter-temporal patterns between CUTS within each MTS, (iv) classifying the MTS based on the similarities and dissimilarities of the temporal patterns discovered via steps (ii) and (iii).

To test the effectiveness of MTSC, a number of experiments have been conducted using both simulated and real datasets. The real datasets used included selected instances of EEG data, ECG data, Physical Action data and Wafer data. For performance evaluation, the classification results were compared against known results using the *Classification Accuracy, Recall* and *Precision* measure. The results have shown that the MTSC algorithm is quite an effective algorithm in classifying MTS.

In order to reduce the size of feature vector to improve performance during classification, we propose yet another unsupervised feature selection method, called as Unsupervised Attribute Clustering Algorithm (UACA), for use in high-dimension datasets. A state-of-the-art similarity index, called MIC (Maximal Information Coefficient), is used to evaluate correlations between each pair of attributes. Next, the clustering algorithm (optimal K-modes) is used for attribute clustering. The UACA can handle the feature selection problem when the information of class label is unknown. The representation attribute can be selected automatically for feature selection.

In summary, the contributions can be summarized as follows: three methodologies, MTSC, FMTSC and UACA are proposed. MTSC can discover intra-/inter-temporal patterns and classify MTS even when the data covered are unequal and contain missing or erroneous values. The classification accuracy is improved when fuzzy measures are used. And finally, an unsupervised feature selection method can reduce the size of feature space without considering class label information and setting the number of cluster manually.

From an applicational point of view, we applied the FMTSC into three applications including predicting stock price, improving the industrial categories of stocks and create portfolios for helping investors, as well as classifying the multi-channel unequal length EEG signals.

1.4 Organization of Thesis

The rest of this thesis is organized as follows.

In chapter 2, we present a summary of existing work on time series classification and discuss the different kinds of problems that they can be used to tackle. A literature survey throwing light on feature selection is introduced in the same chapter. The advantages and drawbacks of major past works are described.

In Chapter 3, we describe the details of each step that MTSC takes to tackle MTS feature extraction and classification. In addition, the proposed algorithm is evaluated using both simulated and real datasets along with details related to how the performance of MTSC is tested.

Because the temporal patterns discovered are not expected to be described and represented precisely, in chapter 4, we extend the MTSC using a fuzzy approach, called as Fuzzy Multivariate Time Series Classification (FMTSC). FMTSC makes use of probabilistic and information theoretic measures in discovering temporal patterns. Subsequently, in order to improve the performance of classification algorithm, we discuss the proposed feature selection methods in chapter 5. For evaluating the performance of the proposed UACA algorithm, classification problems with different classifiers are compared with other methods in the same section. In chapter 6, we apply the proposed FMTSC into several real-world datasets for real cases which include pattern discovery for stock datasets, clustering and creating portfolios for stocks, and an analysis of single-trial EEG data. The experimental results demonstrate that the proposed algorithm is effective not only with respect to the algorithm but also in terms of practical applicability. Finally, in chapter 7, we discuss whether or not, and to what extent, the proposed method can be used for multivariate time series analysis. The chapter also evaluates the limitations of this study and suggests directions for future research.

Chapter 2.

Related Work

A multivariate time series (MTS) is a sequence of vectors with or without class label (Xing, Z. et al. 2010). For example, the signals from Electrocardiography (ECG) recorded from several sensors to describe the electrical activity of the heart can be represented by an MTS. The ECG data may come from either a healthy person or a patient, i.e., it can be labeled as "healthy" or "ill". MTS classification refers to the problem of classifying a set of MTS samples into a pre-defined set of classes (Xing, Z. et al. 2010). Available algorithms of MTS classification can be summarized in two steps: extracting the main features present using a feature extraction method (i.e. PCA (Xing, Z. et al. 2010)) and classifying the feature vectors using a classifier (i.e. SVM and ANN (Cortes, C. et al. 1995; Haykin, S. 1998)).

A review of literature on major feature extraction methods is given in Section 2.1 whereas the classification methods are described in Section 2.2. In order to facilitate the selection of significant features from feature vector, a literature survey of feature selection methods is presented in Section 2.3.

2.1 Feature Extraction for MTS

The design and performance of a classification method is affected greatly by feature extraction. We can define feature extraction as a process that uses the existing feature parameters to compose a lower-dimensional feature space, map the useful information contained in the original features into a smaller number of features while ignoring redundant and irrelevant information (Ding, S. et al. 2012). Traditional feature extraction methods are based on statistical analysis (Ding, S. et al. 2012), such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) and Factor Analysis.

In addition, several new feature extraction methods have been proposed for improving MTS classification performance. Li et al. (Li, C. et al. 2007) proposed a feature extraction method, called Singular Value Decomposition (SVD), to reduce data with different lengths and feature vectors, and then apply SVM on the feature vectors to classify the MTS. Weng et al. (Weng, X. et al. 2008) projected the original MTS on to PCA subspace by throwing away the smallest principal components, so that MTS samples are projected into a lower-dimensional space by using supervised Locality Preserving Projection (LPP). However, the above existing extraction methods may lead to a loss of some dependency relationship information.

Hence, to extract the features among the same and different variables while retaining more significant information for further classification and prove the efficiency of the proposed algorithm, we compare the proposed algorithm with traditional feature extraction methods, e.g., PCA. We focus on discovering the relationship within the same variable (intra-temporal patterns) and between different variables (inter-temporal patterns) at different time points, and combine the measurement values of all patterns into a feature vector.

2.2 Classification for MTS

Traditional classification models include approaches using decision trees, Bayesian networks, nearest neighbor classifiers and support vector machines, which have been proposed and applied for analyzing a variety of datasets. However, the success of classification methods depends heavily on the quality of data and data features used in the models (Batal, I. et al. 2009). This means that existing algorithms for time series classification can be classified into two types: those that take a *direct approach* and those that take an *indirect* approach. Algorithms taking the direct approach work directly with raw time series data in time or frequency domain and they are sometimes referred to as the *raw-data-based approaches* (Batal, I. et al. 2009; Liao, T.W. 2005). This method defines a distance function to measure the similarity between a given pair of time series. Once such a distance function is obtained, one can use some existing classification method, such as K-NN with a Euclidean distance measure and SVM with a local alignment kernel for time series classification (Batal, I. et al. 2009; Saiggo, H. et al, 2004). The similarity or distance measures are among the most appropriate when data are relatively low in noise content and when they are collected using relatively low sampling rates (Liao, T.W. 2005).

In case the data are noisy and exhibit high dimensionality, an indirect approach to time series classification is sometimes used. In such approaches, the raw time series data are usually converted first into a feature vector exhibiting a lower dimensionality or into a vector of estimated model parameters before an existing classification algorithm is executed on the converted data. In other words, the indirect approaches can be classified into *feature-based approaches* (a.k.a. *representation-based approaches*, Rani, S. et al. 2012) and *model-based approaches*, respectively (Batal, I. et al. 2009; Liao, T.W. 2005).

Feature-based approaches (Liao, T.W. 2005) perform their tasks with features extracted from the raw data rather than directly with the raw data. Generally, while classifying MTS data, feature extraction needs to be performed from the original data. Once the features have been identified, a conventional classifiers, such as SVM (She, R. et al. 2003), rule based classifier (Aggarwal, C.C. et al, 2002) and neural networks (Blekas, K. et al, 2005) can be applied to classify the time series (Xing, Z. et al. 2010; Ding, S. et al. 2012). Wu.et al. (Wu, D. et al. 2009) propose an effective classification model, C3M, for motion time series classification consisting of segmentation (preprocessing), dimension ranking and selection (feature extraction), and classification (classifier). Weng, X. et al. (2008) project the original MTS on to a Principal Component Analysis (PCA) subspace, which is a lower-dimensional space, by using supervised Locality Preserving Projection (LPP). A 1-NN classifier with Euclidean distance is then used to enable classification. Li et al. (Li, C. et al. 2007) propose a feature extraction method, called Singular Value Decomposition (SVD), to reduce the different length of data to feature vectors, and then apply SVM on the feature vectors to classify the MTS data. Ye et al. (Ye, J. et al. 2004) propose generalized principal component analysis (GPCA) to enable dimension reduction for images, and then K-NN is used for classification. Similarly, Guo et al. (Guo, C. et al. 2008) apply ICA, the expansion of principal component analysis and factor analysis, to reduce the dimensions of stock data to obtain feature vectors, and then the k-means algorithm is used to cluster the feature vectors found. Spiegel et al. (Spiegel, S. et al. 2011) summarize the several well-known feature extraction methods, such as Fourier

Transform (FT) (Kwon O.-W. et al. 2004), Discrete Wavelet Transform (DWT) (Kwon O.-W. et al. 2004) and Principal Component Analysis (PCA) (Huan, L. et al. 1998; Spiegel, S. et al. 2009; Spiegel, S. et al. 2011), and classification methods, such as K-NN (1-NN for K=1) (Cleary, J.G. et al, 1995), Bayesian networks (John, G.H. et al. 1995), and SVM (Keershi, S.S., et al. 2001) for multivariate time series. In summary, the feature vectors obtained are typically of lower dimensionality than the raw data vector and, given such vectors, a conventional classification algorithm is used (Sudjianto, A. et al. 1996).

Model Based approaches are time series classification methods based on generative models, which assume that the time series in a class have been generated by an underlying model (Xing, Z. et al. 2010). The simplest generative model, the Naive Bayes sequence classifier (Lewis, D.D., et al, 1998), assumes that the features in the sequences are independent of each other. However, the independence assumption is often violated in practice, hence, certain Markov Models and Hidden Markov Models (Wang, L. et al, 2002) are used based on assumption dependences among the elements in the time series (Lewis, D.D., et al, 1998). For example, Yakhnenko et al. (Yakhnenko, O. et al, 2005) apply a k-order Markov model to classify protein and text sequence data. Likewise, Srivastava et al. (Srivastava, P.K. et al, 2007) apply a profile HMM to classify biological sequences. After transforming the original time series into a model using the above methods, the resulting unknown sequence is aligned with the profile HMM for each class through dynamic programming and is classified into the class with the highest alignment score (Xing, Z. et al. 2010). Besides, an autoregressive (AR) model is used to deal with univariate ARIMA time series (Kalpakis, K. et al, 2001). Certain classification methods using the Linear Predictive Coding (LPC) are then applied. Lal et al. (Lal, T.N. et al, 2004) propose a method in which an autoregressive (AR) model of order 3 is encoded into each row of 39 channels EEG data, resulting in a 117 dimensional vector.

The MTSC first extracts features from MTS and then the classifier (i.e. SVM, ANN) (Cortes, C. et al. 1995) is applied in feature vector for classification.

Support Vector Machines use nonlinear mapping to transform original input data into a higher dimensional space and then search a linear separating hyper-plane to enable classification (Han, J. et al. 2007).

An Artificial Neural Network (ANN) is composed of interconnecting artificial neurons to compute values from inputs. Multi-Layer Perceptron (MLP) and Radial Basis Function (RBF) are two of the most widely used neural network architectures for classification or regression problems. RBF is a local type of learning algorithm which is responsive only to a limited section of input space, whereas MLP is a distributed approach (Cohen, S. et al. 2003).

2.3 Feature Selection

For further reducing the dimensions of feature vector, a feature selection method can be applied. Traditional feature selection methods can be classified into three types: the *Filter method*, the *Wrapper method,* and the *Embedded method* (Chandrashekar G. et al. 2014). All the three methods are supervised methods, so they need class label information. However, class labels are unknown in many data mining applications, so using an unsupervised feature selection method becomes important.

2.3.1 Attribute Clustering

When machine learning was first introduced, researchers were targeting a relatively small set of attributes. As the size of data and the diversity of attributes increased, data clustering began to break down; the classification results had not been seriously affected, yet their effectiveness was diminishing. The problems associated with supervised learning were partly solved through feature selection. Later, as data mining and pattern discovery came into play, the dimensionality problems became a little more relaxed. However, many other problems remained. Even up to today, most of the conventional clustering algorithms often face the challenges related to the nature of large scale datasets with large numbers of attributes. This problem is commonly referred to as the curse of dimensionality (Bellman, R. E. 1961).

In unsupervised learning, 'attribute clustering' was proposed to provide a partial solution to the problems. However, in general, class-dependent discretization had to be used to convert the continuous data into interval data (Au, W.H. et al. 2005). When discretization is used, more problems need to be considered, such as determining the number of intervals and the boundaries of each interval. In addition, different discretization methods may lead to varying degrees of information loss.

To cluster or select attributes, the t-value method has been widely used (Agrawal, R. et al. 1992). It is important to note that the t-value can only be used when the samples have been pre-classified. If no class information is available, the method cannot be used for attribute selection. Therefore the Attribute Clustering Algorithm, ACA (Agrawal, R. et al. 1992) was proposed to cluster attributes. In ACA, continuous data have to be converted into interval data before attribute clustering can be applied. To close this gap, in this thesis, we extend ACA to arrive at an algorithm that is able to deal with continuous data without using discretizing class information.

2.3.2 Maximal Information Coefficient

A relationship coefficient between features is usually used for measuring attribute similarity. For example, Combarro et al. (Combarro, E.F. et al. 2005) propose using the Pearson coefficient, which is one of the most popular relationship metrics for choosing relevant features by using linear measures in a text categorization application. However, Pearson cannot handle the situation when the dependence work is functional sine or cubic (Zhao X. et al. 2013). Au et al. (Au, W.H. et al. 2005) used Interdependence Redundancy Measure between two attributes to evaluate the dependency between two attributes. The measure can be obtained by dividing mutual information by entropy. This measure can only handle discrete datasets.

Recently, the Maximal Information Coefficient was introduced as a measure of dependence for describing two-attribute relationships (Reshef, D. N., 2011), which can be used as metrics in the exploration of large datasets. It can also be used for the detection of close associations between tens of thousands of attribute pairs in large datasets. MIC is based on the idea that if a relationship exists between two attributes, then a grid can be drawn on the scatterplot of the two attributes that partitions the data to encapsulate that relationship (Reshef, D. N., 2011). MIC uncovers features that not only have functional associations but also are statistically independent. In addition, MIC has been tested using many real biology-datasets to demonstrate its superiority over to a wide range of other methods, including Pearson/Spearman correlations and Mutual Information. Reshef et al. have addressed some criticisms of MIC (Reshef, D.

N., 2011). They have shown that MIC is an equitable statistic useful for analyzing high-dimensional datasets.

In order to arrive at the value of MIC linking two attributes, we can assume a data-point set, D', consisting of these two attributes, one as x-values divided into x bins and the other as y-values into y bins. This type of partition is called x-by-y grid G. The D' |G can represent the distribution of D' divided by one of x-by-y grid as G. $I^*(D', x, y)$ denotes the maximum mutual information of D'|G with the x-by-y grid. Next, a matrix called the characteristic matrix is constructed. The MIC value of two-attributes in dataset D' with size n and a grid size smaller than *B(n)* is defined as the maximum value of $M(D')_{x,y}$. The function *B(n)* limits the sizes of the grids when searching over feasible partitions.

Chapter 3.

Feature Extraction and Classification for Multivariate Time Series

With the above requirements in mind, we have developed an algorithm called MTSC (the Multivariate Time Series Classification) for classifying MTS. Since each MTS is made up of data obtained by monitoring a set of temporally related or interrelated variables, it can be expected that the values these variables take on are temporally related or interrelated to the previous values of the other variables. Therefore, the main task of MTSC is to uncover these temporal relationships or interrelationships. Since each of the variables being monitored generates a CUTS (component univariate time series) within an MTS, the main task of MTSC is to uncover the temporal relationships or interrelationships hidden between values observed at different time instants within and between different CUTS. These temporal relationships or interrelationships constitute what we call intra-CUTS temporal patterns (intra-CTP) and inter-CUTS temporal patterns (inter-CTP) respectively. MTSC performs its tasks in several steps: (i) discretize numerals data into discrete data, (ii) discover intra-CTP within each MTS, (iii) discover inter-CTP within each MTS, (iv) combine the measurements of intra-CTP and inter-CTP into a feature vector, and (iv) classify the MTS based on the feature vectors. In this section, we specify the process of the proposed algorithm. Section 3.1 describes the notations and definition of the MTS that are used in the proposed algorithm. Section 3.2 specifies the discretization for preprocessing MTS. And then

Section 3.3 and Section 3.4 describe the pattern discovery process and classification process for MTS. Finally, section 3.5 provides the experimental results obtained from both synthetic and real datasets.

3.1 Notations and Definitions

Definition 3-1 (MTS Data): A set of MTS data, **T**, collected over time, has the following characteristics:

i. **T** consists of *m* MTS which is represented as $\mathbf{T} = \{\boldsymbol{T^i} | \boldsymbol{i} = \mathbf{1}, \mathbf{2}, \dots \boldsymbol{m}\}$.

ii. Each $\mathbf{T^i}$=$\{T_j^i | j = 1, \dots n\}$, contains *n* CUTS.

iii. Each S_j^i represents a time series of data values collected for the *j*th CUTS in the *i*th MTS over a period of time and can be represented as T_j^i = ($v_{j,1}^i$, $v_{j,2}^i, \dots, v_{j,t-\tau}^i, \dots, v_{j,t}^i, \dots, v_{j,p}^i$), where $v_{j,t}^i \in [L_{V_j}, U_{V_j}]$, L_{V_j} represents the lower bound and U_{V_j}represents the upper bound of the values, 1≤τ≤$t-1$, $\tau \in \mathbb{Z}^+$ and *t =1,2,…p, p* is the total number of time points.

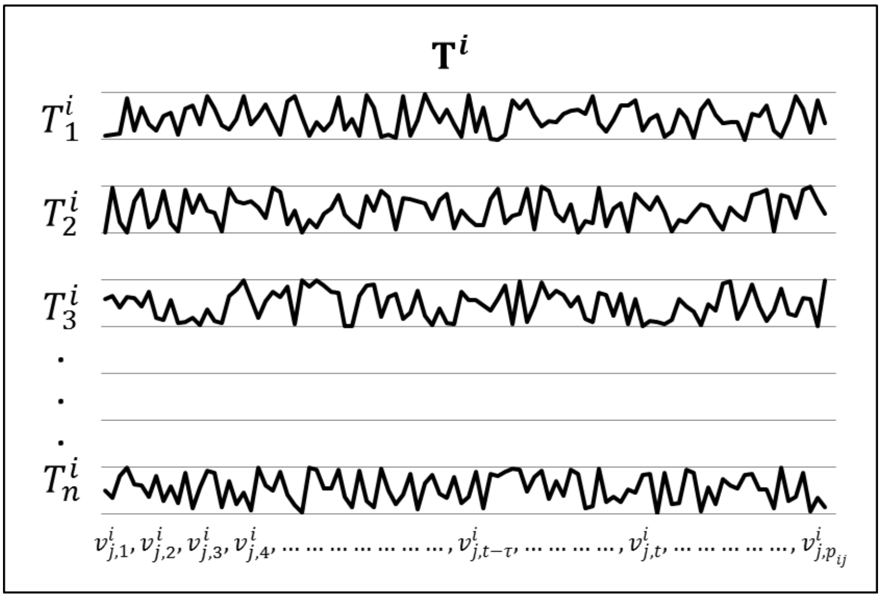

Figure 3.1 The Description of the ith MTS ($\boldsymbol{T^i}$)

One piece of original stock data $\mathbf{T^i}$ can be represented graphically as shown in Figure. 3.1. Given a set of MTS data, **T,** with characteristics as described above, the problem of classification such data involves categorizing $\boldsymbol{T^i}$**,** *i = 1,2,…m* data into *k* predefined classes, denoted as $C_1, C_2, \dots C_k$.

3.2 Discretization for MTS

To discover temporal patterns, the continuous values in the stock data are first discretized into several intervals. Discretization can minimize the impact of noisy data during the data mining process (Ma, P.C.H. et al. 2011). It can also help smooth data to reduce noise (Ma, P.C.H. et al. 2011), speed up the classification process (Ramoni, M. et al, 2000) and make classification results more meaningful and easier-to-understand (Liao, T.W. 2005). Hence, MTSC begins its work by partitioning the domain of the continuous data into a relatively small number of intervals and assign a nominal value to each. Data discretization is a frequently used technique to partition the value space of a continuous attribute into a finite number of intervals and assigning a nominal value to each of them (Liao, T.W. 2005; Dimitrova, E.S. et al. 2010). *Equal Width* and *Equal Frequency* are two of the simplest discretization methods. However, if uncharacteristic extreme values (outliers) exist in the dataset, *Equal Width* can hardly handle this situation. Hence, in our case, we transform the original numerical, $s_{j,t}^{i}$, which represents the *i*th MTS of *j*th attribute at *t* time point into $\mathrm{D}_{j,t}^{i}$ using *Equal Frequency* (Wong, A.K.C. et al. 1979) algorithm. Specifically, the domains of each of the *n* different variables are as follows: $V_j, j = 1, \dots, n$, domain(V_j) = $[L_{V_j}, U_{V_j}]$, $j = 1, \dots, n$, is to be partitioned into a relatively small finite number of representative intervals, as discretized domain values, (V_j) =

$[I_{V_j}^1, I_{V_j}^2, \dots, I_{V_j}^{n_j}]$, $j = 1, \dots, n,$ respectively so that $\bigcup_{k=1}^{n_j} I_{V_j}^k = [L_{V_j}, U_{V_j}]$, and $I_{V_j}^k \cap I_{V_j}^{k'} = \emptyset$ if $k \neq k'$, $j = 1, \dots, n,$ respectively. In this case, we set the number of bins as three, i.e., the original numerical data is transformed into three levels, {high, medium, low}.

3.3 Discovering Intra-/Inter-CTP

Given an MTS, the values that a particular variable takes on are expected to be related or interrelated with the previous values of the same variable or with other variables. These relationships and interrelationships are considered to be constituting, the intra-CTP and inter-CTP in the MTS, respectively. The measurements of such patterns can be combined into a feature vector to represent original MTS for classification. Hence, one main task of MTSC is, therefore, to uncover the intra-CTP and inter-CTP in each MTS.

3.3.1 Discovering Intra-CUTS Temporal Patterns (CTP)

Given a value, say, $v_{j,t}^i$, in T_j^i within $\boldsymbol{T^i}$ that is observed at time, *t*, in order for MTSC to discover how much it is temporally related to another value, say, $v_{j,t-\tau}^i$ previously observed at time, $t - \tau$, $1 \leq \tau < t$, the values are discretized into nominal values and, hence, $v_{j,t}^i$ is transformed into $I_{V_j}^k$ and $v_{j,t-\tau}^i$ is transformed into $I_{V_j}^{k\prime}$. MTSC determines how different the conditional probability, $\Pr(I_{V_j}^k | I_{V_j}^{k\prime})$, is from the a priori probability $\Pr(I_{V_j}^k)$. If the difference is large, it means that when V_j takes on the value, $I_{V_j}^k$, it is usually preceded at τ instants earlier by $I_{V_j}^{k\prime}$ of the same variable. In this case, one can conclude that $I_{V_j}^k$ is temporally related to $I_{V_j}^{k\prime}$ with a position lag of τ. The stronger $I_{V_j}^k$

values are temporally related to $I_{V_j}^{k\prime}$, i.e., the differences between the conditional and a priori probabilities are greater. By comparing the probabilities of difference between $I_{V_j}^{k}$ and the previously observed values, $I_{V_j}^{k\prime}$, for all $\tau \le \tau_{max}$, where τ_{max} is the maximum time-lag the user chooses to explore, we can identify from T_j^i all values that are temporally related to $I_{V_j}^{k}$. These values constitute the intra-CTP patterns within all CUTS T_j^i, j =1, …., n in $\mathbf{T}^{\boldsymbol{i}}$.

Specifically, in order to discover intra-CTP within each variable in T_j^i, j =1, ..., n of MTS, T^i, MTSC defines the intra-CTP as a set of temporal patterns detected between a value, $I_{V_j}^{k}$, at a particular time instant, t, and those that it takes on $I_{V_j}^{k\prime}$ at time instant, τ, $1 \le \tau \le \tau_{max} < t$ ahead of time.

Definition 3-2 (Intra-CTP): Let $L_{j,t-\tau}^{i} = I_{V_j}^{k\prime}$ and $L_{j,t}^{i} = I_{V_j}^{k}$ be the intra-CTP between the $L_{j,t-\tau}^{i} = I_{V_j}^{k\prime}$ and $L_{j,t}^{i} = I_{V_j}^{k}$ for T_j^i, where $L_{j,t-\tau}^{i}$ represents the event in T_j^i and occurs in the time points of $t-\tau$; $L_{j,t}^{i}$ represents the event in T_j^i occurs after τ time intervals in the time points of t; τ is the time interval and $1 \le \tau \le t-1$, $\tau \in \mathbb{Z}^{+}$.

We denote the probability of an event $L_{j,t}^{i}$ as $Pr\left(I_{V_j}^{k}\right)$ and compute it using Eqn. (3.1), where $I_{V_j}^{k}$ represents the value of the states of T_j^i:

$$\Pr(I_{V_j}^{k}) = \frac{freq(I_{V_j}^{k})}{p} \qquad (3.1)$$

where p is the total number of time points.

Given that $L_{j,t}^{i}$ has occurred, the conditional probability of $L_{j,t-\tau}^{i}$ is denoted as $\Pr(I_{V_j}^{k}|I_{V_j}^{k'})$ and can be determined as indicated by Eqn. (3.2):

$$\Pr(I_{V_j}^{k}|I_{V_j}^{k'}) = \frac{freq(I_{V_j}^{k}, I_{V_j}^{k'})}{freq(I_{V_j}^{k'})} \tag{3.2}$$

Given these probability estimations, the magnitude of the difference between the conditional probability $\Pr(I_{V_j}^{k}|I_{V_j}^{k'})$ and the a priori probability $\Pr(I_{V_j}^{k})$ can be defined simply as Eqn. (3.3).

$$\Pr(I_{V_j}^{k}|I_{V_j}^{k'}) - \Pr(I_{V_j}^{k}) \tag{3.3}$$

However, since the characteristics of each variable are different, for the patterns to be detected more accurately, the differences in the two probabilities are normalized as Eqn. (3.4) and *p* represents the total number of time points. (Chan, K.C.C. et al, 1994):

$$d_{j,\tau}^{i} = \frac{p\times\left(\Pr\left(I_{V_j}^{k}\middle|I_{V_j}^{k'}\right)-\Pr(I_{V_j}^{k})\right)}{\sqrt{p\times\Pr(I_{V_j}^{k})\Pr(I_{V_j}^{k'})\left(1-\Pr(I_{V_j}^{k})\right)\left(1-\Pr(I_{V_j}^{k'})\right)}} \tag{3.4}$$

Hence, the magnitudes of the normalized differences in conditional and a priori probabilities, $d_{j,\tau}^{i}$, $\tau = 1, \ldots, \tau_{max}$, can be treated as *Significant Discrepancy Measure* for Intra-FTP. The significance of the above temporal relationship depends on the magnitudes of normalized differences, $d_{j,\tau}^{i}$, each of which can be either ≥ 0 or ≤ 0. If $|d_{j,\tau}^{i}|$ is large, the presence or absence of $I_{V_j}^{k'}$ would likely imply that, at τ time instants later, the CUTS will or will not take on the state value $I_{V_j}^{k}$, respectively. The magnitudes of the normalized differences are between the conditional and a priori probabilities, $d_{j,\tau}^{i}$, $\tau = 1, \ldots, \tau_{max}$,, so that they capture the strengths of the temporal relationships and they constitute the intra-CTP T_j^i.

3.3.2 Discovering Inter-CUTS Temporal Patterns (CTP)

The inter-CTP can also be discovered in very much the same way as the intra-CTP have been discovered. By definition, the inter-CTP defined between two CUTS, say, T_j^i and $T_{j'}^i$, both within $\mathbf{T}^i$, consists of a set of temporal relationships or interrelationships detected between a value of T_j^i at a particular time instant, t, and those that it takes on at an earlier time instant, τ, $1 \leq \tau \leq \tau_{max} < t$. These inter-CTP can be defined as follows.

Definition 3-3 (Inter-CTP): Let $L_{j',t-\tau}^i = I_{V_{j'}}^{k'}$ and $L_{j,t}^i = I_{V_j}^k$ be the inter-CTP between the $L_{j',t-\tau}^i$ and $L_{j,t}^i$, where $L_{j',t-\tau}^i$ represents the value of variable $T_{j'}^i$ at the time points of $t-\tau$; $L_{j,t}^i$ represents the value of variable T_j^i at the time points of t, τ represents the time interval and $1 \leq \tau \leq t-1$, $\tau \in \mathbb{Z}^+$.

If event $L_{j,t}^i$ of a variable of an MTS occurs and if it is often followed by another event $L_{j',t-\tau}^i$ after τ time intervals, then MTSC considers that it is possible that temporal patterns exist between them and this relationship is referred to as inter-CTP.

Therefore, given a value, say, $I_{V_j}^k$, in T_j^i in $\boldsymbol{T^i}$ at time points t, and another value, say, $I_{V_{j'}}^{k'}$, in $T_{j'}^i$ at $\boldsymbol{T^i}$ at time points t- τ**,** MTSC determines how much $I_{V_j}^k$ is related to $I_{V_{j'}}^{k'}$ by determining how different the conditional probability, $\Pr(I_{V_j}^k | I_{V_{j'}}^{k'})$ is from the priori probability $\Pr(I_{V_j}^k)$. Since the magnitude of the difference between the conditional probability, $\Pr(I_{V_j}^k | I_{V_{j'}}^{k'})$, and the a priori probability $\Pr(I_{V_j}^k)$ can be defined simply as $\Pr(I_{V_j}^{k}|I_{V_{j'}}^{k'}) - \Pr(I_{V_j}^{k}))$, the differences between the two probabilities can be normalized to form a test statistic as shown in Eqn. (3.5) so as to be able to detect the temporal patterns more accurately. p represents the total number of time points in the equation.

$$d_{jj',\tau}^{i} = \frac{p\times\left(\Pr(I_{V_j}^{k}|I_{V_{j'}}^{k'})-\Pr(I_{V_j}^{k}))\right)}{\sqrt{p\times\Pr(I_{V_j}^{k})\Pr(I_{V_{j'}}^{k'})\left(1-\Pr(I_{V_j}^{k})\right)\left(1-\Pr(I_{V_{j'}}^{k'})\right)}} \tag{3.5}$$

Algorithm 1 : Intra-CTP and inter-CTP discovery

```
Input:T = {T^1, T^2, ..., T^m}(m is the number of MTS)
    T^i = {T^i_1, T^i_2, ..., T^i_n} (n is the number of variable)
    τ(the time window 1≤τ< t)
Output: finalResult(feature vector for one MTS)

for each MTS T^i in T
    Discretization using Equal Frequency for T^i
    for each univariate time series T^i_j in T^i
        Calculate d^i_{j,τ} /*degree of intra-CTP*/
        Result + = d^i_{j,τ}
    end
    for each channels T^i_{j',t} ≠ T^i_{j,t}  in MTS
        Calculate d^i_{jj',τ} /*degree of inter-CTP*/
        Result + = d^i_{jj',τ}
    end
    finalResult += Result
end /*finalResult is a set of Result for all MTSs*/

classification continue.
```

Figure 3.2 The Pseudo Code for Pattern Discovery from MTS Data

Similarly, the magnitudes of the normalized differences in conditional and a priori probabilities, $d_{j,\tau}^{i}$, $\tau = 1, \dots, \tau_{max}$, can be treated as *Significant Discrepancy Measure* for Inter-FTP. The significance of the temporal relationship depends on the magnitudes of normalized differences, $d_{jj',\tau}^{i}$, each of which can be either ≥ 0 or ≤ 0. If $|d_{jj',\tau}^{i}|$ is large, the presence or absence of $I_{V_{j'}}^{k'}$ would likely imply that at τ time instants later, the CUTS will or will not take on the value $I_{V_j}^k$, respectively. The $d_{jj',\tau}^i$, $\tau = 1, \dots, \tau_{max}$, constitutes the inter-CTP for T_j^i. The pseudo code is shown in Figure.3-2.

3.4 Classification for MTS

Once all the temporal patterns have been discovered, for MTS, we get a set of intra-temporal patterns measure $d_{j,\tau}^i$, (i = 1, 2, …, m, j = 1, 2, …, n, and τ = 1,2,… p) and a set of measured inter-temporal patterns, $d_{jj',\tau}^i$ (i = 1, 2, …, m; jj' = 1, 2, …, n, j ≠j', τ = 1,2,… p_{ij}), associated with it. These intra- and inter-CTP form feature vectors:[$d_{1,\tau}^i, \dots d_{j,\tau}^i, d_{1\ 2,\tau}^i, d_{1\ 3,\tau}^i, \dots, d_{jj\prime,\tau}^i, \dots d_{m-1\ m,\tau}^i$],where n is the total number of variables and 1≤$\tau \le \tau_{max}$ and i = 1, 2, …, m.

After extracting features from original MTS data, a classifier can be applied to classify the output feature vectors. There are two Main classical classification algorithms Support Vector Machine (SVM) (Cortes, C. et al. 1995) and Artificial Neural Networks (ANN) (Cohen, S. et al. 2003). An SVM transforms original input data into a higher dimensional space using a form of nonlinear mapping and then searches for a linear separating hyper-plane (Cortes, C. et al. 1995). There are several kernel functions that an SVM can use, e.g., polynomial, radial basis function (RBF) and hyperbolic tangent. A RBF kernel is a popular kernel function and is commonly used in support of vector machines (Chang, Y. W. et al, 2010). An ANN is composed of interconnecting artificial neurons that can compute values from inputs. Multi-Layer Perceptron (MLP) is generally widely used in neural network architecture meant for a distributed approach (Cohen, S. et al. 2003). Therefore, MTSC makes use of SVM with RBF kernel techniques along with MLP ANN to classify the feature vectors discovered from the intra- and inter-CTPs.

3.5 Experimental Result

To evaluate the performance of MTSC, we used both synthetic and real datasets. Using a synthetic dataset, experiments involving an evaluation of the effectiveness of MTSC for noisy and missing data were conducted. Next, using the real datasets from different applications, we examined the effectiveness of MTSC first, and then compared the accuracies of several different algorithms with respect to multivariate time series classification. The following presents the results and a discussion of our findings from the experiments performed.

3.5.1 Evaluation Methods

Other than the experiments performed to evaluate the effectiveness of MTSC, we also performed a number of tests to determine how accurate MTSC is in classifying multivariate time series data. Our tests included the following:

i) Raw-based Classification: We combined all the CUTS in MTS into one sequence, and then K-NN with Euclidean distance measure is used to classify MTS directly (Raw-based K-NN).

ii) Feature-based Classification: We compared the Method proposed in (Fulcher, B. D. et al, 2014). The original MTS were projected on to the PCA (principal component analysis) subspace using Locality Preserving Projection (LPP). The Matlab code available in (He, X. et al. 2003) was first used during implementation and then an SVM classifier with RBF was used for classification (Feature-based SVM).

iii) Model-based Classification: The univariate time series data was modeled using ARMA (Auto-Regressive Moving Average). The eight Linear Predictive Coding (LPC) coefficients (Fulcher, B. D. et al, 2014) are then obtained from ARMA model. After combining LPC, all the coefficients in CUTS, the SVM classifier was used for classification.

iv) MTSC: The proposed algorithm was used to discover intra-/inter-CTP in MTS and then, different classifiers, the SVM with RBF and MLP-ANN, were used for classification.

The performance measures we used for performance assessment and comparison were the popular *Precision*, *Recall* and *Classification Accuracy* (Powers, D.M.W. 2011). All possible measures were used to assess the classification performance based on the confusion matrix shown in Table 3.1.

For classification performance, since the "correct" class membership is known, *Accuracy* is the most intuitive measure that can be used to evaluate the effectiveness of a classification algorithm. Based on the confusion matrix shown in Table 3.1, the accuracy can be defined as shown in Eqn. (3.6) (Rosenberg, A. 2009).

Table 3.1 Confusion Matrix for Evaluation

		Prediction				Total
		C_1	C_2	…	C_n	
Actual	C_1	TP_1	FN_{12}	…	FN_{1n}	N_1
	C_2	FN_{21}	TP_2	…	FN_{2n}	N_2
	…	…	…	…	…	…
	C_n	FN_{n1}	FN_{n2}	…	TP_n	N_n
Total		N'_1	N'_2	…	N'_n	N

$$ACC. = \frac{\sum_i TP_i}{\sum_i \sum_j FN_{ij} + \sum_i TP_i} * 100\% \qquad (3.6)$$

Next, for a given class, C_i, *Precision* is utilized to evaluate how many instances were correctly predicted (Powers, D.M.W. 2011), which can be defined as (3.7).

$$Precision_i = \frac{TP_i}{TP_i + \sum_j FN_{ji}} \tag{3.7}$$

Similarly, for all instances that should have a class label C_i, *Recall* is used to evaluate how many of these were correctly captured (Powers, D.M.W. 2011), as defined in Eqn. (3.8):

$$Recall_i = \frac{TP_i}{TP_i + \sum_j FN_{ij}} \tag{3.8}$$

In this study, we used the *average of Precision* (*AveP*) and *average of Recall* (*AveR*) to evaluate the final classification results. Hence, the final *AveP* could be calculated as $\sum_i \frac{TP_i}{N'_i}/n$, and AveR as $\sum_i \frac{TP_i}{N_i}/n$.

Given these definitions, it may be noted that both *AveP* and *AveR* take on values in the interval [0, 1]; the closer the value is to 1, the better the classification quality it reflects. For the purpose of performance evaluation, 80% data are selected randomly as training data and the rest 20% as testing data. We iterated the classification process to obtain the final average value of the above measurements.

3.5.2 Performance Evaluation of MTSC Using Synthetic Datasets

For the synthetic dataset, we generated a dataset consisting of 45 MTS belonging to three classes, C_1, C_2 and C_3, and the rules were inserted in the classes. The data were generated in such a way that the patterns, as defined in Figure. 3.3, were embedded into the data. Each of multiple time series consists of five component time series, denoted as v_1, v_2, v_3, v_4 and v_5. A total of 500 data values corresponding to the data collected at 500 consecutive time instants were generated for each variable. Using MTSC, all the patterns shown in Figure. 3.3 were successfully discovered first before applying the classification algorithm SVM or ANN.

	Rules
C1	*1. v_1 and v_5 are generated randomly.* *2. v_4 takes on "A_4" at every interval of 2 time units and then V_2, at next time point, is generated to be "B_2", others are "A_2" or "C_2" randomly.* *3. If v_4 not to be "B_4", v_3 takes on "A_3" at the next time instant 50% or "C_3" at the next time instant 50% of the time.*
C2	*1. $v_2$and v_5 are totally random.* *2. If v_2 in A_2 then v_1 in B_1, others are random* *3. If v_2 in B_2, v_3 takes values in C_3, others are random* *4. If v_2 in C_2, then v_4 in A_4 others are random*
C3	*1. $v_3$and v_5 are totally random.* *2. If $v_3$in A_3 then v_2 in C_2, others are random* *3. v_1 takes on A_1at every interval of 3 time units and then at the next time points, v_4 is generated to be within A_4, others are random.*

Figure 3.3 Patterns embedded in synthetic data

Since each rule was inserted at one time instant, we discovered all temporal patterns at that instant. Therefore, we set τ=1 and considered the intra-CTP within one variable and inter-CTP among different variables only between the previous time point and the next time point. We used the proposed feature extraction to process the MTS data firstly and then used SVM or ANN to classify them. As for performance benchmarking, we compared MTSC with several algorithms corresponding to the raw-based, the feature-based and the model-based approaches described in Section 3.5.1. Table 3.2 summarizes the classification results produced by using different classifiers on the synthetic dataset for τ=1. The results table shows the value of Acc. (the average of classification accuracy), AveP. (the average of precision) and AveR (the average of recall).

Table 3.2 Comparing the Results from the Synthetic Dataset

Algorithms	**Evaluation**		
	ACC.	**AveP**	**AveR**
Raw-based (1-NN)	62.52%	0.63	0.51
Feature-based (SVM)	75.12%	0.72	0.81
Model-based (SVM)	77.27%	0.82	0.79
MTSC (SVM)	92.83%	0.92	0.96
MTSC (NN)	96.15%	0.97	0.97

We can conclude from the experimental results that, when MTSC was applied with MLP-ANN classifier, the classification accuracy was the highest (96.15%) with AveP being 0.97 and with AveR being 0.97. The average of accuracy is 92.83% with AveP being 0.92 and AveR being 0.96 when SVM is used as classifier.

Compared with other approaches, the raw-data based approach has obtained the lowest average of accuracy (62.52%) with AveP being 0.63 and AveR being 0.51. The performance of the feature-based approach is clearly superior with an accuracy of 75.12% with the AveP being 0.72 and AveR being 0.81, which is higher than the classification accuracy achieved by the raw-based approach. Even with PCA, the pattern relationships between values of different variables are averaged out and become less obvious. Nevertheless, this approach is better than the raw data based approach as PCA uses linear projection to transform the original MTS into a feature vector to reduce the dimensions effectively.

The final approach is the model-based approach using ARMA. The average of classification accuracy is 77.27% with AveP being 0.82 and AveR being 0.79. In this approach, the use of the 8 coefficients of ARMA to describe each univariate time series of original MTS. When data dimensionality is high, there will be greater loss of information, and this approach does not consider enough number of relationships between different variables. Given that MTSC takes such a relationship into consideration explicitly, it performs much better than the other algorithms in terms of both classification accuracies (92.83% and 96.15%), AveP and AveR.

In some special cases, it is possible that the data parameters may not have been available at the time of sampling. In such a case, time series like those unevenly sampled time series to be broken (see Figure 3.4) will appear. Since the ability of an algorithm to handle missing data is important, for MTSC to be useful, it has to be very robust and has to be able to cope well even in the presence of stock data containing a high proportion of missing data.

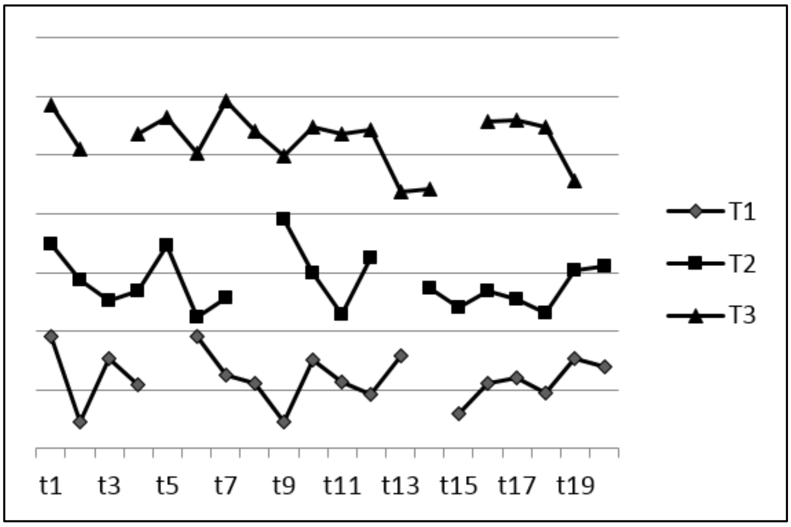

Figure 3.4 Example of Missing or Unevenly Sampling Data

To determine how well MTSC can cope with such datasets, we performed several additional experiments. Specifically, we first removed 10%, and then gradually increased it by 10% up to 50%, of the data values, randomly from the synthetic dataset in a series of experiments to see if MTSC could still classify the multivariate time series into corrected pre-defined classes.

For the experiments, we used 80% MTS that selected randomly from synthetic data for training and the rest of 20% for testing and iterated experiments to determine the average classification accuracy. The results, in terms of classification accuracy, are shown in Figures 3-5. As we continue to remove more and more of the data values from 10% to 20 % and then 30%, etc., until 50% of the data values are removed, we note that the accuracies remain within the 93% to 97% range even when up to 30% of the data values were removed. Even with 50% of the data values removed, MTSC is still able to classify with an accuracy of 95% when ANN is used as classifier and 92% when SVM is used as classifier. A close examination of the patterns shows that even if there is a high percentage of missing values in the synthetic dataset, MTSC is still able to discover most of the intra- and inter-temporal patterns embedded in the synthetic data and classify the multivariate time series data effectively.

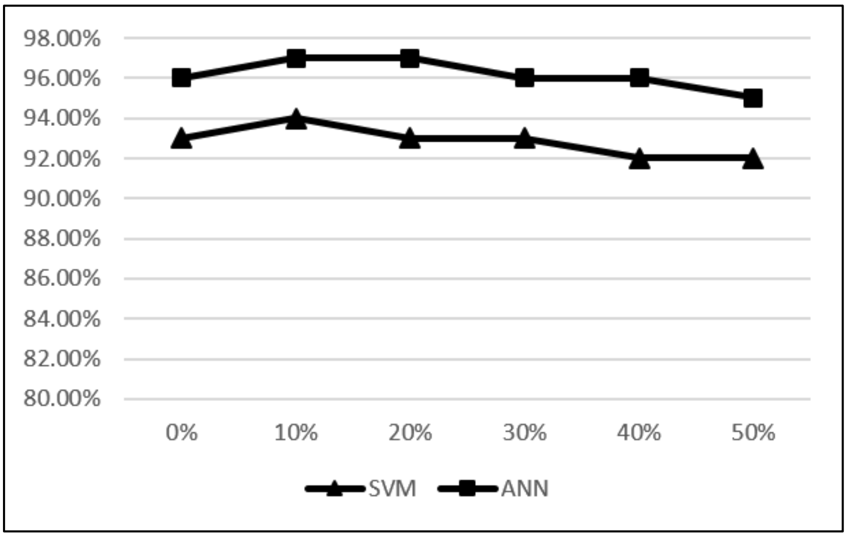

Figure 3.5 Results for Synthetic Data with Missing Values

3.5.3 Further Experiments Using Real Data

In this section, we consider several applications for multivariate time series classification drawn from the following application scenarios: i) EEG classification (Epilepsy Data); ii) ECG Classification (ECG Data); iii) Physical Action Classification (EMG Data); and iv) Classification for Wafer (Wafer Data). Using these datasets, our first set of experiments concerned the evaluation of the effectiveness of MTSC. In addition, we compared the performance of MTSC application to other multivariate time series classification algorithms.

3.5.3.1 Datasets Description

Epilepsy Datasets

The Epilepsy Dataset, *EPL Data*, used in our experiments is taken from (Rosenberg, A. 2009). It consists of multi-channel EEG data collected as focal signals and non-focal signals from epilepsy patients. In the experiments, data were collected only from the Fz and Pz channels corresponding to the extracranial reference electrodes. The EEG signals were sampled at 1024 Hz. The data were collected using 50 focal and 50 non-focal signals. In other words, the data obtained is of size 100 (trials) $\times 2$ (channels) $\times$ 1024 (time points).

Wafer Dataset

The wafer database comprises of a collection of time-series datasets where each file contains the sequence of measurements recorded by one vacuum-chamber sensor during the etch process applied to one silicon wafer during the manufacture of semiconductor microelectronics (Olszewski, R.T. 2004). Each wafer has an assigned classification: normal or abnormal. The abnormal wafers are representative of a range of problems commonly encountered during semiconductor manufacturing. There are a total of 254 observed wafers and for each wafer, six channels with around 150 time points are obtained. Hence, the size of data is 254(wafer) ×6(channels) ×~150 (time points).

EMG Dataset

Three male and one female subjects (age 25 to 30), who have experienced aggression in scenarios such as physical fighting, took part in the experiment (Murph P. M. et al. 1999). Throughout 20 individual experiments, each subject had to perform ten normal and ten aggressive activities. Regarding the rights of the subjects involved, certain ethical regulations and safety precautions based on the code of ethics of the British psychological society were followed. The overall number of electrodes was 8, which corresponded to 8 input time series and one for a muscle channel (ch1-8). Hence the size of database was 80 (experiments) ×8(channels) × ~10000 samples.

ECG Dataset

The ECG database comprises of a collection of time-series datasets where each file contains the sequence of measurements recorded by one electrode during one heartbeat (Olszewski, R.T. 2004). Each heartbeat has an assigned classification of normal or abnormal: 200 (experiments) × 2 (channels) × ~70 (time points).

3.5.3.2 Experimental Result Using MTSC for Real World Datasets

With the above datasets, we performed several experiments using the proposed MTSC first. Considering the dependency or relationship within one CUTS or between different CUTS may not be only in one time interval, we set $\tau = 1$ to 5 for intra-/inter-CTP. For each dataset, after discovering temporal patterns using the proposed algorithm, the set of *Significant Discrepancy Measure* was used to construct the feature vector for each MTS. Next, two popular classification algorithms, ANN and SVM, were used to classify the feature vectors.

As shown in Table 3.3, the average accuracy can go as high as 94.15% using Wafer data when ANN is applied. The average accuracy is 93.1% with the AveP being 0.94 and the AveR being 0.95 when SVM is applied for the same dataset. As for the EMG dataset, MTSC can achieve the second highest average classification accuracy of 90.97% with the AveP being 0.9 and the AveR being 0.9 for ANN classifier. As for the EEG dataset, the classification accuracy is 85.37% with the AveP being 0.85 and AveR being 0.85 for the ANN classifier, and the classification accuracy is 84.57% with the AveP being 0.84 and AveR being 0.85 for the SVM classifier. Finally, as for the ECG data set, the classifier was able to achieve a slightly higher average classification accuracy of 76.22% with the AveP being 0.69 and AveR being 0.73 for the SVM classifier, and accuracy of 75.36% with the AveP being 0.73 and AveR being 0.71 for the ANN classifier.

In summary, for most data sets, ANN can get higher classification result than SVM for most classification results.

Table 3.3 Experimental Result Using MTSC on Real World Datasets

Evaluation			$\tau = 1$	$\tau = 2$	$\tau = 3$	$\tau = 4$	$\tau = 5$	**Average**
EEG Data	**ANN**	**ACC.**	80.95%	85.00%	90.00%	94.44%	76.47%	**85.37%**
		AveP	0.80	0.80	0.91	0.95	0.77	**0.85**
		AveR	0.80	0.80	0.91	0.94	0.78	**0.85**
	SVM	**ACC.**	76.47%	83.33%	85.00%	92.86%	85.19%	**84.57%**
		AveP	0.74	0.84	0.82	0.94	0.86	**0.84**
		AveR	0.78	0.83	0.85	0.93	0.85	**0.85**
Wafer Data	**ANN**	**ACC.**	94.23%	90.48%	98.18%	97.83%	90.01%	**94.15%**
		AveP	0.94	0.90	0.98	0.98	0.91	**0.94**
		AveR	0.94	0.91	0.98	0.98	0.90	**0.94**
	SVM	**ACC.**	92.16%	92.86%	94.87%	94.34%	91.29%	**93.10%**
		AveP	0.92	0.97	0.94	0.94	0.91	**0.94**
		AveR	0.93	0.98	0.96	0.94	0.92	**0.95**
EMG Data	**ANN**	**ACC.**	93.33%	92.86%	82.61%	92.31%	93.75%	**90.97%**
		AveP	0.96	0.90	0.83	0.88	0.95	**0.90**
		AveR	0.83	0.95	0.83	0.95	0.93	**0.90**
	SVM	**ACC.**	92.31%	76.92%	73.91%	84.61%	78.95%	**81.34%**
		AveP	0.88	0.63	0.75	0.75	0.67	**0.74**
		AveR	0.95	0.88	0.82	0.91	0.88	**0.89**
ECG Data	**ANN**	**ACC.**	77.14%	77.08%	73.92%	75.68%	72.97%	**75.36%**
		AveP	0.66	0.72	0.72	0.71	0.65	**0.69**
		AveR	0.78	0.73	0.72	0.74	0.67	**0.73**
	SVM	**ACC.**	80%	81.82%	71.79%	75.76%	71.74%	**76.22%**
		AveP	0.78	0.80	0.67	0.72	0.69	**0.73**
		AveR	0.77	0.74	0.68	0.66	0.68	**0.71**

3.5.4 Comparison Results

In addition, we also performed several experiments that aimed at comparing the classification accuracies between MTSC with other algorithms used for classifying multivariate time series data. The tests included: (i) classify each MTS using K-NN with Euclidean distance measure independently to determine average classification accuracy (Raw-based KNN); (ii) classify based on extracted feature vector using PCA (Fulcher, B. D. et al, 2014) with SVM classifier (Feature-based SVM), (iii) classify each MTS using an SVM classifier based on LPC coefficients extracted from ARMA model (Fulcher, B. D. et al, 2014) (Model-based SVM) and (iv) classify the MTS on the basis of the proposed MTSC and the discovered temporal patterns using two different classifiers, SVM and ANN (MTSC SVM and MTSC ANN). In our experiments, 80% of the available data were selected randomly as training data and the rest of the 20% was selected as testing data. We used performance assessment and comparison methods described in 3.5.1 and the results are shown in Table 3.4.

Table 3.4 Comparison Result on Real World Datasets

Evaluation		**NN**		
		ACC.	**AveP**	**AveR**
EEG Data	**Raw-based (1-NN)**	59.11%	0.71	0.72
	Feature-based (SVM)	61.08%	0.61	0.61
	Model-based (SVM)	62.5%	0.63	0.63
	MTSC (ANN)	85.37%	0.85	0.85
	MTSC (SVM)	84.57%	0.84	0.85
Wafer Data	**Raw-based (1-NN)**	87.76%	0.88	0.89
	Feature-based (SVM)	91.49%	0.91	0.92
	Model-based (SVM)	90.69%	0.91	0.91
	MTSC (ANN)	94.15%	0.94	0.94
	MTSC (SVM)	93.10%	0.94	0.95
EMG Data	**Raw-based (1-NN)**	53.85%	0.63	0.73
	Feature-based (SVM)	78.57%	0.79	0.78
	Model-based (SVM)	78.57%	0.77	0.79
	MTSC (ANN)	90.97%	0.9	0.9
	MTSC (SVM)	81.34%	0.74	0.89
ECG Data	**Raw-based (1-NN)**	75.61%	0.74	0.72
	Feature-based (SVM)	72.19%	0.65	0.73
	Model-based (SVM)	74.36%	0.71	0.72
	MTSC (ANN)	75.36%	0.69	0.73
	MTSC (SVM)	76.22%	0.73	0.71

Similar to the results from the synthetic dataset, for all the real datasets, the proposed algorithm with ANN achieved the highest classification accuracy with highest *AveP* and *AveR*. When the raw-based algorithm is applied, the classification accuracy for classifying short time series data, such as ECG data set, is better than the result from classifying long time series data, such as EEG and EMG data set. When feature-based and model-based algorithms are applied, the classification result could be improved. Only significant features remained after feature extraction, which could filter noisy data from the original MTS. However, since the existing feature extraction methods or models ignore the relationships between different variables, the performance of classification turns out to be the best when the proposed MTSC is applied.

The experimental result depends on both the performance of the proposed algorithm and the datasets in different applications. For example, the "Wafer dataset" and "ECG dataset" are the clean datasets which have been used in previous research by other researchers. So the differences is not quite high when comparing the MTSC with other algorithm in this two datasets. However, the EEG datasets is the original datasets which presents source signals, so when we use the same preprocessing methods before both proposed algorithm and other traditional approach, MTSC performs particularly well for EEG.

Besides classification performance comparison, complexity analysis is another significant target. Suppose that we have *m* MTS with *n* variables and the length of time series is *t* for each univariate time series, then $n \ll t$. If no feature extraction method is used, the classifier has to process for *mnt* dimensions data. When K-NN is used for classification process, the time complexity is *O(kmnt)*. When some feature extraction method, such as PCA, is used, the run-time complexity of the PCA is $O(mt^2)$. After feature extraction, the size of the feature vector should be n^2, so when SVM is used for classification, the time complexity is for the classification process and that for the feature-based algorithm O(m^2n^2), i.e., the time complexity is O(mt^2+m^2n^2) in total. Finally, the run-time for complexity analysis by our proposed algorithm is $O(mt * n^2)$ (n<<t). After discovering the patterns, the size of feature vector should be n^2, so when SVM is used for classification, the time complexity for classification is O(m^2n^2). So, the total time complexity is O($mt * n^2$+m^2n^2) for the MTSC. In other words, for the MTS containing high dimensions of length and less variables, the MTSC only needs to count the value at each time point, thus reducing overall time complexity.

Figure 3.6 summarizes the results showing the performances of the different algorithms based on classification accuracies for four real world data sets. Based on these results, MTSC seems to perform consistently better than the other popular algorithms studied.

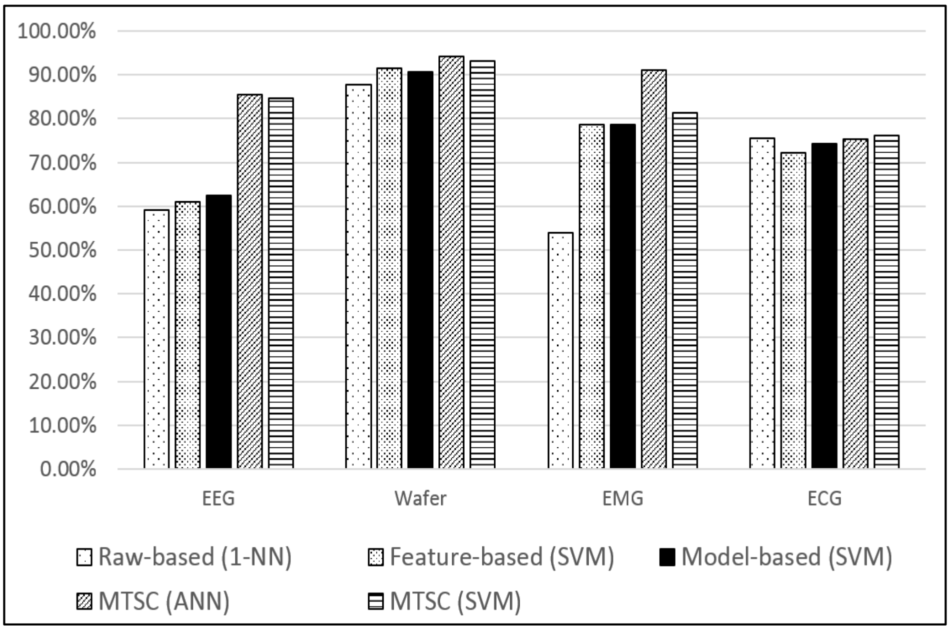

Figure 3.6 Performance of the Different Algorithms on Real World Datasets

Chapter 4.

A Fuzzy Approach for Discovering Patterns in Multivariate Time Series

Consider a set of data consisting of *M* MTS data sets that are pre-classified into *k* classes with each MTS consisting of *n* CUTS. A Fuzzy Multivariate Time Series Classification (FMTSC) algorithm is developed in this chapter to enable the discovery of Fuzzy Temporal Patterns/Associations (FTP). The algorithm performs its tasks by discovering fuzzy temporal patterns within and between MTS so that an SVM-based classification can be used to learn from these patterns discovered in each class of MTS for the purpose of classification.

In order to make the classification of MTS data more effective, the temporal patterns discovered are best modeled using fuzzy sets, so these patterns are fuzzy temporal patterns.

The task of our proposed algorithm is to uncover temporal relationships or interrelationships at different time instants within and between different variables of MTS. These temporal relationships or interrelationships constitute what we call *intra-fuzzy temporal patterns (intra-FTP)* and *inter-fuzzy temporal patterns (inter-FTP)* respectively. The proposed algorithm involves the following steps: i) fuzzify the MTS; ii) discover intra-FTP for each variable collected for each MTS; iii) discover the hidden inter-FTP between different variables of the same MTS; iv) classify or cluster the MTS into different classes based on the similarities and dissimilarities of measures associated with the discovered patterns. In this section, we describe each step of the FMTSC in more detail.

4.1 Fuzzification

To discover temporal patterns, the numerical values of MTS are first discretized into several intervals. Discretization can minimize the impact of noisy data during the data mining process (Ma. P. C. H. et al. 2011). It also can help smooth data to reduce noise (Han, J. et al. 2007), speed up classification and clustering process (Ramoni, M. et al. 2000) and make the clustering results more meaningful and easier-to-understand (Liao, T.W. 2005). This is why FMTSC starts by partitioning the domain of numerical data into a relatively small number of intervals and assigns a nominal value to each. However, a major drawback of this discretization process is that it generates crisp interval boundaries. Knowledge representation is not always based on such discretization; for example, we usually use certain linguistic terms (e.g. Tall, Middle, Short) for dividing height into some fuzzy categories. In our case, we also cannot use exact interval values to describe the stock prices. To eliminate this problem, FMTSC fuzzifies the data instead. In particular, we propose a fuzzy approach to transform original numerical MTS into linguistic terms with a fuzzy set. The triangular membership function is among the most common, fuzzy membership functions used in practice, for example, in both (Roy P. et al. 2015) and (Ijegwa, A.D. et al. 2014), the triangular membership function is used. Hence, we follow the idea of previous researchers and examine the triangular membership function in this study.

Definition 4-1 (Fuzzification Discretization): A set of linguistic terms, denoted as L_j^i= $(L_{j,1}^i,\ L_{j,2}^i, \dots L_{j,t}^i \dots, L_{j,p}^i)$, $L_{j,t}^i \epsilon L_j^i$ correspond to $v_{j,t}^i \in S_j^i$, where the domain of the values of L_j^i is denoted as $dom(L_j^i) = [L_{V_j}, U_{V_j}]$, and L_{V_j}represents the lower bound and U_{V_j} represents the upper bound of the values. Hence, the set of discrete states for $L_{j,t}^i$ is denoted as $T(L_{j,t}^i) = \{t_{jk} | k = 1,2, \dots l\}$, where t_{jk} are the discrete states characterized by a fuzzy set, F_{jk}, with membership function $\mu_{F_{jk}}$, l is the total number of states.

In FMTSC, they are defined by the three fuzzy sets shown in Figure 4.1. For each variable, v_{jmax}^i and v_{jmin}^i represent the maximum and minimum values that S_j^i takes on, respectively, and b_{j1}, b_{j2} and b_{j3} are the three boundaries.

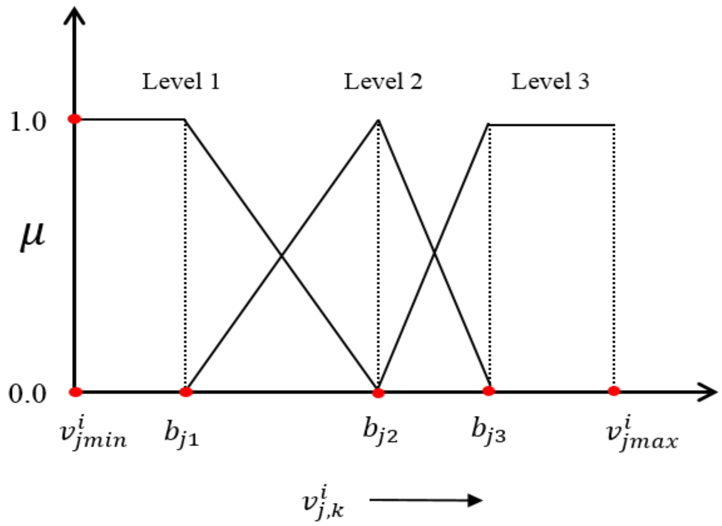

Figure 4.1 Membership functions for One CUTS

In defining the membership function, traditional discretization methods such as *Equal Width* and *Equal Frequency* are the two simplest discretization methods (Wong, A. K. C. et al. 1979) that could be adopted to define the initial boundaries. However, if outliers exist in the dataset, the *Equal Width* approach can turn out to be quite meaningless. Thus, for FMTSC, we use the *Equal Frequency* approach to determine the initial boundaries for the fuzzy sets (Roy, P. et al. 2015). The values of b_{j1} to b_{j3} are therefore calculated by dividing the data points to ensure that each of them have the same probability of falling into an interval defined without $v^i_{j,max} - v^i_{j,min}$. i.e. b_{j1} = $v^i_{j,min}$ +25% $\times$ ($v^i_{j,max}$ - $v^i_{j,min}$), b_{j2} = $v^i_{j,min}$ +50% $\times$ ($v^i_{j,max}$ - $v^i_{j,min}$) and b_{j3}=$v^i_{j,min}$+75% $\times$ ($v^i_{j,max}$-$v^i_{j,min}$). The degree of membership, $\mu_{F_{jk}}$, of a parameter value $v^i_{j,k}$ where S^i_j in S^i can be defined and computed as shown in Figure 4.2.

$$\mu_{F_{j1}}(v^i_{j,k}) \begin{cases} 1 & v^i_{j\,min} \le v^i_{j,k} \le b_{j1} \\ (b_{j2} - v^i_{j,k})/(b_{j2} - b_{j1}) & b_{j1} \le v^i_{j,k} \le b_{j2} \\ 0 & v^i_{j,k} \ge b_{j2} \end{cases}$$

$$\mu_{F_{j2}}(v^i_{j,k}) \begin{cases} 0 & v^i_{j\,min} \le v^i_{j,k} \le b_{j1} \\ (v^i_{j,k} - b_{j1})/(b_{j2} - b_{j1}) & b_{j1} \le v^i_{j,k} \le b_{j2} \\ 1 & v^i_{j,k} = b_{j2} \\ (b_{j3} - v^i_{j,k})/(b_{j3} - b_{j2}) & b_{j2} \le v^i_{j,k} \le b_{j3} \\ 0 & b_{j3} \le v^i_{j,k} \le v^i_{j\,max} \end{cases}$$

$$\mu_{F_{j3}}(v^i_{j,k}) \begin{cases} 0 & v^i_{j,k} \le b_{j2} \\ (v^i_{j,k} - b_{j2})/(b_{j3} - b_{j2}) & b_{j2} \le v^i_{j,k} \le b_{j3} \\ 1 & b_{j3} \le v^i_{j,k} \le v^i_{j\,max} \end{cases}$$

Figure 4.2 Degree of Membership

4.2 Discovering Intra-Fuzzy Temporal Patterns (Intra-FTP)

With the given data related to one MTS, the values that a particular variable takes can be expected to be related or interrelated with the previous values of this variable or with other variables. These relationships or interrelationships considered here constitute, the intra-FTP and inter-FTP for one MTS, respectively.

In order to discover intra-FTP within each CUTS (T_j^i, j =1, ..., n, of T^i), the intra-FTP are defined as consisting of a set of temporal relationships detected between a value, t_{jk}, at a particular time instant, t, and what it takes on, $t_{jk'}$ at an earlier time instant, τ, $1 \leq \tau \leq \tau_{max} < t$.

Definition 4-2 (Intra-FTP): Let $P(L_{j,t-\tau}^i = t_{jk} \rightarrow L_{j,t}^i = t_{jk'})$ be the Intra-FTP between the $L_{j,t-\tau}^i = t_{jk}$ and $L_{j,t}^i = t_{jk'}$ for T_j^i. Here $L_{j,t-\tau}^i$ represents the event in T_j^i occurs in the time points of $t-\tau$; $L_{j,t}^i$ represents the event in T_j^i occurs after τ time intervals between the time points of t; τ is the time interval and $1 \leq \tau \leq t-1$, $\tau \in \mathbb{Z}^+$.

The probability of an event $L_{j,t}^i$ is denoted as $Pr(t_{jk})$ and estimated as Eqn. (4.1). t_{jk} represents the value of the states of T_j^i:

$$\Pr(t_{jk}) = \frac{\sum_{n=1}^{p} \mu_{F_{jk}}}{\sum_{t=1}^{p} \sum_{k=1}^{l} \mu_{F_{jk}}} \tag{4.1}$$

where p is the total number of time points, l is the total number of states CUTS T_j^i has.

A conditional probability measures the probability of an event, $L_{j,t-\tau}^i$, given that another event, $L_{j,t}^i$, has occurred, which is denoted as $Pr(t_{jk}|t_{jk'})$, t_{jk} represents the value of the states for $L_{j,t}^i$, and $t_{jk'}$ represents the value of the state for $L_{j,t-\tau}^i$. $Pr(t_{jk}|t_{jk'})$ can be estimated as Eqn. (4.2).

$$\Pr(t_{jk}|t_{jk'}) = \frac{\Pr(t_{jk}, t_{jk'})}{\Pr(t_{jk'})} \tag{4.2}$$

where $\Pr(t_{jk}, t_{jk'})$ is defined as Eqn. (4.3).

$$\Pr\left(t_{jk}, t_{jk'}\right) = \frac{\sum_{n=1}^{p} \min(\mu_{F_{jk}}, \mu_{F_{jk'}})}{\sum_{t=1}^{p} \sum_{k=1}^{l} \sum_{k'=1}^{l} \min(\mu_{F_{jk}}, \mu_{F_{jk'}})} \tag{4.3}$$

where $Pr(t_{jk'})$ can be estimated by Eqn. (4.1) and p is the total number of time points the pattern has.

Given the probability estimations, decide whether the fuzzy temporal pattern $P(L_{j,t-\tau}^{i} = t_{jk} \rightarrow L_{j,t}^{i} = t_{jk'})$ is interesting. For this, we need to determine whether the magnitude of the difference between the conditional probability $Pr(t_{jk}|t_{jk'})$ and the priori probability $Pr\left(t_{jk}\right)$ is statistically significant. The magnitude of the difference between conditional probability and the a priori probability can estimate the dependency between $L_{j,t-\tau}^{i}$ and $L_{j,t}^{i}$, which can be defined simply as Eqn. (4.4).

$$\Pr\left(t_{jk} \middle| t_{jk'}\right) - \Pr(t_{jk}) \tag{4.4}$$

As the characteristics of CUTS are different, in order to detect the patterns more accurately, the differences in the two probabilities are normalized as Eqn. (4.5) (Chan, K.C.C. et al, 1994).

$$d_{j,\tau}^{i} = \frac{\Pr\left(t_{jk} \middle| t_{jk'}\right) - \Pr(t_{jk})}{\sqrt{\Pr(t_{jk})\Pr(t_{jk'})\left(1 - \Pr(t_{jk})\right)\left(1 - \Pr(t_{jk'})\right)}} \tag{4.5}$$

The magnitudes of the normalized differences can capture the strengths of the temporal relationships. If $|d_{j,\tau}^{i}|$ is large, the presence or absence of $L_{j,t-\tau}^{i}$ can ascertain whether, τ time instants later, the variable will or will not take on the value $L_{j,t}^{i}$, respectively. Hence, the magnitudes of the normalized differences in conditional and a priori probabilities, $d_{j,\tau}^{i}$, $\tau = 1, \ldots, \tau_{max}$, can be treated as *Significant Discrepancy Measure* for Intra-FTP. In this study, all the patterns for different states' values (i.e.

$d_1, d_2, \dots d_i, \dots d_n$, n is the total number of the states) of the same MTS are detected as $P(L^i_{d_i,t-\tau} \rightarrow \ L^i_{d_i,t})$.

4.3 Discovering Inter-Fuzzy Temporal Patterns (Inter-FTP)

Definition 4-3 (Inter-FTP): Let $P(L^i_{j',t-\tau} = t_{j'k} \rightarrow \ L^i_{j,t} = t_{jk'})$ be the Inter-FTP between the $L^i_{j',t-\tau}$ and $L^i_{j,t}$, where $L^i_{j',t-\tau}$ represents the event in T^i_j occurs in the time points of $t-\tau$; $L^i_{j,t}$ represents the event in T^i_j occurs after τ time intervals, at the time points of t, and τ represents time interval and $1 \leq \tau \leq t-1$, $\tau \in \mathbb{Z}^+$.

When an event $L^i_{j,t}$ occurs and is followed by another event, $L^i_{j',t-\tau}$, occurring after τ time intervals, the proposed algorithm can determine how much of $L^i_{j,t}$ is related to $L^i_{j',t-\tau}$ by determining how different the conditional probability $Pr\left(t_{j'k}\middle|t_{jk'}\right)$ and the a priori probability $Pr(t_{j'k})$ are, where $t_{j'k}$ represents the value of the states for $L^i_{j',t-\tau}$, and $t_{jk'}$ represents the value of the state for $L^i_{j,t}$. Similarly the magnitude of the difference between the probabilities can be defined as $\Pr\left(t_{j'k}\middle|t_{jk'}\right) - \Pr(t_{j'k})$. For the patterns to be detected more accurately, the differences in the two probabilities are normalized as Eqn. (4.6).

$$d^i_{jj',\tau} = \frac{\Pr\left(t_{j'k}\middle|t_{jk'}\right) - \Pr(t_{j'k})}{\sqrt{\Pr(t_{j'k})\Pr(t_{jk'})\left(1-\Pr(t_{j'k})\right)\left(1-\Pr(t_{jk'})\right)}} \tag{4.6}$$

Similarly, if $|d^i_{j\ j',\tau}|$ is large, the presence or absence of $L^i_{j',t-\tau}$ would likely imply that τ time instants later, the variable will or will not take on the value $L^i_{j,t}$. Hence, the magnitudes of the normalized differences between conditional and a priori probabilities, $d^i_{jj',\tau}$, $\tau = 1, \dots, \tau_{max}$, can be treated as the *Significant Discrepancy Measure* for Inter-FTP. In this study, all the patterns for different states values (i.e., $d_1, \dots d_i, \dots d_j, \dots d_n$, n is the total number of the states) of the same CUTS are detected as $P(L^i_{d_j,t-\tau} \rightarrow L^i_{d_j,t})$ and the patterns for different states values between different CUTS are detected as $P(L^i_{d_i,t-\tau} \rightarrow L^i_{d_j,t})$. The pseudo code of the process of feature extraction using the proposed algorithm is shown in Figure 4.3.

Algorithm: Feature Extration By Using Discovered Intra-/Inter-FTP

Input: $\mathrm{T}=\{T^i | i = 1, 2, \dots N\}$, $T^i = \{T^i_1, T^i_2, \dots T^i_n\}$

Output: ***finalResult*** *(feature vectors for all EEGs)*

foreach T^i

Remove noise of EEG data using FastICA

Discretize data using fuzzy membership function

foreach channel T^i_j in each EEG with value $T^i_{j,t}$

Calculate significant discrepancy measure for intra-channel FTP with

$$d^i_{j,\tau} = \frac{\Pr(t_{jk}|t_{jk'}) - \Pr(t_{jk})}{\sqrt{\Pr(t_{jk})\Pr(t_{jk'})(1-\Pr(t_{jk}))(1-\Pr(t_{jk'}))}}$$

Result+=$d^i_{j,\tau}$

end

foreach channels $T^i_{j'}$, $(T^i_{j'} \neq T^i_j)$ with value $T^i_{j',t}$

Calculate significant discrepancy measure for intra-channel

FTP with $d^i_{jj',\tau} = \frac{\Pr(t_{j'k}|t_{jk'}) - \Pr(t_{j'k})}{\sqrt{\Pr(t_{j'k})\Pr(t_{jk'})(1-\Pr(t_{j'k}))(1-\Pr(t_{jk'}))}}$

Result+=$d^i_{jj',\tau}$

end

finalResult+=Result

end

Figure 4.3 Pseudo Code for Process of Discovering Intra-/Inter-FTP

4.4 Prediction Based on the Discovered Intra- and Inter-FTP

Since the discovered intra-FTP and inter-FTP are not completely deterministic, the uncertainty of associations between $L^i_{j,t-\tau}$ and $L^i_{j',t}$ can be modeled with the confidence measure defined as $Pr(t_{jk}|t_{j'k'})$ (*j*=*j'* for intra-FTA, and $j\neq j'$ for inter-FTA). Hence, in order to predict whether $L^i_{j',t}$ takes on the value of $t_{j'k}$ at a future time point using $L^i_{j,t-\tau}$=$t_{jk'}$, the weight of evidence measure (Osteyee, D.B. et al., 1974), $W(L^i_{j',t-\tau} = t_{j'k} \rightarrow L^i_{j,t} = t_{jk'})$, needs to be assigned to the association and defined in terms of mutual information, $I(L^i_{j,t-\tau}: L^i_{j',t})$, as Eqn. (4.7) shows.

$$W\left(L^i_{j',t-\tau} = t_{j'k} \rightarrow L^i_{j,t} = t_{jk'}\right) = I\left(t_{j'k}: t_{jk'}\right) - I(t_{j'\neg k}: t_{jk'}) \tag{4.7}$$

where $I\left(t_{j'k}: t_{jk'}\right) = log \frac{Pr\left(t_{j'k}|t_{jk'}\right)}{Pr\left(t_{j'k}\right)} = log \frac{Pr(t_{j'k}, t_{jk'})}{Pr\left(t_{j'k}\right) \cdot Pr\left(t_{jk'}\right)}$

and $I(t_{j'\neg k}: t_{jk'}) = log \frac{Pr\left(t_{j'\neg k}|t_{jk'}\right)}{Pr\left(t_{j'\neg k}\right)} = log \frac{Pr(t_{j'\neg k}, t_{jk'})}{Pr\left(t_{j'\neg k}\right) \cdot Pr\left(t_{jk'}\right)}$.

In the equation, *j* represents the *j*th variable of stock S^i and *k* represents the state value of *j*. The term $Pr(t_{j'k}, t_{jk'})$ is the probability of the observations that $t_{jk'}$ is followed by $t_{j'k}$ after τ time points (Eqn. (4.3)). Similarly, $Pr(t_{j\neg k}, t_{jk'})$ is the probability of an observation that $t_{jk'}$ is followed by $t_{j\neg k}$ later after τ time point. The weight of evidence measures the amount of positive or negative evidence that is provided by $L^i_{j',t-\tau} = t_{j'k}$ supporting or refuting $L^i_{j,t} = t_{jk'}$ being observed together. Because this measure is probabilistic, it can work effectively even when the periods of the stock data covered are unequal and contain missing or erroneous values.

If an association between $L^i_{j,t-\tau}$ and $L^i_{j\prime,t}$ is discovered, we can conclude that there is some evidence supporting the observations like: "*IF* the $L^i_{j,t-\tau} = t_{jk\prime}$ occurs, *THEN* $L^i_{j\prime,t} = t_{j\prime k}$ will occur after τ time points with the *WEIGHT* of evidence $\mathrm{W}'(L^i_{j,t-\tau} \rightarrow\ L^i_{j',t})$ which supports the prediction rules", and the term $\mathrm{W}'(L^i_{j,t-\tau} \rightarrow L^i_{j',t})$ can be defined as in Eqn. (4.8) (Chan, K.C.C. et al., 1991).

$$W'\left(L^i_{j,t-\tau} \rightarrow\ L^i_{j\prime,t}\right) = W\left(L^i_{j,t-\tau} \rightarrow\ L^i_{j\prime,t}\right) \times \mu_{F_{jk\prime}} \tag{4.8}$$

Combining all the weights provided by the associations that support the observation $L^i_{j',t}$ by computing a total weight (*TW*) defined as in Eqn. (4.9), the value of $L^i_{j\prime,t}$ can be predicted by computing the maximum value of total weights:

$$\begin{aligned} TW\left(L^i_{j,t-\tau} \rightarrow\ L^i_{j',t}\right) &= \sum W\left(L^i_{j,t-\tau} \rightarrow\ L^i_{j',t}\right) \\ &= \sum_{k=1}^{n} W\left(t_{jk\prime} \rightarrow\ t_{j\prime k}\right) \end{aligned} \tag{4.9}$$

where n is the total value of states of variable j in stock i.

4.5 Illustrating Association Analysis using Synthetic Data Set

In order to facilitate an appreciation of how the FMTSC works, we provide below an illustrative example using a synthetic dataset involving three parameters (A, B, C) with 10 time points as shown in Table 4.1.

The approach for fuzzification described above was used for the dataset. The degrees of membership were determined from the membership functions. In this case, we considered three values for each parameter so that variable: A takes on *High*, *Medium*, and *Low*; B takes on *Up, No Change, Down*; and C takes on *Large, Medium, Small*. The fuzzy sets with the degrees of membership can be obtained as shown in Table 4.2.

Table 4.1 Synthetic Dataset for Illustration

	A	B	C
1	0.82	-0.02	0.09
2	1	-0.08	0.53
3	0.21	-0.41	-0.21
4	0.57	0.34	0.46
5	0.94	-0.12	0.1
6	0.17	-0.01	0.39
7	0.73	0.39	0.45
8	0.21	-0.08	0.21
9	0.41	1	0.09

Table 4.2 Fuzzification Result

	A			B			C		
	H	*S*	*L*	*U*	*N*	*D*	*L*	*M*	*S*
1	1.00	0.00	0.00	0.00	0.00	1.00	0.55	0.45	0.00
2	1.00	0.00	0.00	0.00	0.00	1.00	1.00	0.00	0.00
3	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00
4	0.00	1.00	0.00	0.00	1.00	0.00	0.00	0.22	0.78
5	1.00	0.00	0.00	0.00	0.00	1.00	0.50	0.50	0.00
6	0.00	0.00	1.00	0.98	0.02	0.00	0.00	0.44	0.56
7	0.00	0.38	0.62	0.00	0.84	0.16	0.00	0.25	0.75
8	0.00	0.00	1.00	0.00	0.00	1.00	0.02	0.98	0.00
9	0.49	0.51	0.00	0.00	1.00	0.00	0.55	0.45	0.00

Note that the intra-FTP and inter-FTP have been discovered successfully by using FMTSC. In order to determine whether the fuzzy temporal association is actually interesting, FMTSC determines whether the magnitudes of the differences between the conditional probability and the a priori probability are statistically significant.

FMTSC first discovers the associations between the value of B at the previous and current time points as intra-FTP. In order to determine whether the relationship between $\mathrm{B}_{t-\tau} = \mathrm{U}$ and $\mathrm{B}_t = \mathrm{U}$ is significant, $\Pr(\mathrm{B}_{t-1} = \mathrm{U})$ is computed: it is found to be 0.123 and $\Pr(\mathrm{B}_{t-\tau} = \mathrm{U}, \mathrm{B}_t = \mathrm{U})$ is 0.00, and $\Pr(\mathrm{B}_{t-1} = \mathrm{U}|\mathrm{B}_t = \mathrm{U})$ is 0.00. The test statistics can now be calculated using Eqn. (4.5). In order to avoid the problem of the denominator being zero, we set the minimal probability as 0.01 instead of 0. Similarly, the test statistic is calculated for inter-FTP. All the values of the test statistics for the measures are shown in Table 4.3. For example, the value of test statistic is -0.40 for the association $\mathrm{B}_{t-1} = \mathrm{U}$ and $\mathrm{B}_t = \mathrm{U}$. The first row presents the values of the test statistic for the intra-FTP, and the subsequent two rows presents the values for the inter-FTP.

To predict the value of movement at the 9th time point, the relevant FTPs with their weights of evidence can be obtained using the previous time points used while training. For example, when we set $\tau = 1$, one of the prediction rules is generated as "A=H$\underset{\tau=1}{\longrightarrow}$U (W = 0.84)", which means "If the value of *the differences between maximum and minimum stock price* is *High* the previous day, then the *movement of stock price* will go *Up* on the next day with the value of *weight* as 0.84".

Table 4.3 Results Concerning Significant Discrepancy Measure

Measure	$\mathrm{B}_t = \mathrm{U}$			$\mathrm{B}_t = \mathrm{N}$			$\mathrm{B}_t = \mathrm{D}$		
$\mathrm{P}\ (\mathrm{B}_{t-1} = \mathrm{U/N/D} \rightarrow \mathrm{B}_t = \mathrm{U/N/D})$	-0.40	-0.41	1.06	1.54	-0.56	0.13	-0.53	1.31	0.60
$\mathrm{P}\ (\mathrm{A}_{t-1} = \mathrm{H/M/L} \rightarrow \mathrm{B}_t = \mathrm{U/N/D})$	1.37	-0.40	-0.61	-1.18	-0.59	2.67	1.22	1.32	-1.06
$\mathrm{P}\ (\mathrm{C}_{t-1} = \mathrm{D/L/S} \rightarrow \mathrm{B}_t = \mathrm{U/N/D})$	0.60	0.70	-0.72	-0.94	0.01	1.62	1.13	0.16	-0.15

In order to predict the item 9 in this case, the weights were multiply by fuzzy set of the 8^{th} item (Eqn. (4.8)). Then the TW was calculated for variable B using Eqn. (4.9) as TW(U) = -1.31, TW(N) = 1.25, and TW(D) = 0.78. So the item 9 is predicted to “No Changed” with weight = 1.25 and “Down” with weight = 0.78.

4.6 Classification and Clustering

Once all the fuzzy temporal patterns have been discovered, for each MTS, a set of intra-FTP measures, $d^i_{j,\tau}$, (i = 1, 2, …, N, j = 1, 2, …, n, and τ = 1,2,… p_{ij}), and a set of inter-FTP measures, $d^i_{jj',\tau}$ (i = 1, 2, …, N, j,j' = 1, 2, …, n, j ≠j’ and τ = 1,2,… p_{ij}) associated with it are obtained. These intra-/inter-FTPs forms a feature vector as $\{d^i_{1,\tau}, d^i_{1\ 2,\tau}, d^i_{1\ 3,\tau}, \dots, d^i_{j\ j',\tau}, \dots d^i_{n-1\ n,\tau}\}$ for one MTS, where n is the total number of variables, $1 \le \tau \le \tau_{max}$ and i = 1, 2, …, N. In addition, each $d^i_{j\ j',\tau}$ may contain a set of values when $t^i_{j,t}$ equals to different discrete value. Then the traditional classifier, SVM, is used for classification.

A Support Vector Machine (SVM) transforms the original input data into a higher dimensional space using nonlinear mapping and then searches for a linear separating hyper-lane (Han, J. et al. 2007). For classification, it applies a kernel function to the original input data. There are three main kinds of kernel functions. In our case, we chose the Gaussian Radial Basis Function (RBF) Kernel.

Besides classification, clustering analysis can also be applied to feature vectors. The problem of clustering n sets of stocks into k clusters is transformed into the problem of clustering n features vectors into k clusters. The MTS within the same cluster is more similar to each other than to those outside the cluster. For example, the clustering algorithm, K-means, can be used to cluster these feature vectors.

4.7 Performance Evaluation

To evaluate the performance of FMTSC, we used both synthetic and real data sets. For the synthetic data set, we assumed that the performance of each stock was being monitored and for each stock, we generated a multiple time series consisting of three component time series corresponding to three data parameters that can best reflect the price performance of the stock. These three data parameters include (i) the difference between the daily maximum and minimum stock price (denoted as v_1), (ii) the percentage increase or decrease in the closing price between yesterday and today (denoted as v_2), and (iii) the trading volume (denoted as v_3). A total of 500 data values corresponding to data collected at 500 consecutive time instants were generated for each of v_1, v_2 and v_3. The data were generated in such a way that the patterns, as defined in Figure 4.4, were embedded into the data.

Using FMTSC, all the patterns defined in Figure 4.4 could be discovered successfully. For example, we could discover the significant inter- and intra-FTPs (shown in Table 4) where: (i) "$\underset{\tau=1}{\longrightarrow}$" denotes a fuzzy temporal association between the values observed on the left hand side and the values observed on the right hand side at one time instant later, and (ii) *w* denotes the weight of evidence. It should be noted that the weights of evidence are different for the different association relationships discovered. This is because the synthetic dataset was mostly generated randomly except for the patterns embedded in the data. As the relative proportions of different patterns in the data set are different, the weights of evidence are also different.

Patterns embedded in time series data

1. Assuming that the values of v_1,v_2, v_3 were all normalized. Their values were first randomly generated to take on any value within [0,1] with equal probability.

2. The patterns, if v_1 takes on a value within [0.07, 0.42] then v_2 takes on a value within [0.03, 0.39] in the next time instant, and if v_1 takes on a value within [0.62, 0.89], v_2 takes on a value within [0.24, 0.57] in the next time instant, are embedded into the data.

3. The patterns, if v_3 takes on a value within [0.83, 0.98] then v_2 takes on a value within [0.32, 0.61] in the next time instant, and if v_3 takes on a value within [0.28, 0.49], v_2 takes on a value within [0.71, 0.97] in the next time instant, are embedded into the data.

4. The patterns, if v_2 takes on a value within [0.31, 0.72] then v_2 takes on a value within [0.52, 0.89] in the next time instant, and if v_2 takes on a value within [0.59 ,0.93], v_2 takes on a value within [0, 0.39] in the next time instant, are embedded into the data.

Figure 4.4 Patterns Embedded in Synthetic Dataset

Table 4.4 Significant Associations Discovered in the Synthetic Dataset

Inter-FTA for v_1 and v_2	Inter-FTA for v_3 and v_2	Intra-FTA for v_2
$v_1 = H \underset{\tau=1}{\longrightarrow} v_2 = M$	$v_3 = H \underset{\tau=1}{\longrightarrow} v_2 = M$	$v_2 = H \underset{\tau=1}{\longrightarrow} v_2 = L$
(w=0.62)	(w=0.67)	(w=0.43)
$v_1 = L \underset{\tau=1}{\longrightarrow} v_2 = L$	$v_3 = M \underset{\tau=1}{\longrightarrow} v_2 = H$	$v_2 = M \underset{\tau=1}{\longrightarrow} v_2 = H$
(w=1.56)	(w=0.95)	(w=0.77)

In stock analyses involving daily closing prices, it is not usual for the data to contain missing values. However, for a stock analysis involving intra-day tick data, it is possible that, at the time of sampling, the data parameters or the price of a stock may not be available. In such a case, an unevenly sampled time series that appears to be broken will appear (see Figure 4.5). As the ability of the algorithm used for stock analysis to handle missing data is important, for FMTSC to be useful, it has to be very robust and has to be able to cope well even in the presence of stock data containing a high proportion of missing data.

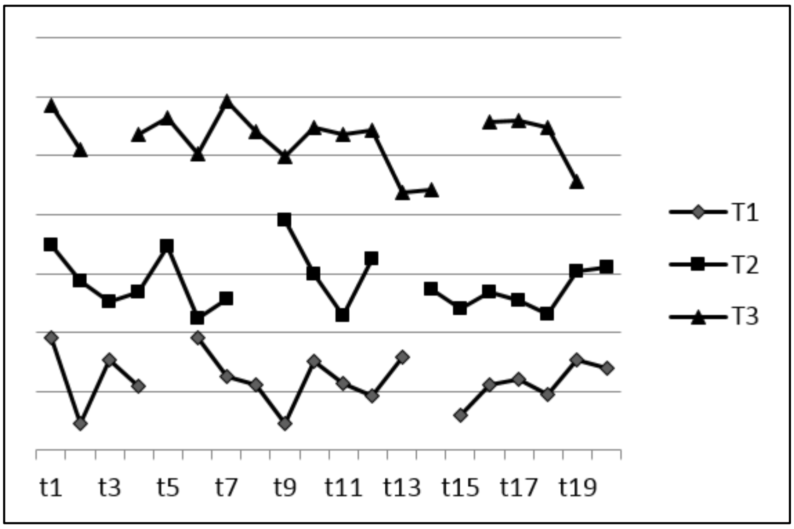

Figure 4.5 Example of Missing or Unevenly Sampling Data

To determine how well FMTSC can cope with such datasets, we performed several additional experiments. Specifically, we first removed 10%, and then gradually increased the figure by 5% each time up to 50%, of the data values randomly from the synthetic data set in a series of experiments to see if FMTSC could still discover the temporal patterns embedded originally in the data. For performance evaluation, we made use of the temporal patterns that FMTSC discovered in these data sets for prediction. We expected that, if FMTSC could discover most of the originally expected patterns, it would be able to make predictions with high accuracy in general.

For the purpose of the experiments, we used 80% of the data for training and the rest of the 20% for testing. The results, in terms of prediction accuracy, are shown in Figure 4.6. As we continue to remove more and more of the data values from 10% to 15% to 20% and then 25%, etc., until 50% of the data values are removed, we noted that the prediction accuracies are within the 70% to 75% range even when up to 30% of the data values were removed. The rate of decrease of the prediction accuracy increases as we data values were further removed beyond 30%. However, even with 50% of the data values removed, FMTSC is still able to predict with an accuracy of 55%. A close examination of the patterns shows that, even if there is a high percentage of missing values in the synthetic data set, FMTSC is still able to discover most of the intra- and inter-FTAs embedded in the synthetic data.

For benchmarking, we compared the performance of FMTSC with one of the most popular algorithm for association mining, the A priori algorithm (Agrawal, R. et al, 1994). By trial and error, we identified the best setting of the parameter values for the *minimum support* and *minimum confidence* to be 5% and 40% respectively for the algorithm to work on. Using the same training and testing datasets, we found, as shown in Figure 4.6 that the a priori algorithm was not able to perform as well as FMTSC; its prediction accuracy ranged from around 60% when 20% of the data values are removed, to around 50% when 25% of the data are removed. As the percentage of missing data is increased to 50%, the percentage accuracy dropped to below 30% which is no better than the prediction accuracy when one makes random guesses.

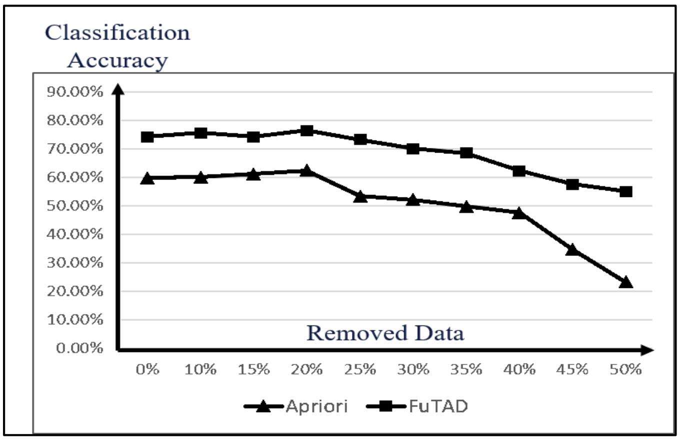

Figure 4.6 Comparison Results for Synthetic Data with Missing Values

Chapter 5.

Unsupervised Attribute Clustering Algorithm for Feature Selection

The curse of dimensionality refers to the problem that one faces when analyzing datasets with thousands or hundreds of thousands of attributes. In order to overcome the above challenges, we proposes an unsupervised feature selection method, called as Unsupervised Attribute Clustering Algorithm (UACA), for high-dimension dataset. The state-of-the-art similarity, MIC (Maximal Information Coefficient), is used to evaluate correlations between each pair of attributes. And then, the clustering algorithm, optimal K-modes, is used for attribute clustering.

5.1 Proposed Algorithm

Consider a dataset D containing a set of M-tuples of continuous data. Every tuple is described by N attributes, denoted as $A=\{A_1, \dots A_N\}$.

Definition 5-1. (Attribute Clustering) Attribute Clustering is a process which find c disjoint clusters, $C_1, \dots C_c$, of correlated attributes by assigning each attribute in $\{A_1, \dots A_N\}$ to one of these clusters. Formally, we define attribute clustering as a process that $\forall \mathrm{A}_i, i \in \{1, \dots N\}$, A_i is assigned to a $C_r, r \in \{1, \dots c\}$, where $C_r \cap C_s = \emptyset$ for all $s \in \{1, \dots c\} - \{r\}$.

The attributes within a cluster should have high correlation with or high interdependence to each other, and the attributes in different clusters are less correlated or more independent. The distance metric is used to measure the dissimilarity or distance between two objects in the most of clustering methods. Maximal Information Coefficient (MIC) is more meaningful if interdependent patterns are the most significant characteristics of a cluster reflecting the interrelationship among attributes.

5.1.1 Attribute Interdependence Measure

The basic elements required for mode finding and attribute clustering for continuous large scale data are the interdependence measures between attributes. *Maximal Information Coefficient (MIC)* (Reshef, D. N., 2011) is used because it can explore the large datasets and detect close associations between tens of thousands of attribute pairs in large datasets. Let $D'=\{A_i, A_j\}$ be the set of *n* ordered pairs of elements of x and y. The data space is partitioned in a X-by-Y grid, grouping the A_i and A_j values in X and Y bins respectively.

***Definition 5-2*. *(Maximal Information Coefficient*)** The Maximal Information Coefficient (MIC) (Reshef, D. N., 2011) that is a measure of dependence for two attributes relationship, A_i and A_j, can be denoted as $\mathrm{MIC}(A_i\!:A_j)$ and defined as Eqn. (5.1).

$$MIC(A_i\!:A_j) = \max_{XY<B(n)} M(D')_{X,Y} = \max_{XY<B(n)} \frac{I^*(D',X,Y)}{log(min\,X,Y)} \tag{5.1}$$

where $B(n) = n^{\alpha}$ is the search-grid size, $I^*(D', X, Y)$ is the maximum mutual information overall grids X-by-Y, of the distribution induced by D' on a grid having

X and Y bins where the probability mass on a cell of the grid is the fraction of points of D' falling in that cell.

RapidMic (Reshef, D. N., 2011) is used to calculate the value of MIC between two attributes. The method can effectively analyze large-scale datasets with a markedly reduced computing time through parallel processing (Reshef, D. N., 2011). When calculating a $m \times n$ matrix, the computational complexity of *RapidMic* for finding all pairs of attributes is $O(m * (m-1)/2)$.

The MIC is a measure of the strength of the linear or non-linear association between two variables *X* and *Y*, which means if MIC(A_i, A_j)>MIC(A_i, A_h) ($h \neq i \neq j$), the correlation between A_i and A_j is stronger than A_i and A_h. Besides, MIC(A_i, A_j) reflects the degree of deviation from independence between A_i and A_j. The value of MIC(A_i, A_j) is in the range of [0,1]. If MIC(A_i, A_j)=0, A_i and A_j are statistically independent; and If MIC(A_i, A_j)=1, A_i and A_j are strictly dependent.

In order to investigate the interdependency of an attribute with all the others within a cluster, we introduce the concept of *multiple MIC.*

Definition 5-3. (multiple MIC) The *multiple MIC* of an attribute $A_i$within an attribute group or cluster, $C = \{A_j | j = 1, \dots N\}$, is defined as Eqn. (5.2).

$$MMIC(A_i) = \sum_{i=1}^{N} MIC(A_i : A_j) \qquad (5.2)$$

where $MIC(A_i : A_j)$ is the MIC measure between A_i and A_j.

Based on the concept of $MMIC(A_i)$, the concept of the **mode** is introduced for an attribute cluster, which is defined as the attribute with the highest multiple interdependence redundancy in that cluster.

Definition 5-4 (mode) The mode of an attribute cluster $C = \{A_j | j = 1, \dots N\}$, denoted by A^c, is an attribute, say A_i, in that cluster such as Eqn. (5.3) shows.

$$MMIC(A_i) \geq MMIC\left(A_j\right), for\ all\ j \in \{1, \dots N\} \quad (5.3)$$

5.1.2 Attribute Clustering Algorithm

For a large dataset, clustering attributes into smaller groups not only significantly reduces the search dimension but also brings out various aspects of attribute correlation among those clusters. The basis of attribute clustering is to cluster the attribute set such that attributes within a cluster should have high interdependence whereas attributes in different clusters are less correlated using k-mode Attribute Clustering Algorithm (Reshef, D. N., 2011).

There are two phases in the proposed method of using MIC measure. Phase I is calculating MIC for each pair of attributes to construct similarity matrix, whereas Phase II is clustering attributes using optimal K-mode algorithm to find the most optimal *k* clusters among all attributes.

The K-Mode Attribute Clustering Algorithm is evolved from the k-means sample clustering algorithm where i) the attributes are treated as the samples, ii) the mode of an attribute cluster is treated as the sample mean, iii) the MIC matrix of the attributes is treated as the distance matrix of the samples. By assigning an integer *k*, K-Mode Attribute Clustering Algorithm can obtain k clusters optimizing the intra-cluster attribute interdependence. The best choice for *k* can be determined by the highest normalized sum of the *multiple MIC* denoted as *SMIC* obtained from clusters for each cluster configuration obtained.

In k-mode clustering process, the best choice for k is the k cluster configuration with highest SMIC, which defined in Eqn. (5.4).

$$SMIC(k) = \sum_{r=1}^{k} \sum_{A_i \in C_r} MIC(A_i : A^r) \tag{5.4}$$

i.e. for all $k \in \{2, \dots N\}$, we select k such as Eqn. (5.5).

$$k = arg\,max_{k \in \{2,\dots N\}} \sum_{r=1}^{k} \sum_{A_i \in C_r} MIC(A_i : A^r) \tag{5.5}$$

A pseudo code of the proposed algorithm (UACA) is given in Figure 5.1, in which *itr* is the pre-specified number of iteration for attribute clustering algorithm; *cnt* is the current iteration; *C* is an attribute cluster; A^c is the mode of *C*; *k* is the number of attribute cluster, and *SMIC(k)* is the sum of the multiple MIC of *k*. The output, *acSet* is a set of attribute clusters, and *optK* represents optimal cluster configuration. Finally, the modes of clusters are selected for classification.

```
Input: D = {A_1, A_2, ... A_N}(N is the number of attributes)
Output: acSet, optK

Procedure UACA(D)
Begin
   For each attribute pair, A_i and A_j
      /*calculate distance matrix for all attributes*/
      Calculate MIC_ij (equation (1));
      Store MIC_ij to MIC';
   End
   For k=2 to total number of attributes
      Randomly select k attributes to form initial acSet;
      cnt=1;
      Repeat
         For each attribute, A_i
            Assign A_i to C with the highest MIC to A^c
         End
         Update all A^c by calculating MMIC (equation(3))
         Update acSet;
         cnt++;
      Until all A^c does not change or itr=cnt;
      Calculate SMIC(k) using equation(4);
   End
   optK=k with highest SMIC(k) using equation(5)
   Return acSet and optK;
End
```

Figure 5.1 The pseudo code of unsupervised attribute clustering algorithm

To investigate the complexity of UACA, we consider a relational table composed of m samples such that each sample is characterized by n attribute values. The UACA algorithm requires $O(mn)$ operations to assign each attribute to a cluster. It then performs $O(mn^2)$ operations to compute the mode for each cluster. Let t be the number of iterations and k is the number of clusters. The computational complexity of the UACA is given by Eqn. (5.6).

$$O(UACA) = O(k(mn + mn^2)t) = O(kmn^2t) \tag{5.6}$$

5.2 Experimental Result on Synthetic Data Set

In order to explain the UACA algorithm and evaluate the performance of it, we first applied it to a synthetic dataset. Each record in the synthetic dataset is composed of 10 continuous attributes which denoted as A_1,……A_{10}, and is pre-classified into one of the 3 classes: C_1, C_2, and C_3. In the designed experiment, as shown in Figure 5.2, A_1 and A_2 are pre-defined as the mode of attributes and the data points lying on the rectangles, the circle, and the triangle belong to C_1, C_2, and C_3, respectively. Values of the other attributes (i.e. A_3,……A_{10}) in the tuple are randomly generated in the following manner:

- $\boldsymbol{A_3, A_4}$**:** if the value of A_1<0.5, the values are in range of (0, 0.5), otherwise, in the range of (0.5, 1).
- $\boldsymbol{A_5, A_6}$**:** if the value of $A_1 \geq 0.5$, the values are in range of (0, 0.5), otherwise, in the range of (0.5, 1).
- $\boldsymbol{A_7, A_8}$**:** if the value of A_2<0.5, the values are in range of (0, 0.5), otherwise, in the range of (0.5, 1).

- A_9, A_{10}**:** if the value of $A_2 \geq 0.5$, the values are in range of (0, 0.5), otherwise, in the range of (0.5, 1).

Obviously, $A_3,\ldots A_6$ are correlated with A_1 whereas $A_7,\ldots A_{10}$ are correlated with A_2. For an attribute clustering algorithm to be effective, it should be able to reveal such correlations. In this experiment, we generated 240 tuples (80 for each classes) in the synthetic dataset and added noises to the dataset by replacing the attribute values of $A_3,\ldots\ldots A_{10}$ in 20% of the types with random real number between 0 and 1.

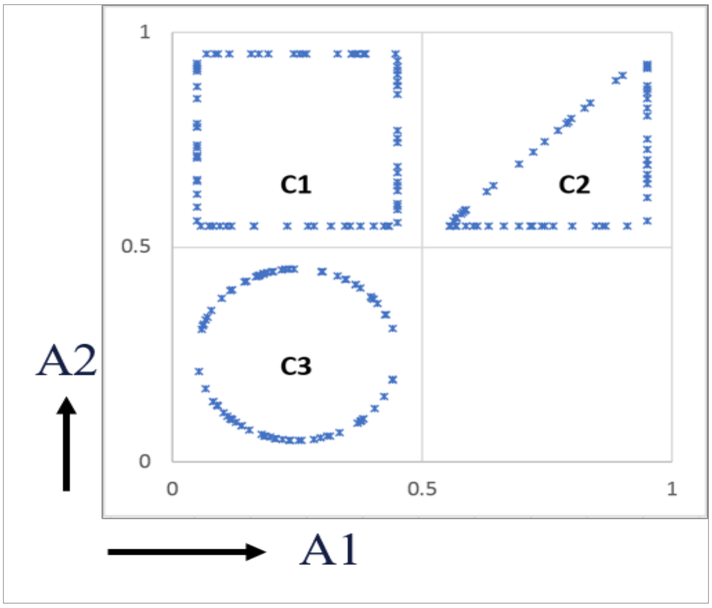

Figure 5.2 Attribute value of A_1 and A_2 in the tuples in the synthetic dataset

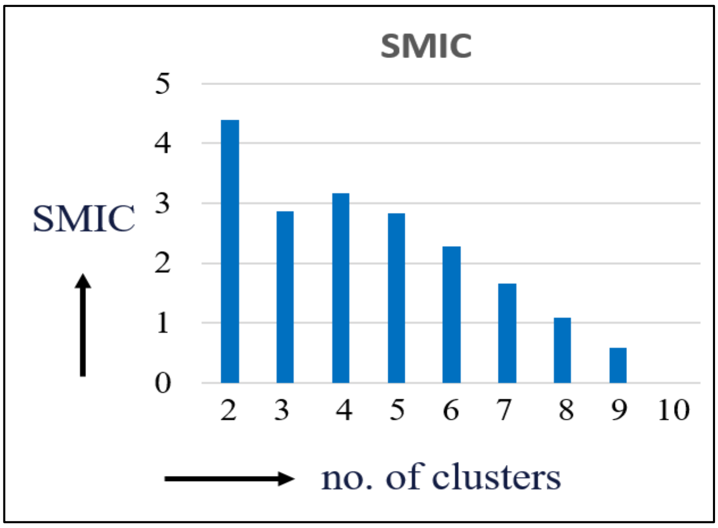

Figure 5.3 The value of SMIC over all the clusters found in the synthetic dataset

Firstly, we applied UACA to the synthetic dataset to find optimized clusters of attributes. Figure 5.3 shows the value of the ***normalized sum of the multiple MIC*** **(*SMIC*)** when different number of clusters is selected. It finds that the optimal number of clusters is 2 with the value of SMIC is 7.35. UACA identifies 2 clusters of attributes: $\{A_1, A_3, A_4, A_5, A_6\}$ *and* $\{A_2, A_7, A_8, A_9, A_{10}\}$, and A_1 is the *mode* of the former cluster whereas A_2 is the *mode* of the latter. It shows that UACA is able to reveal the correlations between the attributes hidden in the synthetic dataset.

5.3 Experimental Result on Real Datasets

We design appropriate experiments to verify the premises and reveal how realistic the proposed approach is when applied to various types of continuous data. Considerable works have been done in supervised feature selection on continuous data, but not many on unsupervised methods. Besides, although some attribute clustering methods (Zhao X. et al. 2013;Au, W.H. et al. 2005) have been applied into feature selection, the methods only focus on discrete data or distribute continuous data into discrete data. The experimental datasets are selected to illustrate the specific objective that our proposed method is able to achieve.

This section reports experiments using six benchmark datasets obtained from UCI repository (Asuncion, A. et al. 2007). Table 5.1 summaries the descriptions of datasets. The first two columns of Table 5.1 show the number of samples and attributes of the datasets, and the third columns shows the number of class labels.

To evaluate the performance of the proposed algorithm the comparisons of the classification accuracy between the proposed algorithm (UACA) and other methods using different classifiers are shown in this section. Because both *Wrapper* and *Embedded Feature Selection methods* utilize classification result to evaluate and refine feature subsets while choosing them, but classification accuracy is used as criteria for comparison, we only compare the proposed algorithm with two *Filter Feature Selection* methods - Information Gain and Relief (Kenji, K. et al. 1992). The comparison experiments are applied into different datasets when using i) all features, ii) the features are selected by Information Gain, iii) and the features are selected by Relief (Kenji, K. et al. 1992). The reason we compare the proposed algorithm with IG and Relief is that they are individual evaluations (Guyon, I. et al. 2002), which means a weight value could be assigned for each feature. Features could be ranked by this measure and then a specific number of attributes could be selected from the top of this ranking list.

Table 5.1 Descriptions of Datasets in the Experiment

Datasets	***Instances #***	***Features #***	***Classes #***
breast-w	683	9	2
ionosphere	351	34	2
sonar	208	60	2
spambase	4601	57	2
waveform	5000	40	3
wine	178	13	3

Several classifiers are used for testing the performance of different feature selection methods, i.e. Native Bayes (John, G.H. et al. 1995), KNN (Aha, D.W. et al. 1991), C5.0 (Webb G.I. et al. 1999), which are all popular classification algorithms and widely applied in various research fields. For the purpose of performance evaluation, 60% data are selected randomly as training data and the rest 40% as testing data.

Finally, the Classification Accuracy (CA) is used to evaluate how accurate UACA is and the CA can be defined as Eqn. (5.7).

$$CA = \frac{Total\ Number\ of\ testing\ data\ in\ the\ correct\ classes}{Total\ number\ of\ testing\ data} \tag{5.7}$$

5.3.1 Comparison Result Between With and Without UACA

In this section, we first provide the classification result with and without the proposed feature selection method. Table 5.2 summarizes the feature selection result when UACA algorithm is applied for all datasets. The table consists of three sections, *original features (OF)*, *selected features (SF)* and *pruning rate.* The first two columns of Table 5.2 show the number of original attributes and the number of attributes after selected by using UACA. Finally, the pruning rate in the tables is defined as: (*OF*-*SF*)/*OF*. In Table 5.2, the average of pruning rate is 87.16%. That is to say, after selecting features, the size of dataset is reduced effectively.

Table 5.2 Feature Selection Result Using UACA

Datasets	**Original Features #**	**Selected Features #**	**Pruning Rate**
breast-w	9	2	**77.8%**
ionosphere	34	3	**91.2%**
sonar	60	8	**86.6%**
spambase	57	3	**94.7%**
waveform	40	2	**95%**
wine	13	2	**84.6%**

On the other hand, it is also necessary to evaluate the performance of classification result for the data with feature selection. To show generality of our algorithm, classification performance are tested with several classical classifiers, i.e. Naive Bayes, KNN, and C5.0. Table 5.3 shows the classification result of three different classifiers when all features of dataset are used. And Table 5.4 shows the classification accuracy result of three classifiers when UACA is used as feature selection method. For dataset “breast-w”, “ionosphere”, “sonar” and “wine”, the classification accuracy with selected features by UACA is higher than the classification accuracy of original features, and for other datasets, the classification result after feature selection by UACA is comparable with the result of original features.

Table 5.3 Classification Result without Feature Selection

Datasets	**Bayes**	**KNN**	**C5.0**	**Average**
breast-w	89.72%	94.68%	92.55%	**92.32%**
ionosphere	76.43%	86.1%	85.25%	**82.59%**
sonar	43.21%	80.25%	77.78%	**67.08%**
spambase	73.22%	79.23%	81.75%	**78.07%**
waveform	60.3%	73.2%	74.31%	**69.27%**
wine	85.8%	97.1%	92.7%	**91.87%**

Table 5.4 Classification Result Using Selected Features by UACA

Datasets	**Bayes**	**KNN**	**C5.0**	**Average**
breast-w	93.97%	92.91%	92.55%	**92.55%**
ionosphere	81.44%	85.57%	82.47%	**83.16%**
sonar	67.92%	79.25%	69.81%	**72.33%**
spambase	63.66%	82.19%	82.85%	**76.23%**
waveform	62.43%	61.9%	67%	**63.78%**
wine	86.67%	97.78%	93.33%	**92.59%**

5.3.2 Comparison Result for Different Feature Selection Algorithms

We compared the proposed UACA with the other two feature selection methods, Information Gain and Relief (Webb G.I. et al. 1999), because these methods are individual evaluation (Guyon, I. et al. 2002). That is to say, a weight value is assigned for each feature as the measure and features could be ranked by the measure and then a specific number of attributers could be selected from the top of this ranking list. We also compared the algorithms using different classifiers. Table 5.5, Table 5.6 and Table 5.7 show the classification accuracy when different classifiers are used for the features selected data. The column 'UACA', 'IG' and 'Relief' represent the classification accuracy when using selected features by UACA, Information Gain, and Relief respectively.

Table 5.5 Bayes Classification Accuracy Using Different Feature Selection Methods

Bayes	**UACA**	**IG**	**Relief**
breast-w	93.97%	91.13%	90.43%
ionosphere	81.44%	75.69%	81.43%
sonar	67.92%	48.72%	69.23%
spambase	63.66%	62.93%	69.69%
waveform	62.43%	61.59%	60.26%
wine	86.67%	92.42%	71.21%
Average	**76.02%**	**72.08%**	**73.71%**

Table 5.6 KNN Classification Accuracy Using Different Feature Selection Methods

KNN	**UACA**	**IG**	**Relief**
breast-w	92.91%	92.2%	91.13%
ionosphere	85.57%	79.17%	85.51%
sonar	79.25%	75.64%	71.79%
spambase	82.19%	81.54%	66.32%
waveform	61.9%	56.69%	56.44%
wine	96.78%	93.94%	81.82%
Average	**83.10%**	**79.86%**	**75.50%**

Table 5.5 shows the Bayes classification accuracy for different feature selection methods. We can conclude that the average of classification accuracy is 76.02% for UACA which is higher than 72.08% with IG and 73.71% with Relief. Table 5.6

summarized the result for KNN classification accuracy. Similarity, the proposed algorithm can get the higher classification result with 83.10% when comparing with IG algorithm (79.86%) and Relief algorithm (75.50%). Finally, when the classification algorithm, C5.0, is used, the accuracy is 81.00% with UACA, which is still the highest accuracy result, comparing with 80.01% in IG and 75.53% in Relief.

Table 5.7 C5.0 Classification Accuracy Using Different Feature Selection Methods

C5.0	**UACA**	**IG**	**Relief**
breast-w	92.55%	92.2%	91.84%
ionosphere	82.47%	77.18%	84.48%
sonar	69.81%	76.92%	67.95%
spambase	82.85%	81.14%	71.19%
waveform	67%	61.69%	61.98%
wine	91.33%	90.91%	75.76%
Average	**81.00%**	**80.01%**	**75.53%**

Hence, comparing with traditional supervised feature selection method, the classification accuracy is the highest when UACA is applied as feature selection method.

To make the results more clear, we calculated and grouped the average of classification accuracy using different classifiers. The packed results are illustrated in Figure 5.4, representing the average of classification accuracy for all classifiers and each group stands for one dataset with three different feature selection methods. Four different colors bars in one group represent the classification accuracy for features when using i) without selection algorithm, ii) using proposed UACA algorithm, iii) using information Gain algorithm and iv) using Relief feature selection algorithm. We can conclude that when UACA is used, the performance of classification result is comparable, and even the best one for some cases.

In Summary, unlike many existing methods, the proposed unsupervised feature selection method can handle dataset without information of class label. As the proposed feature selection method is generic, it can perform its tasks without requiring any special assumption about data models. In addition, the UACA do not need to discretize continuous data into discrete data before attribute clustering. For performance evaluation, UACA was tested with synthetic datasets and six UCI datasets in real world. The classification results with different classifiers show that it is promising algorithm for feature selection. The classification result after feature selection by UACA even outperforms the classification result using whole dataset in some cases.

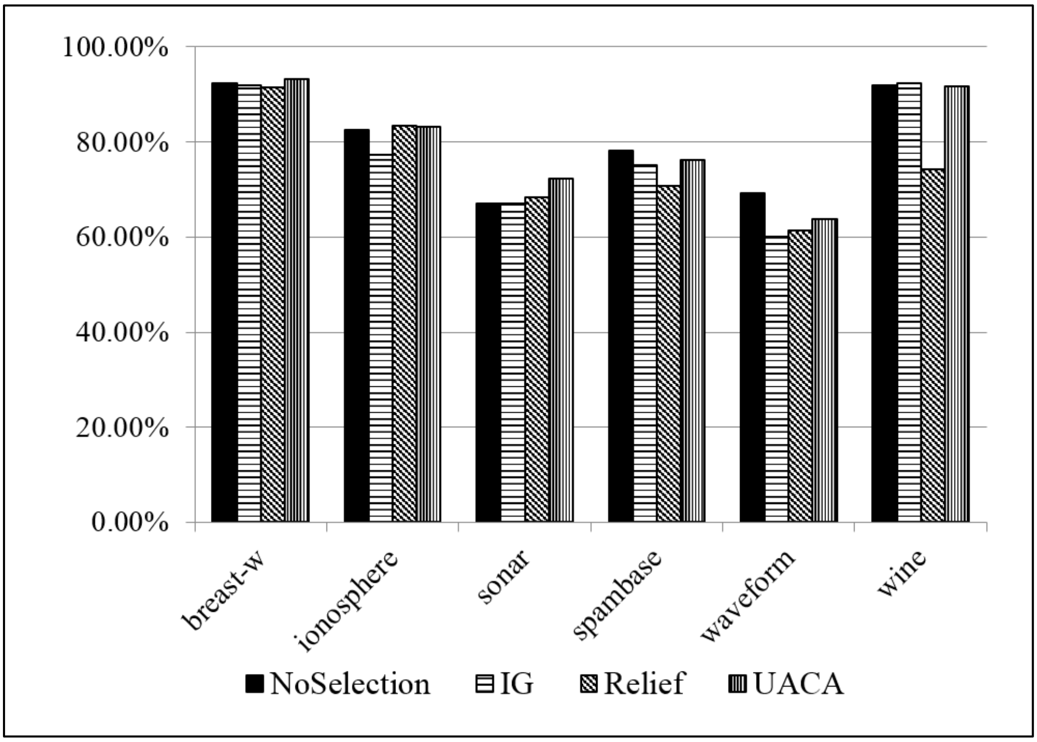

Figure 5.4 Average classification accuracy using different feature selection methods

Chapter 6.

Experimental Result in Real Cases

6.1 Pattern Analysis and Prediction in the Application of Stock Market

A stock market is well-known to be difficult to model and predict because it is chaotic and highly random in nature (Roy, P. et al. 2015). Analysts and investors have been relying on a range different mathematical techniques to do so (Nicholis, S.C. et al. 2011). More recently, they have started to use data mining or machine learning techniques, such as decision trees, artificial neural networks (Theodorus, Z. et al, 2013), support vector machines (Sands, T.M. et al. 2015), and genetic algorithms (Chandwani, D. et al. 2014) to analyse stock by finding correlations between stocks (Srisawat, A. 2011; Ting, J et al. 2006; Kumar, M. et al. 2011), classifiers that can differentiate between different stock categories (Ou, P. et al. 2009; Nair, B. B. et al. 2010; Nair, B. B. et al. 2011; Nair, B.B. et al. 2012), clusters of stocks that behave in similar ways (Cai, F. et al. 2012; Nanda, S.R. et al. 2010; Todd, W. 2002), trends that can be predicted (Kumar, M. et al. 2006; Nair, B. B. et al. 2011; Gavrilov, M. 2013), and anticipating stock prices on the following days (Ou, P. et al. 2009; Nair, B. B. et al. 2010; Nair, B.B. et al. 2011; Nair, B. B. et al. 2012; Ramoni, M. et al. 2000).

Given a set of such multivariate time series data, the discovery of inherent clusters can allow important information to be revealed. However, most of the existing algorithms have considered only historical prices of stock without taking into account the intra- and inter- relationship between different stock variables. The reason behind such choices is that the stock analysis problems that have been typically studied and formulated as a data analysis problem involve univariate time series made up of the data values of a single variable representing the daily closing prices of a stock collected over a period of time. Even though there have been some attempts to make use of intra-day data for stock forecasting (Royo, R.C. et al, 2015), the data concerned are still univariate time series involving a single variable.

However, for stock analysis to be more complete, other data parameters, such as the daily maximum and minimum, the bid and ask spread, the trading volume, etc., which are also collected usually for each stock on a daily basis, should also be considered. In other words, there is a need for an effective computational technique to be developed to ensure that important associations between different parameter values observed at different time instants can be taken into consideration during stock analysis. For example, if the differences between the maximum and minimum stock prices on a particular day indicate a significant rise or drop in the closing price of the stock on the following day, the uncovering of such associative relationships should allow us to better identify similar stocks and predict the corresponding stock trends. Since a univariate time series analysis technique may not be able to discover such relationships, we need a computational technique that can perform such tasks on multivariate stock time series. Specifically, such a technique must be able to uncover hidden temporal association relationships between the parameters observed for a stock at different time instants so that, based on the discovered relationship, we will be able to better predict prices, determine trends, select stocks, reduce risks, manage portfolios, etc. (Cai, F. et al. 2012). In addition, since stock data are typically very noisy, the association relationship are expected to be describable using only imprecise terminologies; the technique has to be able to appropriately model such relationship. Given such requirements, what is needed is a technique that can handle fuzzy multivariate time series data.

For performance evaluation, FMTSC has been tested using several sets of synthetic and real data, including those collected from the New York Stock Exchange (NYSE) and the Hong Kong Stock Exchange (HKSE). The results have shown that FMTSC can be a promising algorithm for more complete stock analysis. The results have also shown how the uncovering of hidden association patterns can allow us to determine whether average daily, weekly or bi-weekly stock prices are most predictable and whether the stock markets should best be predicted one day, two days, or several days, etc., ahead of time.

6.1.1 Previous Work about Association Analysis for Stock Market

In the area of data mining finance, the analysis of stock market involves the following three parts: Association Analysis, Classification and Clustering (Cai, F. et al. 2012). Association Analysis is a data mining technique that can discover interesting relationships, such as the association rules or sets of frequent item-sets, hidden in large datasets (Tan, P.-N. et al. 2006). Classification is a process of assigning objects to one of the predefined categories (Cai, F. et al. 2012). Cluster analysis groups similar data objects into clusters (Tan, P.-N. et al. 2006) when the classes or clusters are not defined in advance. For stock data, it is difficult to assign the class label for each stock, so clustering is more useful when considering stock data, and applying the clustering results for creating portfolios is the most significant application for stock analysis.

Association Analysis is used by data mining researchers to discover interesting association relationships between items or sets of items in a transactions database (Tan, P. et al. 2005). For the stock market, algorithms, such as an a priori algorithm (Agrawal, R. et al, 1994) developed for association analysis, have been used to discover inter-transaction associations for predicting stock prices (Lu, H. et al. 1998). Argiddi et al. (Argiddi R. V. and Apte S. S., 2012) note that such a priori algorithms have been used to mine stock transactions in the Indian IT stock market via association rules in the form of "If the price of stock A goes up then the price of stock B also goes up". Similarly, in (Srisawat, A. 2011), associations between individual stocks have also discovered using a priori algorithm. The same algorithm is also used in (Ting, J et al. 2006) to identify frequently occurring associations in stock time series and to find interrelationships between stock price movements of stock pairs. In (Kumar, M. et al. 2011), association analysis is used to find similarities between stocks traded on Indian stock markets.

Since an association rule mining algorithms, e.g., a priori algorithms, do not take into consideration sequential information. They have been modified in certain cases when such information has to be considered. For example, in (Todd, W. 2002), associations are discovered between the movements of different stocks constituting the Dow Jones Industrial Average. In (Hung C.F. et al. 2005), a two phase association mining algorithm, is modified from an a priori algorithm, to take into consideration the sequential associations used in mining in univariate time series.

Another variant of the apriori algorithm is that introduced in (Tung, A.K.H. et al. 2003) to mine special inter-transactional associations. There have also been some attempts to make use of association analyses for stock prediction. For example, in order to improve the quality of prediction, an algorithm is proposed in (Gavrilov, M. 2013) to decompose a time series into multiple components for examining trends, seasonality and other irregular components. Prediction is then made for each component separately.

In addition to the above, there have also been some attempts to perform association analysis using fuzzy approaches. For example, the Equal Frequency approach is used to determine the boundaries for the fuzzification of consecutive close-open values and a 14-day historic data is used for predicting the movement of the future 5-day (Roy, P. et al. 2015). In (Ijegwa, A.D. et al, 2014), a fuzzy approach is proposed for discovering the association rules from four technical indicators: the Moving Average Convergence or Divergence, the Relative Strength Index, the Stochastic Oscillator and the On-Balance Volume.

Unlike traditional time series analysis techniques which are mainly developed to handle univariate time series data, the association discovering algorithms can allow inter-transaction data to be discovered and can handle multivariate time series data to some extent. Their main limitations arises in their handling of noisy and missing data as they are not developed originally with stock analysis in mind. In order for the complexity in stock analysis to be effectively tackled, what we need is a technique that can handle multivariate time series data in the presence of irrelevant, noisy, inconsistent and missing data. The technique should be able to capture important inter- and intra-time series temporal association relationships and perform temporal abstractions that may be vague and imprecise. It should also be able to disregard irrelevant noise and identify interesting patterns.

6.1.2 Datasets Description

A number of additional experiments are performed using real stock datasets obtained from the NYSE and the HKSE respectively. The datasets were used to test how well FMTSC is able to perform the tasks of association analysis and if there is any general trend that we may be able to discover in the behaviors of the NYSE and the HKSE stocks. As for the experiments with the real datasets, the use of FMTSC to discover how the settings of the time periods, such as average daily, half-weekly, weekly or bi-weekly, etc., for which predictions are made should be set and how the settings of the numbers of days ahead of predictions are made, etc., investigated. In addition, we also compared the performance of FMTSC with linear regression, the crisp and fuzzy versions of the most popular algorithm for association analysis, the a priori algorithm (Agrawal, R. et al, 1994) and a fuzzy a priori algorithm (Jafarzadehm, H. et al, 2014).

In addition, in order to evaluate the clustering results, two evaluation measurements are used i) classification accuracy, which can reflect the quality of the discovered clusters (Roy, P. et al. 2015) and ii) creating portfolios using different clustering result and the weekly returns from stocks.

One of the datasets we had used in our experiments was obtained from the Yahoo finance Website (Yahoo Financial). In total, the data for 65 stocks listed with respect to the "Dow Jones Composite Average" were collected from New York Stock Exchange (NYSE). The Dow Jones Composite Average, is computed from stocks that make up the Dow Jones Industrial Average, the Dow Jones Transportation Average, and the Dow Jones Utility Average. Each of the 65 stocks were classified into one of eight standard industrial categories: Basic Materials, Consumer Goods, Financial, Healthcare, Industrial Goods, Services, Technology and Utilities as given by (Yahoo Financial). For each of the 65 stocks, a total of 1399 days of data from January, 2010 to April, 2015 were collected under the five parameters of Open, High, Low, Close, and Volume for our experiments.

The other dataset we had used in our experiments was collected from 50 blue chip stocks comprising the Hang Seng Index (HSI) of the Hong Kong Stock Exchange (HKSE). Each of these 50 stocks is classified into one of six categories which include Basic Materials, Consumer Goods, Financial, Services, Technology and Utilities as given by (Yahoo Financial). The lengths of the time series corresponding to each stock are not the same since the companies were not all listed in the Hong Kong Stock Exchange at the same time. For example, for one of the stocks, the earliest date for which we had data was in June, 2004. Hence, we had a shorter multivariate time series for this stock; for other stocks, the time series obtained were of different lengths. The longest multivariate stock time series consisted of a total of 4,110 days covering the period from January, 2000 to October, 2015; it was made up of data corresponding to five parameters including Open, High, Low, Close, and Volume.

6.1.3 Comparing with Association Analysis Algorithms

Considering the application of algorithm to stock market analysis, the price fluctuation ratio is used to represent such parameters as: i) the movement of stock price (*MVT*), ii) the differences between the maximum and minimum stock price (*DIFF*), and iii) the trading volume of a stock (*TVL*). $Value_t(DIFF) = \frac{H_t - L_t}{L_t}$, $Value_t(MVT) = \frac{C_t - O_t}{C_t}$, and

$Value_t(TVL)$ = $\frac{V_t - V_{t-1}}{V_{t-1}}$ (where H_t represents the highest price on the *t*th day; L_t represents the lowest price on the *t*th day; C_t represents the close price on the *t*th day; O_t represents the open price on the *t*th day; V_t represents the total trading volume on the *t*th day) are defined such that fuzzy linguistic terms can be used to represent the values that different parameters can take on.

In the case of discretization, the values of different variables of stock can be defined by a finite number of different values, such as “Up”, “No Change” and “Down” for the variable *MVT*.

With the above datasets, we performed several benchmarking experiments aiming to compare FMTSC with i) linear regression, ii) the a priori algorithm (Agrawal, R. et al, 1994), and iii) the fuzzy Apriori algorithm (Jafarzadehm, H. et al, 2014).

As for the experiments with the NYSE dataset, we used the data collected from the beginning of 2010 to the end of 2014 for training. FMTSC and the above algorithms were used to discover association relationships in the training data. The patterns discovered were then used for prediction using the 2015 testing data. For the experiments using the HKSE dataset, the time series were all of different lengths, so we used the first 70% of the data for training and the remaining 30% for testing.

To ensure fairness in the comparison, we used the same methods to pre-process the original data sets. Firstly, all values of the data sets were discretized into three levels. For the linear regression method, we used “1”, “2”, “3” to represent different levels. And all parameter values were used as input to predict the price of stocks in the next time point. For experimentation, when discovering intra- and inter-FTAs, we only look back one time instant. In other words, FuTAD considered only temporal associations between the stock parameters of two consecutive days (i.e., τ is set to 1). Tables 6.1(a) and 6.1(b) show the results for the two sets of data respectively.

Table 6.1(a) Prediction Results for the NYSE Data Set

Prediction Accuracy	**Regression**	**Apriori**			**Fuzzy Apriori**	**FMTSC**
		sup=5%, con=30%	**sup=10%, con=40%**	**sup=10%, con=50%**		
Basic Materials	33.67%	46.52%	36.90%	20.42%	52.09%	53.26%
Consumer Goods	44.97%	51.33%	51.33%	24.79%	54.45%	56.28%
Financial	18.37%	47.86%	45.91%	19.45%	46.67%	53.38%
Healthcare	55.73%	47.86%	38.50%	17.99%	54.04%	55.35%
Industrial Good	43.27%	44.38%	44.38%	20.90%	49.59%	60.05%
Services	41.33%	48.13%	43.32%	16.05%	59.80%	71.49%
Technology	40.11%	43.31%	39.04%	22.35%	48.86%	68.08%
Utilities	42.29%	48.13%	36.82%	22.36%	61.40%	62.15%
Average	**39.97%**	**47.19%**	**42.03%**	**20.54%**	**53.36%**	**60.01%**

From the tables, we note that linear regression does not perform too well when it comes to stock prediction. The average prediction accuracy for both datasets of around 39% is only slightly better than random prediction. It is likely that this relatively poor performance is due to the invalidity of the underlying assumptions that need to be made for linear regression to use. The noise in the data cannot probably be assumed to be identically and independently distributed and the distributions are not likely to be Gaussian and linear. These experimental results seem to confirm the impression that the stock market cannot be best predicted using traditional statistical methods (Todd, W. 2002).

For experiments involving the a priori algorithm, the user-defined parameters of *minimum support* and *minimum confidence* depend on the tasks performed and are set as 5% and 30%, respectively, for the NYSE dataset. These settings are found to be able to produce higher prediction accuracies than those attainable by other settings.

Under these settings, the a priori algorithm has performed with a prediction accuracy of 47.19% (see Table 6.1(b)). When we set the *minimum support* to 10% and the *minimum confidence* to 40%, the prediction accuracy is decreased slightly to 42.03%. If the minimum confidence is set to 50%, average value prediction accuracy drops to 20.54% even if the *minimum support* is kept at 10%. The a priori algorithm can be very sensitive to noise in the data as a small change in parameter settings can result in rather big differences in prediction accuracy.

Table 6.2(b) Prediction Result for the HKSE Data Set

Prediction Accuracy	Regression	Apriori			Fuzzy Apriori	FMTSC
		sup=5%, con=20%	sup=10%, con=40%	sup=10%, con=50%		
Basic Materials	33.20%	51.64%	51.64%	22.71%	68.76%	86.72%
Consumer Goods	42.59%	51.04%	42.09%	21.28%	64.80%	89.75%
Financial	34.18%	52.77%	46.74%	22.12%	66.96%	75.63%
Services	42.11%	47.76%	45.79%	20.46%	68.31%	76.90%
Technology	41.16%	50.57%	45.87%	22.39%	67.11%	65.88%
Utilities	44.81%	52.42%	46.86%	21.46%	67.76%	69.58%
Average	**39.68%**	**51.03%**	**46.50%**	**21.74%**	**67.28%**	**77.41%**

Similarly, for the HKSE dataset, the a priori algorithm performs better when the *minimum support* is set to 5% and the *minimum confidence* to 20%. Under such settings, the average prediction accuracy reaches 51.03%. When we set the *minimum support* to 10% and the *minimum confidence* to 40%, the prediction accuracy is decreased slightly to 46.5%. When the *minimum support* is maintained but the *minimum confidence* is increased to 50%, the average prediction accuracy drops rather drastically from 46.5% to 21.7%.

Again, while the prediction accuracy can be optimized by setting different support and confidence, it should be noted that the a priori algorithm may not be a robust algorithm, because a small change in parameters can result in a very big difference in prediction accuracy.

The relatively poor performance of the a priori algorithm can be argued to be due to its ignoring patterns or associations that are vague and imprecise. In other words, using a fuzzy a priori algorithm may be more appropriate when performing association analysis. It is for this reason that, in our experiments, we also compared FMTSC with a fuzzy a priori algorithm (Jafarzadehm, H. et al, 2014). Using the same fuzzy set definitions as with FMTSC, the fuzzy support was set equal to the sum of the minimal membership degrees, i.e., *fuzzysup*($A \rightarrow B$) = $\sum_{i=1}^{n} \min(f_A(x), f_B(x))$, where $A \rightarrow B$ indicates the associative relationship between two parameter values, *A* and *B*. In the same way, the fuzzy confidence can be determined as *fuzzycon*($A \rightarrow B$) = $\frac{\sum fuzzysup(A \rightarrow B)}{\sum fuzzysup(A)}$ (Jafarzadehm, H. et al, 2014).

In addition, the threshold of *minimum fuzzy support* and *minimum fuzzy confidence* is automatically determined on the basis of statistical information concerning Average Middle, Variance, Standard Deviation (Jafarzadehm, H. et al, 2014). In this research, the same approach to determining the thresholds of minimum support and confidence are adopted; the results shown are listed in Tables 6.1(a) and 6.1(b).

From Table 6.1(a), the average prediction accuracy resulting from the fuzzy a priori algorithm for the NYSE dataset has increased from the best value of 47.19% in the case of the crisp algorithm to 53.36% in the case of the fuzzy algorithm.

As for the HKSE dataset, there is also a significant increase in prediction accuracy from 51.03% in the case of the crisp a priori algorithm to 67.28% in the case of the fuzzy a priori algorithm. This confirms our belief that the use of fuzzy set concepts in the modeling of stock associations does indeed allow stock trends and movements to be better analyzed.

The main disadvantage with the fuzzy a priori algorithm is again with the setting of the threshold of fuzzy support and fuzzy confidence. Different settings of minimal support and minimal confidence may greatly affect the prediction result.

Besides the above tradionnal association analysis methods, we also compared FuTAD with popular algorithms such as (i) the Decision Tree algorithm (Al-Radaideh, Q. A. et al. 2013), (ii) the Artificial Neural Network (ANN) (Devadoss A.V. et al. 2013), (iii) the Support Vector Machine (SVM) (Wang Y. et al. 2013) and (iv) the dual-factor modified fuzzy time-series model (DF-Fuzzy) (Chu H.H., et al. 2009).

For experimentation, as described in (Al-Radaideh, Q. A. et al. 2013), the parameters of open, close, high, and low of the stocks were used to characterize a stock and were discretized into three levels -- "positive", "negative" and "equal". Given the time series of these parameters, FuTAD was used to detect for intra- and inter-FTA patterns and used to predict of stock prices of the following time instant. The prediction accuracy of FuTAD was then compared against that of the use of decision tree classifiers for the same task was applied for prediction. Their experimental results show that the performance of the decision tree classifier C4.5 is better than that of another decision tree classifier of ID3.

In addition to decision tree, ANN (Devadoss A.V. et al. 2013) was also used iin the same experiments to predict closing prices. An ANN was used which consisted of an input layer with input corresponding to open, high, low, close, and volum. One hidden layer and an output layer was used for this purpose. In order to ensure fairness, the movement of stock prices in the next time instant was used as output. The multilayer perceptron (MLP) algorithm was used to train the ANN for the prediction task.

In addition to the ANN, we also compared FuTAD with the SVM classifier. The same input and output as with the ANN classifier were used and the same parameter setting as stated in (Wang Y. et al. 2013) (parameter C=100 with RBF kernel function) was used.

Besides the above non-fuzzy methods, we also compared FuTAD with fuzzy association algorithns, especially the one known as the dual-factor modified fuzzy time-series model (DF-Fuzzy) (Chu H.H., et al. 2009). Again, we attempted to predict stock movement with these algorithms. The stock movement predicted for the next time instant was determined using the weighted fuzzy associations discovered from the time-series data (Cheng, C. H. et al. 2006) and based on the three linguistic terms of stock price. Finally, the stock movement at the time point t+1 was predicted using the proposed formula in (Chu H.H., et al. 2009). Table 6.2 (a) and Table 6.2(b) shows the comparison results with traditional methods for the two sets of data respectively.

From the above table, the "non-fuzzy" algorithms do not seem to perform as well as those that make use of fuzzy concenpts in the modeling of patterns hidden in the time series data. The average prediction accuracy of these non-fuzzy algorithms are only around 30% to 40%. When fuzzy modeling is considered, the average prediction accuracy can be much improved to 53.83% and 64.83% for the two data sets with the DF-Fuzzy algorithm. With FuTAD, the average prediction accuracy can even go as high as 85.99%.

Table 6.2 (a) Comparison Result for the NYSE Data Set

Prediction Accuracy	**Decision Tree C4.5**	**ANN**	**SVM**	**DF-Fuzzy**	**FuTAD With *EFD***	**FuTAD with *CLU.***
Basic Materials	36.06%	34.62%	34.02%	54.12%	53.26%	63.21%
Consumer Goods	46.1%	60.20%	63.91%	52.81%	56.28%	71.25%
Financial	35.59%	37.57%	36.69%	49.25%	53.38%	70.11%
Healthcare	36.81%	38.46%	39.05%	55.06%	55.35%	60.53%
Industrial Good	31.97%	31.66%	31.36%	51.05%	60.05%	67.15%
Services	37.71%	37.87%	36.98%	55.99%	71.49%	81.25%
Technology	34.57%	35.50%	35.21%	51.31%	68.08%	68.21%
Utilities	52.04%	40.24%	43.21%	61.24%	62.15%	58.27%
Average	**38.86%**	**39.52%**	**40.05%**	**53.83%**	**60.01%**	**67.05%**

Table 6.2 (b) Comparison Result for the HKSE Data Set

Prediction Accuracy	**Decision Tree C4.5**	**ANN**	**SVM**	**DF-Fuzzy**	**FuTAD With *EFD***	**FuTAD with *CLU.***
Basic Materials	42.24%	37.85%	37.15%	67.54%	86.72%	91.25%
Consumer Goods	43.68%	36.33%	38.75%	63.26%	89.75%	94.38%
Financial	39.29%	36.55%	35.67%	71.45%	75.63%	85.12%
Services	41.03%	32.95%	34.24%	61.44%	76.90%	81.35%
Technology	50.45%	40.61%	41.09%	62.93%	65.88%	83.59%
Utilities	41.59%	37.21%	39.03%	62.33%	69.58%	80.25%
Average	**43.05%**	**36.92%**	**37.66%**	**64.83%**	**77.41%**	**85.99%**

In Figure 6.1, we summarize the results of the performance of the different algorithms based on their prediction accuracies for the two data sets of NYSE and HKSE. Based on these results, FuTAD seems to perform consistently better than the other popular algorithms including that of the fuzzy Apriori algorithms. In fact, FuTAD dose not only find interesting fuzzy temporal associations that can predict better on average, the patterns that it found allows predictions to be made accurately with all categories of industries in both the NYSE and HKSE.

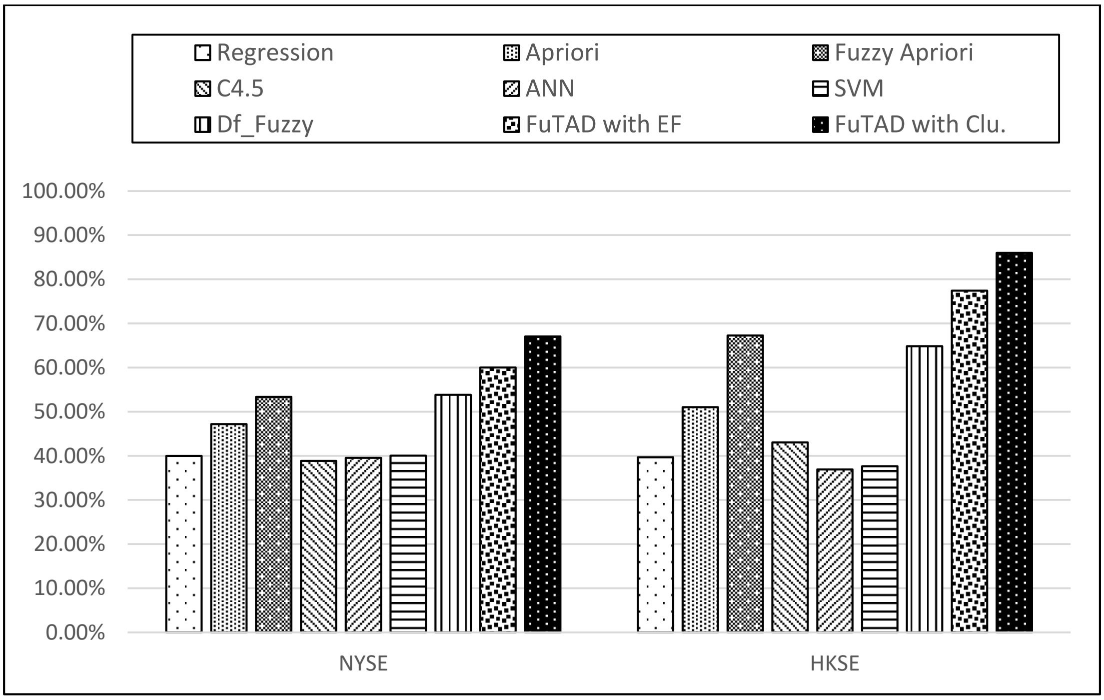

Figure 6.1 Performances of Different Algorithms on NYSE and HKSE data

6.1.4 Analysis for Stock Market

Having noted that FMTSC is a relatively effective approach for stock analysis, we used it to find answers to some of the questions that many people may have when it comes to stock prediction. Given that many people believe in the random walk hypothesis – a financial theory stating that stock market prices evolve according to a random walk and thus cannot be predicted (Godfrey M.D. et al. 1964), one such question is whether or not the stock market movements can be predicted at all (as argued to be consistent with the efficient-market hypothesis). While there are supporters of the random walk hypothesis, there are also people who believe that the stock market is predictable to some degree. They believe that prices may move in bulk trends and therefore the past price movements can be used to forecast the future. Using FMTSC, we investigated this controversy. If the stock market can be shown to be predictable to some degree, then the question that many people often have is whether or not it is better off predicting the average daily or average semi-weekly or average weekly or bi-weekly prices.

In addition, the problem of efficient frontier can be solved more effectively by clustering the stocks and then choosing to enhance the criteria of diversification. Therefore, FMTSC can be incorporated into the application of portfolio management using clustering results. We now present the results of our investigations.

6.1.4.1 Random Walk or Non-Random Walk?

To answer the above question, we consider the NYSE data first. Using FMTSC, fuzzy temporal association relationships were discovered in the multivariate time series corresponding to the stocks that belong to the eight industrial categories of Basic Materials, Consumer Goods, Financial, Healthcare, Industrial Goods, Services, Technology and Utilities in the dataset. For the purpose of finding out whether or not there are patterns in the data, we used the data from 2010-2014 for training. Using the data for 2015 for testing, if prediction of the change of stock price can be made accurately, we can conclude that patterns in the form of fuzzy temporal associations do exist in stocks.

For those people who do not believe that the stock market is predictable may have attempted to discover patterns in stock data so as to predict stock price one day ahead of time. Using FMTSC, we tried to determine if there are “daily” intra- and inter-FTAs between stock parameters of the current day with the previous day.

With the dataset and the fuzzy temporal associations discovered in them, stock price data in the testing set were predicted one day ahead of time. From the results shown in Table 6.3, we see that the average accuracy can go as high as 60% which is significantly higher than the 33%, if prediction is made through totally random guesses. The differences in prediction accuracy between stock prices in different industries do not seem to be very significant. We can conclude that there are indeed hidden regularities in the data in the form of intra- and inter-FTAs for relatively accurate predictions.

Table 6.3 Prediction Result for Stocks from the NYSE Using Different Time Periods

Prediction Accuracy	1 Day	3 Days	week
Basic Materials	53.26%	57.28%	**69.45%**
Consumer Goods	56.28%	**74.37%**	62.31%
Financial	53.38%	57.28%	**75.37%**
Healthcare	55.35%	76.38%	**88.44%**
Industrial Good	60.05%	66.67%	**82.50%**
Services	71.49%	70.37%	**75.48%**
Technology	68.08%	**74.07%**	68.75%
Utilities	62.15%	70.37%	**87.50%**
Average	**60.01%**	**68.35%**	**76.23%**

If discovering daily associations does not give a high enough accuracy for one to risk investing money based on FMTSC's prediction, one may wonder if FMTSC may be able to predict a semi-weekly three-day or a weekly five-day average. If the patterns are not strong enough to convince us that stock markets do not follow a random walk model, then one may wonder if it is possible that there may be more convincing patterns for semi-weekly or weekly prediction.

So, in our next set of experiments, we let FMTSC discover "three-day" semi-weekly and five-day "weekly" intra- and inter-FTAs. Based on the fuzzy temporal associations discovered in the training data, the testing data were used to determine the prediction accuracies; the results are shown in Table 6.3.

The results obtained are very interesting; they might actually explain why the random walk hypothesis was supported in some cases but not in some other cases. From Table 6.3, it is noted that, while discovering daily patterns may not allow daily closing prices to be predicted very accurately, whereas semi-weekly, three-day averages or weekly, five-day averages, can be predicted much more accurately. What is interesting is that, as prediction is made over a longer term, such as the prediction of weekly averages, the patterns discovered by FMTSC can be as accurate as 76.23% on average. In fact, this is the case for stocks in almost all industries except for Consumer Goods and Technology.

For stocks in these industries, the semi-weekly, three-day averages could be predicted more accurately than the weekly averages. This discovery can be important as financial experts generally look at weekly prices when judging the long-term trends of stocks (Todd, W. 2002) and this may be interpreted as evidence to support why the experts prefer to use weekly average prices rather than focusing on daily fluctuations.

In order to facilitate a better understanding of why predicting weekly averages can be more accurate, we present some of the patterns that FMTSC had discovered for each industrial category in Table 6.4 below. We show how the Movement (*MVT*) (which can take the values of *Up* (*U*), *Down* (*D*) and *No Change* (*NC*)) may be affected by the movement of the previous period and the differences between the maximum and minimum stock price of a stock (*DIFF*) (which can take the values of *High* (*H*), *Low* (*L*), and *Same* (*S*)), and its trading volume (*TVL*) (which can take the values of *Large*

(*L*), *Medium* (*M*) and *Small* (*S*)) of the last time period. In this experiment, the time period is set as *weekly*.

Table 6.4 The Most Significant Factors Affecting Movement in the NYSE

Category	Company	Significant Factor for Associations of Movement		
		Up	**No_Change**	**Down**
Basic Materials	**Chevron Corporation**	DIFF=L (W=0.29, C=0.38)	TVL=M (W=0.92, C=0.12)	MVT=D (W=0.38, C=0.51)
	Exxon Mobil Corporation	DIFF=L (W=0.42, C=0.44)	TVL =M (W=0.92, C=0.13)	MVT=D (W=0.37, C=53)
Consumer Goods	**Apple Inc.**	TVL =S (W=0.72, C=0.83)	TVL =M (W=0.84, C=0.37)	MVT=D (W=0.76, C=0.58)
	The Coca-Cola Company	TVL =S (W=0.31, C=0.41)	TVL =M (W=0.7, C=0.12)	MVT=D (W=0.82, C=0.55)
Financial	**The Goldman Sachs Group, Inc.**	DIFF=S (W=0.6, C=0.46)	TVL =M (W=0.66, C=0.09)	TVL =M (W=0.35, C=0.25)
	JPMorgan Chase & Co.	DIFF=S (W=0.51, C=0.43)	TVL =M (W=0.72, C=0.13)	TVL =M (W=0.55, C=0.3)
Healthcare	**Johnson & Johnson**	TVL =S (W=0.16, C=0.7)	TVL =M (W=0.66, C=0.11)	MVT=D (W=0.22, C=0.4)
	Merck & Co. Inc.	TVL =S (W=0.78, C=0.8)	TVL =M (W=0.79, C=0.13)	TVL=L (W=0.45, C=0.29)

Table 6.4 lists the most significant factors that have the strongest associations with the movement of the stock price, *MVT*, and hence are the most important factors signifying movement of the stock in the next period. Since such factors vary from industry to industry, they are listed by industrial categories. The first column in the table shows the industrial categories and the companies that belong to the category along with the most important factors (the highest *Weight-of-Evidence*, *w*, and *Confidence*, *c*) affecting whether the stock moves up, down or remains unchanged in the next period.

We can conclude from Table 6.4 that the most significant associations for stocks in the same categories are usually the same, whereas those that belong to different categories are usually different. Also, for almost all stocks, when the volume is at the medium level, as expected, the movements of the stock price are usually steady.

The experiments with the NYSE dataset were repeated with the HKSE dataset. FMTSC was used again to discover "daily", "semi-weekly (three-day)" and "weekly (five-day)" intra- and inter-FTAs. Based on the fuzzy temporal associations discovered in the training data, the testing data were used to determine the prediction accuracies and the results are shown in Table 6.5.

Table 6.5 Prediction Results for Stocks from HKSE Using Different Time Periods

Prediction Accuracy	**1 Day**	**3 Days**	**Week**
Basic Materials	**86.72%**	73.61%	76.13%
Consumer Goods	**89.75%**	87.73%	79.16%
Financial	**75.63%**	74.62%	57.48%
Services	**76.90%**	68.75%	59.07%
Technology	65.88%	**77.41%**	**77.41%**
Utilities	**69.58%**	63.15%	63.66%
Average	**77.41%**	**74.21%**	**68.82%**

Very interestingly, the prediction results obtained from experiments with the HKSE dataset were quite different from the NYSE dataset. While the results confirm that there are patterns in stock data, the stocks in the HKSE seem to be more predictable. The prediction accuracies that could be achieved were much higher. Furthermore, it is noted from Table 6.5, that the closing price movements could actually be predicted very accurately with the daily fuzzy temporal association discovered in the data when compared to that of the three-day and weekly averages.

In fact, on average, the daily prediction accuracy was as high as 89.75% for the industrial categories of Consumer Goods. The three-day and weekly averages could only be predicted with averages of 74.21% and 68.82% respectively. In other words, the HKSE stocks were more predictable in the short-term. We can also conclude, based on these results, that the prediction accuracy of a stock can be affected by the discovered patterns when different time periods are set for different categories and different markets of stock exchange. However, the degree to which they are affected seems to be relatively small when compared with the stocks in the NYSE.

To better understand the patterns discovered, we present some of patterns that FMTSC had discovered for each industrial category, when the time period was set to *"daily",* in Table 6.6. Using a similar analysis as with the NYSE data, we can see from the table that the most significant factors that affect the stock movements have been listed. The first column shows the industrial categories that the companies in the second column belong to. The third column shows the most significant factors with the highest *Weight-of-Evidence* (w) and *Confidence* (c), that affect whether the stock moves up, down or remains unchanged in the next period. The *Confidence* (c) of association between $T\left(L_j^i\right)_{t-\tau}$ and $T\left(L_j^i\right)_t$ can be best modeled with a confidence measure defined in terms of $Pr\left(t_{jk} \middle| t_{j'k'}\right)$.

We can conclude from the results that the most important factors are similar when the stocks belong to the same industrial category and the associations of stocks are different when the stocks belong to different industrial categories. For almost all stocks, when the trading volume is at a medium level, as expected, the movement of the stock price is relatively steady. And for most stocks, when the values of the differences between maximum and minimum stock price are about the same in the previous day, usually, the values of the movement of stock prices are equally *Down* the next day.

Table 6.6 The Most Significant Factors Affecting price movements in HKSE

Category	Stocks	Significant Factor for Associations of Movement		
		Up	**No_Change**	**Down**
Basic Materials	**0883.HK**	TVL=S(w=0.26, c=0.45)	TVL =M(w=0.38,c=0.36)	DIFF =S(w=0.46,c=0.39)
	1088.HK	TVL=S(w=0.13, c=0.41)	TVL =M (w=0.38,c=0.33)	DIFF =S(w=0.21,c=0.38)
Consumer Goods	**0151.HK**	MVT=U (w=0.25, c=0.44)	TVL =M (w=0.36,c=0.34)	DIFF =S(w=0.20,c=0.39)
	2319.HK	MVT=NC(w=0.19, c=0.33)	TVL =M (w=0.3,c=0.32)	DIFF =S(w=0.28,c=0.37)
Financial	**0939.HK**	MVT=U (w=0.12, c=0.28)	TVL =M (w=0.59,c=0.35)	DIFF =S(w=0.33,c=0.38)
	1398.HK	MVT=U (w=0.23, c=0.45)	TVL =M (w=0.41,c=0.36)	DIFF =S(w=0.28,c=0.40)
Services	**0066.HK**	DIFF=H(w=0.25,c=0.45)	TVL =M (w=0.44,c=0.33)	DIFF =S(w=0.27,c=0.37)
	1928.HK	DIFF=H(w=0.11,c=0.38)	TVL =M (w=0.6,c=0.33)	DIFF =S(w=0.24,c=0.40)
Technology	**0700.HK**	MVT=U (w=0.15, c=0.44)	MVT =NC(w=0.35,c=0.31)	TVL =S (w=0.2, c=0.34)
	0941.HK	MVT=NC(w=0.21,c=0.46)	MVT =NC(w=0.36,c=0.33)	TVL =S(w=0.17, c=0.34)
Utilities	**0003.HK**	DIFF=L(w=0.24,c=0.41)	TVL =M (w=0.47,c=0.37)	MVT=D(w=0.18, c=0.40)
	0836.HK	DIFF=L(w=0.25,c=0.43)	TVL =M (w=0.38, c=0.34)	MVT=D(w=0.12, c=0.40)

6.1.4.2 How Many Days Ahead?

The above results show that for the case of the NYSE dataset, discovering daily temporal associations allows us to predict the next day's stock price movement based on that of the current day, with an average accuracy of about 60%. In the case of HKSE, it can be as high as 77.41%. One may ask if we are to predict tomorrow's stock movement based also on the previous two to three days of data, will we be able to make more accurate decisions? In order to answer this question, we now consider how many days we should look back when predicting the price movement of the current day.

Figure 6.2 illustrates the idea of the days of look-back. It can be any number of days from 1 to 5 and even longer. In this experiment, when considering “1 day”, we focus on discovering the fuzzy associations between the value of different parameters on the previous day and the current day. When “2 days” of look-back are considered, we also try to consider fuzzy temporal associations between the average values of different parameters between the current day, the previous day and the day prior to the previous day.

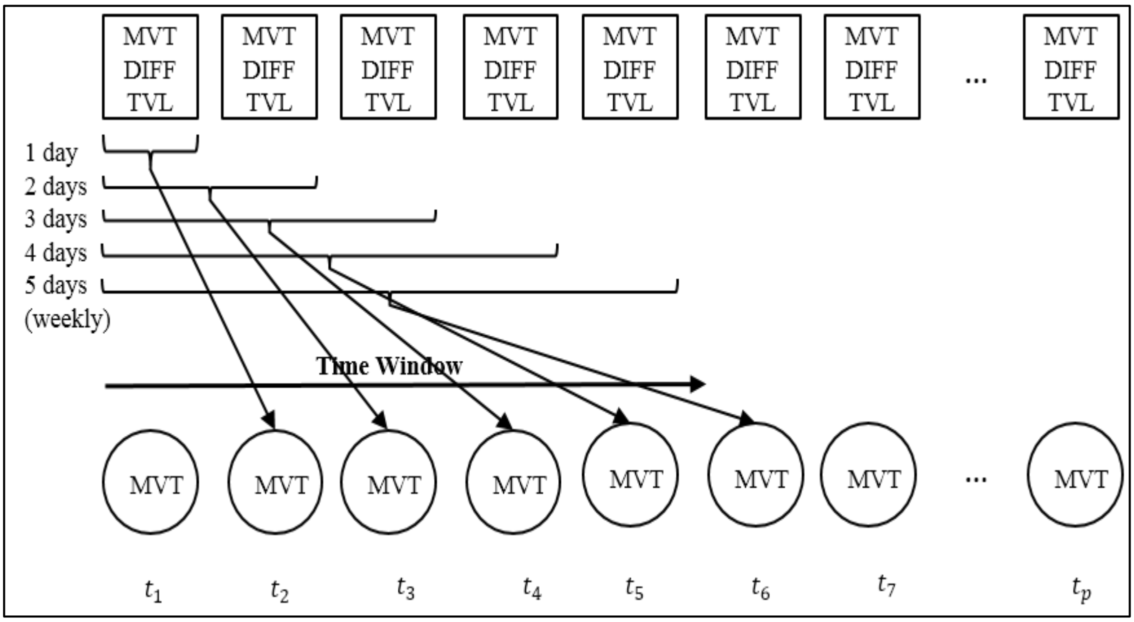

Figure 6.2 Different Time Intervals for Prediction

The averages of the prediction accuracy of all stocks from the eight categories collected from the NYSE data are shown in Table 6.7. We can conclude from it that when look-back is set to “3 days”, the average of the prediction accuracy is at the highest with 65.08%. And, for different categories, the best results for different look-back are actually different. For example, in the category of Industrial Goods and

Utilities, the prediction accuracies are at their highest when the look-back is set to “2 days”, while for the stocks in Healthcare and Technology, the prediction accuracy is at their highest when the look-back is set to “1 day”.

Table 6.7 Prediction Results for Stocks from the NYSE Using Different Time Intervals

Prediction Accuracy	**1 Day**	**2 Days**	**3 Days**	**4 Days**	**5 Days**
Basic Materials	53.26%	52.26%	**56.28%**	51.26%	52.26%
Consumer Goods	56.28%	63.32%	**73.37%**	74.37%	66.33%
Financial	53.38%	53.27%	**63.32%**	57.29%	51.26%
Healthcare	**55.35%**	54.27%	54.27%	51.26%	50.25%
Industrial Good	60.05%	**69.35%**	65.33%	64.32%	64.32%
Services	71.49%	72.36%	**78.39%**	75.38%	77.39%
Technology	**68.08%**	59.30%	67.34%	67.34%	61.31%
Utilities	62.15%	**66.33%**	62.31%	65.33%	62.31%
Average	**60.01%**	**61.31%**	**65.08%**	**63.32%**	**60.68%**

As for HKSE stocks, we have performed similar experiments to investigate into the effects of the duration of look-back on the prediction accuracy. The results are shown in Table 6.8. We note that when the look-back is set to “4 days”, the average of prediction accuracy is at its highest at 79.60%. But, in general, compared with the NYSE, the differences that the setting of the look-back makes are very small. In fact, the same can be said for all stocks in all industrial categories.

Table 6.8 Prediction Result for Stocks from the HKSE Using Different Time Intervals

Prediction Accuracy	**1 Day**	**2 Days**	**3 Days**	**4 Days**	**5 Days**
Basic Materials	86.72%	87.65%	**89.75%**	89.22%	88.73%
Consumer Goods	89.75%	87.65%	89.75%	91.76%	**93.63%**
Financial	75.63%	**81.68%**	75.63%	76.64%	75.63%
Services	76.90%	**81.68%**	75.63%	76.64%	75.63%

Technology	65.88%	66.55%	70.70%	70.70%	**71.21%**
Utilities	69.58%	64.54%	**75.63%**	72.61%	69.58%
Average	**77.41%**	**78.29%**	**79.52%**	**79.60%**	**79.07%**

6.2 Stock Clustering

In data mining finance, there has been some effort to make use of clustering algorithms such as the *k-means* and the *hierarchical agglomerative* algorithm, to cluster stocks with similar trends and behaviors into the same groups. For instance, in (He, H. et al. 2006), the k-means clustering algorithm and the linear regression technique have been used to partition a set of stock time series to uncover trends to form different clusters. In (Ozkan, I. et al. 2008), the fuzzy c-means algorithm is used to develop a perception based decision matrix to analyze currency crises. In (Wu, K.P. et al. 2014), a combination of k-means and a priori. All are presented to predict stock trends. In (Gavrilov, M. et al. 2000) the hierarchical agglomerative clustering algorithm is used to identify the industrial category to which a stock belongs, given only the historical price records. There is a less suitable method to make use of any additional information even if they are made available. Besides, clustering stocks is very difficult because stocks are correlated across industries (Todd, W. 2002), hence, a natural extension of the attempt is to transform precise clustering into fuzzy clustering through considering the relationships between the difference variables with the price of stock.

6.2.1 Clustering Analysis for Stocks

Various clustering techniques have been used in various research areas, such as mathematics, multimedia, finance and other domains (Nanda, S.R. et al. 2010). In finance, clustering algorithms are present in several applications (Lemieux V. et al.

2014). For example, Musetti (Musetti, A.T.Y. 2012) compared several clustering algorithms to study S&P 100, and select the best clustering algorithm for market segmentation. De Andrés et al. (De Andrés, J. et al. 2011) proposed a hybrid system which combines fuzzy clustering and multivariate adaptive regression splines and applies it in the forecasting of bankruptcy.

The *K*-means clustering algorithm is one of the simplest unsupervised learning algorithms, which can be used to partition n observations into k clusters with each observation belonging to the cluster with the nearest mean (McQueen, J. 1967). *Fuzzy C-means* is developed by Bezdek (Bezdek, J.C. 1981), which is a variation of the K-means clustering algorithm. It can assign data elements into more than one cluster with different degrees of membership. In some cases it can perform better than hard clustering methods because of its ability to assign probability of a data point belonging to a particular cluster.

In the area of financial analysis, K-means and Fuzzy c-means have found increasing use. For instance, He *et al.* (He, H. et al. 2006) used the *K*-means clustering algorithm and linear regression to partition stock price time series data and analyze the trend within each cluster. Ozkan (Ozkan, I. et al. 2008) employs fuzzy system modeling with fuzzy c-means clustering to develop a perception-based decision matrix to analyze currency crises within the decision theory framework. Wu et al. (Wu, K.P. et al. 2014) present a model to predict stock trends based on a combination of sequential chart association, k-means and the a priori algorithm.

By using the clustering result, another significant application is in creating efficient portfolios to helps investors to decide where and how to invest the earned money for guaranteed returns in the future (Nanda, S.R. et al. 2010). By using portfolio management, it can refer to the art of managing various financial products and assets to help an individual earn maximum revenues with minimum risks over the long run. The classical model of creating efficient portfolio was developed by Markowitz (Beste, A. et al. 2002) which considered that the mean return is the return of a stock and the standard deviation of stock returns is the risk of a stock. Subsequently, a number of works on portfolio management were done. For instance, the authors worked on a dynamic stochastic programming model for international portfolio management, a solution that determines capital allocations to international markets, the selection of assets within each market, and appropriate currency hedging levels (Topaloglou, N. et al. 2008). Oh et al (Oh, K.J. et al. 2005) applied GA algorithm into portfolio optimization problem for index fund management. Fernandez (Fernandez, L. 2005) presented a stochastic control model that includes ecological and economic uncertainty for jointly managing both types of natural resources. Besides, Fuzzy models (Ostermark, R. 1996) for dynamic portfolio management have also been implemented.

From the above literature review we can see that the current approaches for analyzing stocks data only consider the ups and downs of stock, but ignore the interrelationships between different variables of stock. Hence, in the following sections, we model stocks as multivariate time series data and propose an algorithm to do association analysis, clustering and portfolio management in the application of stock market.

6.2.2 Comparison Result for Stock Clustering

With the above datasets, we performed several benchmarking experiments aiming to compare FMTSC with i) *raw-based clustering*: k-means clustering using raw data from different stock exchanges (K-Means). ii) *feature-based clustering:* k-means clustering using features vector which extracted using PCA (PCA Clustering) (Lameira, P. 2015), and iii) *model-based clustering algorithm*: k-means clustering using coefficients of ARMA model (ARMA Clustering) (Kalpakis, K. et al. 2001).

As for the stocks collected from the NYSE, we used 40 stocks chosen from the 65 stocks randomly as training data and the remaining 25 as testing data. As for the stocks collected from the HKSE, 30 stocks were chosen from 50 stocks as training data and the remaining 20 stocks are used as testing data. The experimental result using the NYSE and the HKSE stocks are shown in Tables 6.9 and 6.10, respectively.

By using the proposed FMTSC algorithm, in order to clustering the stocks, each set of stock is treated as a multivariate time series data which is made up of several temporally univariate time series (i.e. open, high, low, close, volume). Considering the application of the stock market algorithm, price fluctuation ratio was used to represent the variables. The original stock data were transformed into multivariate time series data with three parameters: i) the movement of stock price (*MVT*); ii) the differences between the maximum and minimum stock price of a stock (*DIFF*), and iii) the trading volume (*TVL*). Next, the intra-FTAs and inter-FTAs were discovered to understand how the movement (*MVT*) may be affected by movement of the previous time instants and the differences between the maximum and minimum stock price of a stock (*DIFF*) and its trading volume (*TVL*) at the last time instants. Next, the feature vector containing all *significant discrepancy measures* of temporal associations is obtained for each set of stock. Then, the K-means clustering algorithm is applied for all the feature vectors to cluster all the stocks into different groups. Finally, a decision tree (C5.0) is constructed using a set of training data based on clustering result (Ma.P.C.H. et al. 2005). The classification accuracy can reflect the quality of the discovered clusters. As shown in Tables 6.9 and 6.10, the classification accuracy is calculated using the rest of the testing data. The average of classification accuracy resulting from the proposed algorithm was 91% for the NYSE dataset and 86% for the HKSE dataset respectively.

As for experiments involving the K-means clustering algorithm, we set the maximum number of iteration as 1000 with Euclidean distance measure for K-means algorithm. We set the number of clusters as k=3 to 8 and k=2 to 6 respectively for the NYSE dataset and the HKSE dataset. Under these settings, the k-means algorithm was applied to raw stock data only when the movement parameter was considered. The average prediction accuracy was 72% for the NYSE dataset and 76% for the HKSE dataset.

Table 6.9 Comparing the Results from the Proposed Algorithm with Those from Other Approaches for the NYSE Dataset

Value of K	K means	PCA	ARMA	FMTSC
K=3	78%	92%	76%	96%
K=4	68%	92%	52%	84%
K=5	68%	80%	60%	96%
K=6	65%	84%	88%	92%
K=7	71%	80%	88%	88%
K=8	79%	82%	85%	89%
average	**72%**	**85%**	**75%**	**91%**

Table 6.10 Comparing the Results from the Proposed Algorithm with Those from Other Approaches for the HKSE Dataset

Value of K	K means	PCA	ARMA	FMTSC
K=3	79%	84%	80%	92%
K=4	72%	81%	69%	82%
K=5	75%	79%	71%	89%
K=6	76%	82%	65%	81%
average	**76%**	**81.25%**	**71.25%**	**86%**

As the second feature-based clustering approach, we use Principal Components Analysis (PCA) to project original multivariate stock time series data into univariate time series data to reduce the dimension of original data. PCA performs analysis on $n \times p$ data matrix S and returns the principal component coefficients. In this study, p is the number of variables of stocks, and n is the number of records/observations of stocks. Then p coefficients are returns for variables. Finally, a univariate time series, U_i, is achieved as $\mathrm{U}_i = \sum_{j=1}^{p} S_j^i$. And then K-means with Euclidean distance is used to cluster the data. The average classification accuracy from this approach was 85% for the NYSE dataset and 81.25% for the HKSE dataset respectively.

The third approach made use of a model-based clustering algorithm. We also used PCA to transform original data into univariate time series data firstly, and then use ARMA to construct a linear model of the univariate time series data. The model can return the coefficients of a pth-order linear predictor. We used the same parameters setting as in (Kalpakis, K. et al. 2001), the first eight LPCs (Linear Prediction Filter Coefficients) were chosen to represent each stock. And then K-means with Euclidean distance was used to cluster the transformed data. Since the original high dimensions of the original multivariate time series data were reduced to eight values, there might have been substantial information loss during dimension reduction. The average of classification accuracy was only 75% for the NYSE dataset and 71.25% for the HKSE dataset.

The above experimental results show that the fuzzy associations discovered by FMTSC can represent original stock data more accurately and effectively. To better visualize the comparison results, we now summarize the results of the performance of the different algorithms based on their prediction accuracies for the two datasets.

Based on these results shown in Figure 6.3, FMTSC seems to perform consistently better than other algorithms.

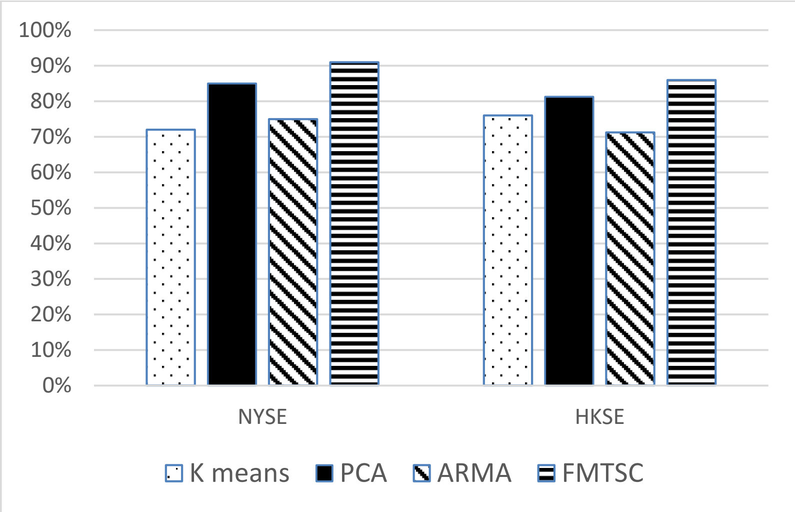

Figure 6.3 Comparison of Clustering Result for the Two Datasets

6.2.3 Creating the Efficient Portfolio for Stock Management

As for the NYSE data, using the industry classifications given by (Yahoo Financial), the 65 stocks span 8 industrial categories: Basic Materials, Consumer Goods, Financial, Healthcare, Industrial Goods, Services, Technology and Utilities. Similarly, as for the HKSE stocks, the 50 stocks span 6 industrial categories: Basic Materials, Consumer Goods, Financial, Services, Technology and Utilities. Stocks may not belong to a single category because of the existence of certain commonalities between industrial categories (Todd, W. 2002). Hence, in this section, we describe a different clustering algorithm to improve the original categorization of stocks and arrive at different portfolios for stocks.

Firstly, we apply FMTSC to stock data to discover fuzzy temporal associations. Once the feature vectors have been discovered for each stock, K-means and Fuzzy C-means are applied for clustering the stocks. In this study, we set the maximum number of iterations as 1000 and incorporate a Euclidean distance measure in the K-means algorithm. As for the Fuzzy C-means algorithm, we set the maximum number iterations as 1000 again, the exponent for the partition matrix as 1.5, and the minimum amount of improvement as $1e^{-5}$. After getting the clustering result, the Markowitz model is used to select the sample stocks with highest weight from different groups. For evaluating the clustering performance using the Markowitz model, 2013–2014 records are used while the remaining records in 2015 are used for testing.

Table 6.11 Weights for Stocks Chosen for Portfolio in the NYSE Dataset

Portfolio 1: Original Industrials	
Stocks	**Weight**
The Procter & Gamble Company	0.19
E. I. du Pont de Nemours and Company	0.20
The Goldman Sachs Group, Inc.	0
UnitedHealth Group Incorporated	0.14
United Technologies Corporation	0.06
Avis Budget Group, Inc.	0
Verizon Communications Inc.	0.21
NiSource Inc.	0.20
Portfolio 2: FMTSC + Fuzzy c-mean	
Stocks	**Weight**
UnitedHealth Group Incorporated	0.11
3M Company	0.24
Southwest Airlines Co.	0.08
Microsoft Corporation	0.05
Cisco Systems, Inc.	0.2
FedEx Corporation	0.08
Exelon Corporation	0.24
Avis Budget Group, Inc.	0
Portfolio 3: FMTSC + K-means	
Stocks	**Weight**
Con-way Inc.	0.03
Matson, Inc.	0
The Walt Disney Company	0.25
Avis Budget Group, Inc.	0
The Goldman Sachs Group, Inc.	0
FedEx Corporation	0.19
Delta Air Lines, Inc.	0.06
Cisco Systems, Inc.	0.47

For the first portfolio (p1), the original industry categories without clustering are used; for the second portfolio (p2), the clustering results from the proposed algorithm with fuzzy-c means are used; and, for the third portfolio (p3), clustering results from the proposed algorithm with K-means are used. Finally, the selected stocks and their corresponding weights are found by using Markowitz model to create portfolios. The weekly returns are used to test the performance of different portfolios. Tables 6.11 and 6.12 show the stocks selected from different groups and their corresponding weights for the stocks from the NYSE and the HKSE, respectively. This cluster-based approach has been found to be able to significantly reduce the times needed for creating an efficient portfolio.

Table 6.12 Weights for Stocks chosen for the Portfolios for the HKSE Dataset

Portfolio 1: Original Industrials		**Portfolio 2: FMTSC + Fuzzy c-mean**		**Portfolio 3: FMTSC + K-means**	
Stocks	**Weight**	**Stocks**	**Weight**	**Stocks**	**Weight**
0883.HK	0.05	0857.HK	0.06	0823.HK	0.08
0151.HK	0.09	0005.HK	0.64	0066.HK	0.11
0005.HK	0.55	0823.HK	0.14	0005.HK	0.50
0066.HK	0.12	0386.HK	0.00	0002.HK	0.17
0941.HK	0.00	0066.HK	0.16	0003.HK	0.14
0002.HK	0.19	1880.HK	0.00	1880.HK	0.00

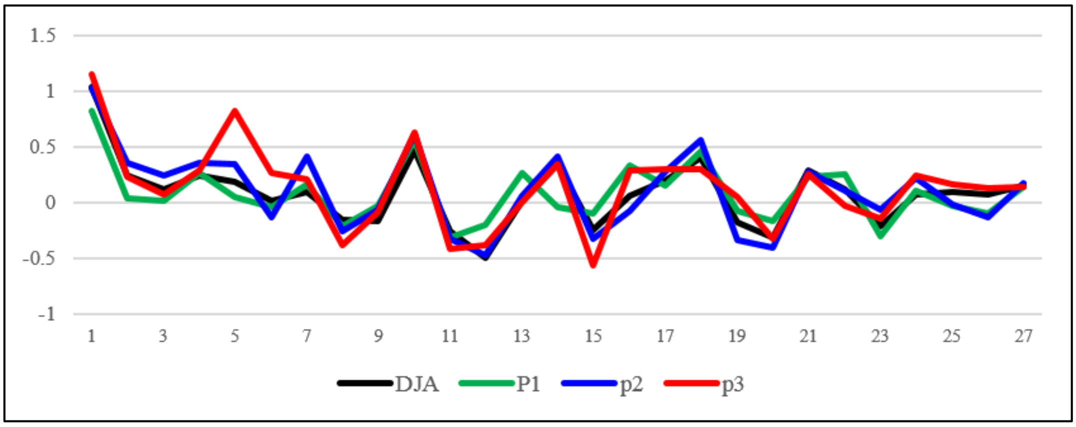

Figure 6.4 Plots of Returns from Different NYSE Portfolios

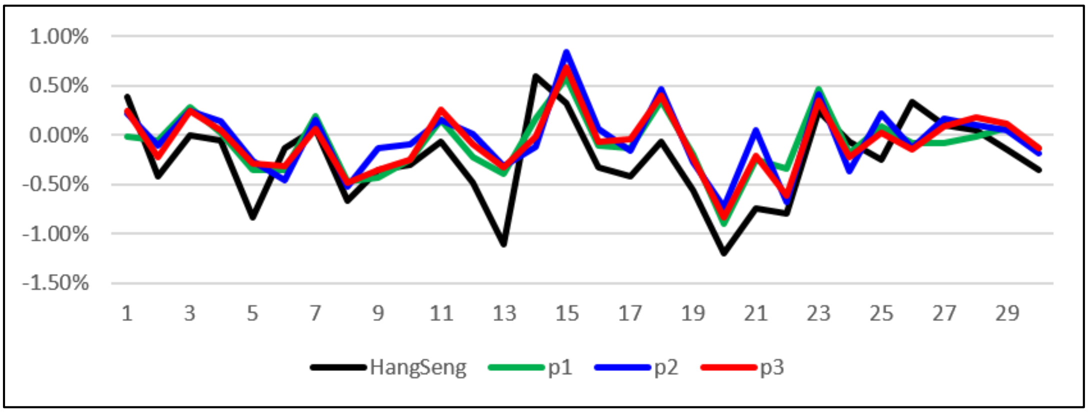

Figure 6.5 Plots of Returns from Different HKSE Portfolios

To better visualize the comparison results, Figures 6.4 and 6.5 show the weekly returns from the NYSE and the HKSE stocks over the previous 27 weeks.

Finally, Tables 6.13 and 6.14 present the comparison results of total returns using testing data for the NYSE and the HKSE stocks, respectively. Note that the total returns from the original index are 2.19% and -6.9% for the NYSE and the HKSE stocks, respectively. When the original industrial categories were used to create portfolios using the Markowitz model, the returns are improved to 1.66% and 65.18%. And when the FMTSC with K-means and Fuzzy C-means was used to cluster the stocks, the portfolios could improve the returns even further. Hence, we can conclude that the proposed algorithm can indeed improve the industrial categories for stocks.

Table 6.13 Comparison Results for Total Returns from Different NYSE Portfolios

Different Portfolios	**Total Return**	**Improved Return**
DJA	2.19%	-
Portfolio 1 (Original Industries)	2.23%	1.66%
Portfolio 2 (Proposed Algorithm +Fuzzy c-means)	2.88%	31.14%
Portfolio 3 (Proposed Algorithm + K-means)	3.56%	61.94%

Table 6.14 Comparison Results for Total Returns of Different HKSE Portfolios

Different Portfolios	**Total Return**	**Improved Return**
HSI	-6.90%	-
Portfolio 1 (Original Industries)	-2.47%	64.18%
Portfolio 2 (Proposed Algorithm +Fuzzy c-means)	-1.04%	84.86%
Portfolio 3 (Proposed Algorithm + K-means)	-2.07%	69.99%

In summary, the work presented above is motivated by the lack of a suitable algorithm that can deal effectively with multivariate time series data arising in stock analyses. FMTSC is therefore developed for this purpose. By making use of the concept of fuzzy sets, FMTSC is able to discover inter- and intra-fuzzy temporal associations in multivariate time series data for stock analysis. It is able to discover patterns even when the lengths of the time series being analyzed are different and even when the data are noisy or some data are missing. FMTSC has been tested with synthetic and real data. Experimental results have indicated that FMTSC can be an effective approach for stock analysis.

The contribution of the proposed algorithm, from a theoretical and methodological perspective to real world applications, has been confirmed. Using FMTSC, we have confirmed that there are indeed interesting associations in stock data that can be discovered for predicting not only daily price movements but also semi-weekly, or weekly price movements. With FMTSC, we have also determined the number of days of look-back that it takes to arrive at the most accurate predictions of stock movements. In addition, the method's usefulness for conducting a stock market analysis has been demonstrated by predicting the stock prices, improving the industrial categories of stocks, and creating portfolios for helping investors. The data from New York Stock Exchange was used to test and the results show that the proposed algorithm has high promise for analyzing multivariate time series data.

6.3 Multi-Channel EEG Classification

As a result of an ionic current flowing within the neurons of the brain, electrical activities over the scalp can be detected. These activities can be recorded in the form of electroencephalograph (EEG) signals (Lofhede, J. et al. 2008) in a set-up such as that shown in Figure 6.6, where electrodes are placed along the scalp to record cerebral electrical activities at different positions. EEG signals have been shown to reflect the prevailing state of the brain and can be useful for neuroscience or neurological practices such as the classification of seizure types and the diagnosis of epileptic disorders (Niedermeyer, E. et al. 2005). They can also be analyzed for gaining a better understanding of human thought processes for clinical and scientific applications in psychology, pharmacy, linguistics, and biomedicine or for better human-computer interactions (Kubler, A. et al. 2005).

In order to obtain EEG data for analysis, one or more human subjects are usually presented with the same or different stimuli to see how they respond to them. The similarities and differences between different EEG signals obtained at different locations on the scalp of the same subject or at the same location on the scalp of different subjects (Kaplan, A.Y. et al. 2000) can then be analyzed. To classify or differentiate between the EEG signals of different human subjects, EEG data can be analyzed visually using a variety of visualization software (Kubler, A. et al. 2005). However, due to the small differences in measurements and the presence of continuing brain dynamics, discovering interesting patterns in the data by visualization is very difficult. Different computational approaches have, therefore, been used for this task. Since EEG Signals are often analyzed to classify and predict mental states, these approaches are, by and large, based on the use of different classification algorithms. Given the increasing need for EEG data collected at multiple channels to be accurately analyzed and classified for various applications, computational algorithms capable of discovering patterns from multi-channel EEG data have to be developed (Kaplan, A.Y. et al. 2000). In this section, we apply the proposed FMTSC to single-trial EEG analysis.

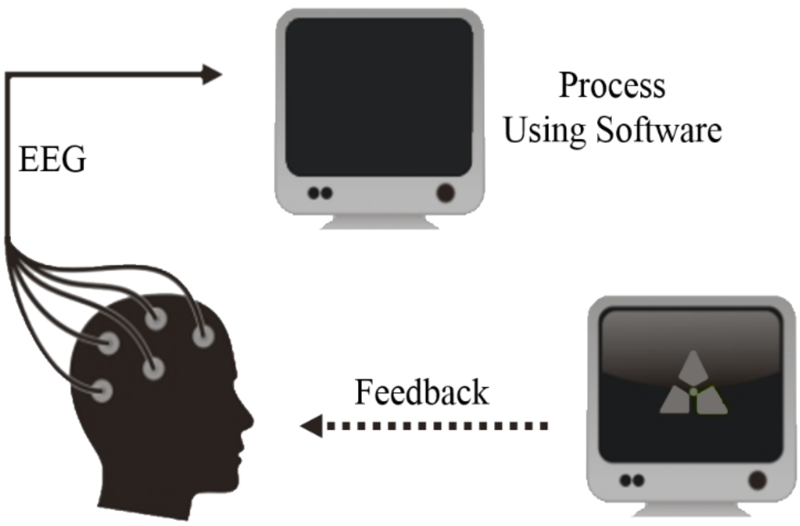

Figure 6.6 Method to Collect EEG Data

Our FMTSC is able to discover i) intra-channel patterns for each channel of EEG data being monitored and ii) inter-channel patterns between different channels of EEG data collected at different locations and time instants. Since the data are noisy, FMTSC makes use of probabilistic and information theoretic measures for discovering these patterns, since the patterns discovered are not expected to be described and represented precisely, FMTSC makes use of fuzzy set concepts to represent and model them.

FMTSC has been tested for performance evaluation using several sets of real EEG data. The results have shown that it is a promising algorithm for classifying multi-channel EEG data.

6.3.1 Previous Work about Classification Multi-Channel EEG Signals

Several computational approaches have already been developed to classify multi-channel EEG signal data (Blankertz, B. et al. 2011). The general approaches can be classified roughly into *raw-data based* or *feature based* approaches.

For methods based on the raw-data based approach, raw EEG data are usually pre-processed using such techniques as Common Average Referencing (CAR) (Alhaddad, M.J. et al. 2012) and Independent Component Analysis (ICA) (Alhaddad, M.J. et al. 2012; Bell, A.J. et al. 1995) with the intention to improve signal-to-noise ratio (SNR) and to tackle the blind-source-separation problem by decomposing signals into temporally independent and spatially fixed components. After pre-processing, data from the best channel are selected as input for a classifier (Bell, A.J. et al. 1995). Wong et al (Wong, D.K. et al. 2004; Wong, D.K. et al. 2006) provides some examples of how methods based on the raw-data based approach can be used to classify multi-channel EEG data. As these examples show, ICA is usually first used to address the blind-source separation problem. A single channel classifier is then used to select the best channel from amongst all possible channels, in terms of classification accuracy. In some cases, instead of selecting one single best-channel, the best *k* channels are selected. When deciding which *k* channels must be to be selected, classification accuracy is once again taken as the main criterion. However, the data from multi-channels are handled by concatenating data of each channel into a long single signal, so that they can be used with such classifiers as SVM or ANN. In other words, the data collected from each channel are considered independently of each other. The raw-data based approach to classifying multi-channel EEG data is illustrated in Figure 6.7.

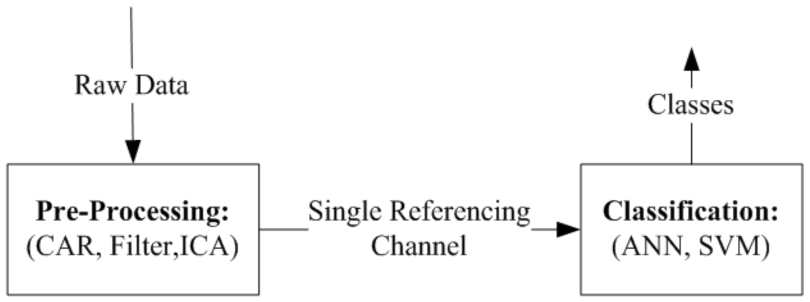

Figure 6.7 Process of Raw-Data Based Classification

Compared to the raw-data based approach, methods adopting a feature-based approach (Tian, L. et al. 2006; Gutierrez, D. et al. 2005; Subasi, A.et al. 2005) are much more popular. This kind of algorithms applies a transformation method to convert the multi-channel EEG signal into a new signal and input it into a standard machine-learning classifier. In general, these methods work in three main steps (Figure 6.8). The first step involves the pre-processing of data to remove noise (Suleiman, A.B.R. et al. 2007; Hosni, S.S. et al. 2007) using an algorithm such as the ICA (Chiappa, S. et al. 2005; Bell, A.J. et al. 1995). After identifying and removing irrelevant components from the data, *feature extraction* algorithms, such as Autoregressive (AR) modeling (Subasi, A.et al. 2005), AR spectral analysis (Akin, M. et al. 2000), Band Powers and power differences (Wolpaw, J.R. et al. 2002), etc., are used to extract features from the pre-processed data. These features consist typically of a set of "coefficients" of the linear models built for each channel of EEG data. For example, after pre-processing, AR modeling may be used to extract features from a $N \times M$ (N channels, M data time points) EEG pre-processed data matrix into an $N \times P$ feature data matrix (N channel, P coefficients, $P<N$) (Gutierrez, D. et al. 2005). Once the features have been extracted, classification algorithms such as SVM and ANN are used to classify the feature vectors.

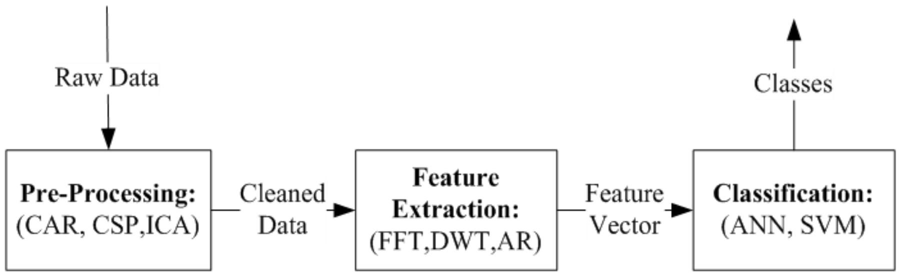

Figure 6.8 Process of Feature-Based Classification

Instead of considering all channels of EEG data in the classification process, a subset of channels is sometimes selected for the classification process so as to reduce training time and improve accuracy of the process. To select the specific subset of channels, some measures are usually used to rank the usefulness of each channel of data being examined for classification. Some researches (Gutierrez, D. et al. 2005; Nandish, M. et al. 2012; Tian, L. et al. 2006) have proposed selecting the channels by extracting frequency-band features, such as delta, theta, alpha, beta, and gamma, for each channel using band pass filters. Based on these features, the Mutual Information (MI) measure can then be computed for ranking all channels. The highest rank n channels are then selected to form n feature vectors of dimension 5 each, thus arriving at a matrix of size $5 \times n$. Additional measures such as taking the average or finding the maximum and minimum of the frequency features may also be considered. The features identified can then be used for classification using an SVM- or ANN-based classifier.

In addition, in order to improve the classification process, the fuzzy approach is usually considered. There have been some attempts (Tian, L. et al. 2006) to use a fuzzy technique to classify EEG data. Similar to traditional crisp classification algorithms, this approach combines multi-channel EEG and applies a fuzzy classifier to the combined data. Coyle et al. (Coyle, D. et al. 2008) first classify EEG data using a feature-based approach. The approach extracts features using a self-organizing fuzzy neural network and then classifies EEG data based on these features using a classifier. It is shown that this approach arrives at a more accurate classification than that achievable by working with raw EEG data. Coyle, D. et al. (2009) have implemented a two-class EEG based classifier. Likewise, Lotte F. (2006) uses an epilepsy disorder dataset consisting of data divided into three subsets for evaluating their fuzzy EEG data classification algorithm. This algorithm extracts and makes use of three features - Detrended Fluctuation Analysis (DFA), Bispectral Analysis (BIS), and Standard Deviation (SD) - obtained from single-channel EEG data, and classifies the EEG data using a fuzzy classifier. More recently, Deng et al. (Deng, Z.H. et al. 2015) have proposed a minimal probability fuzzy system classifier for EEG signal recognition capable of training the classifier and simultaneously determining the model's reliability.

Both the above algorithms, the raw-data and the feature-based approaches, have been used effectively for multi-channel EEG data classification. They have been shown to be useful for a number of different applications (Lofhede, J. et al. 2008; Mueller, A. et al. 2010). However, as described above, these approaches basically treat data from different channels independently and, as a result, temporal associations between different channels considered sufficiently, resulting in the loss of information which may not facilitate the classification process. In order to avoid any loss of information, we propose a feature-based classification method, FMTSC, to address this issues by taking into consideration both intra- and inter-channel fuzzy temporal patterns.

6.3.3 Datasets Description

To evaluate the performance of FMTSC, we have performed a number of experiments using several sets of real multi-channel EEG datasets used for evaluation: *Gender Discrimination Dataset (GDD Data)*, *Alcoholism Dataset (ALC Data)* and *Epilepsy Dataset (EPL Data)*. Using these datasets, the first set of experiments is concerned with the evaluation of the effectiveness of FMTSC in the selection of the most appropriate channels from among all possible channels for tasks related to gender discrimination, genetic disposition and alcoholism, and response to signals by epilepsy patients. Using the three sets of data, we also performed several experiments that aimed to compare the accuracies of several different algorithms usable for multi-channel EEG classification. We now present the results and discuss our findings.

The Gender Discrimination Data Set

The first set of real data, *GDD Data*, we used in our experiments was a set of multi-channel EEG data recorded for a normal, right-handed subject, who was 24 years old paid volunteer with normal vision (Inoue, Takuya. Et al. 2001). The ID card photos of the faces of a company's male and female employees, aged between 23 to 31, were presented to the subject. The subject was requested to distinguish photos of the different genders.

For the purpose of obtaining the EEG data, 160 stimuli were used. In the experiments, the subject was fitted with a 64-channel electrode cap and the EEG data were recorded with 32-bit quantization level at a sampling rate of 1000Hz by using the Neuro Scan software. Hence, the data collected had a total size of 160 (trials) × 64 (channels) ×600 (time points). In other words, we had 160 sets of multi-channel EEG signal data and, for each dataset, we had 64 time series of length 600. Using the notation used in Section 3.1, M = 160, n = 64, p_k = 600 for all k.

The Alcoholism Data Set

The second set of real data, *ALC Data*, we used in our experiments was obtained from a study aiming to examine the correlation between genetic predisposition and alcoholism. This set of data is made available in the UCI repository (Larsen, B. et al. 1999). It contains measurements from 64 electrodes placed on the subjects' scalps which were sampled at 256 Hz (3.9-msec epoch). Two groups of subjects, *alcoholic* and *control*, were requested to perform a visual matching task. Each subject was exposed to either a single stimulus (S_1) or a pair of stimuli (S_1 and S_2) consisting of pictures of objects chosen from the 1980 Snodgrass and Vanderwart picture sets.

For the purpose of our experiments, six subjects, three from the alcoholic group and three from the control group were selected. Each subject was asked to complete 50 trials for which different stimuli were shown to them. Hence, the total size of the dataset was size 300 (trials) × 64 (channels) × 256 (time points). In other words, we had 300 sets of multi-channel EEG signal data and, for each dataset, we had 64 time series of length 256. Using the notation in Section 3.1, M = 300, n = 64, p_k = 256 for all k.

Epilepsy Dataset

The third set of data, *EPL Data*, used in our experiments was taken from (Murph P.M. et al. 1999). It consists of multi-channel EEG data collected as focal and non-focal signals from epilepsy patients. In the experiments, data were collected only from the Fz and Pz channels corresponding to the extracranial reference electrodes. EEG signals were sampled at 1024 Hz. The data were collected using 50 focal signals and 50 non-focal signals. In other words, the total size of the data obtained was 100 (trials) $\times 2$ (channels) $\times$ 1024 (time points), i.e., we have 100 sets of multi-channel EEG signal data, and for each dataset, we have 2 time series of length 1024. Using the notation in Section 3.1, $M = 100$, $n = 2$, $p_k = 1024$ for all k.

6.3.4 Performance Evaluation

Before the datasets described above were used in our experiments, they were pre-processed using the ICA algorithm to separate the source components of the EEG data signals. The first set of experiments aimed at testing how effective FMTSC was in selecting appropriate channels to collect EEG data for different tasks. For the *GDD Data*set, the set of channels that FMTSC identified for the gender discrimination task were compared with that found by (Inoue, Takuya. Et al. 2001) (see Figure 6.9 (a)). As for the *ALC Data*, the channels identified by FMTSC were compared with those identified in (Andrzejak, R.G. et al. 2012) (see Figure 6.9 (b)). As for the *EPL Data*, they were collected from two channels, Pz and Fz. We only considered relationships within and between Pz and Fz in the experiments, so further channel selection was not considered.

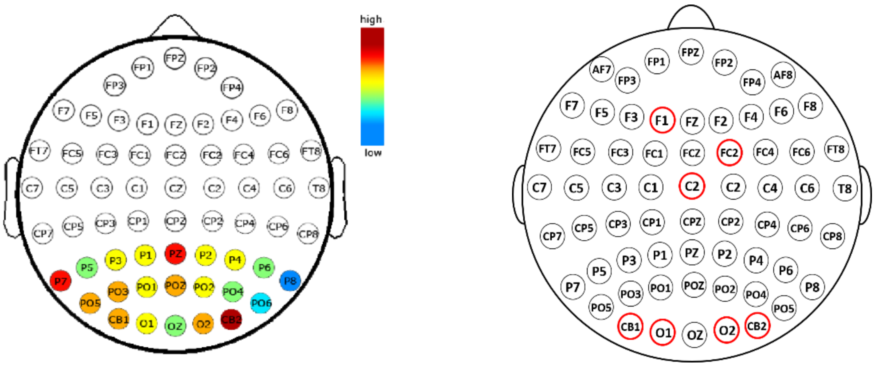

Figure 6.9 (a) Notable Channels for GDD Data Set (Inoue, Takuya. Et al. 2001)

(b) Optimal Channels Location for ALC Data Set (Andrzejak, R.G. et al. 2012)

Other than the experiments performed to evaluate the effectiveness of FMTSC in selecting appropriate channels for various applications, we have also performed a number of tests to determine how accurate FMTSC is, in classifying multi-channel EEG signals. The tests included i) classify each selected channel of EEG data independently to determine the average classification accuracy (Original Data), ii) execute raw-based classification to pre-process data with CAR and then classify each channel of EEG data independently to determine the average classification accuracy (Lotte, F. et al. 2007) (CAR+SVM), iii) use the ICA-based Single Channel Classifier (Lotte, F. et al. 2007) to classify the EEG data (ICA-SCC), (iv) discretize the EEG data so that a non-fuzzy version of FMTSC can be used to discover patterns in the data so that an SVM can construct classifiers based on the patterns (No-Fuzzy), and (v) fuzzify the data so that the FMTSC can be used to discover fuzzy temporal patterns for SVM classifiers to be constructed (FMTSC).

In all our experiments, 85% of the available data was selected randomly as training data and the rest of the 15% was selected as testing data. The experiments are iterated to calculate average classification accuracy and average value of *F-measure/F1-Score* (Larsen, B. et al. 1999) for performance assessment and comparison. All the measures were used to determine how good the classification results are and based on the confusion matrix presented in Table 3.1.

Since the “correct” class membership is known, *classification accuracy* (ACC.) is the most intuitive evaluation measure that can be used to evaluate the effectiveness of classification algorithms.

Based on the confusion matrix, C_i, $i \in \{1, \ldots, n\}$ is the label of the predicted class and C_j, $j \in \{1, \ldots, n\}$ is the label of original assigned class. TP_i represents the number of true positive instants of class C_i, and FN_{ij} represents the number of instances that have been predicted to be members of the class C_j with original class label as C_i. The ACC. is defined in Eqn. (6.1)

$$\text{Classification Accuracy} = \frac{\sum_i TP_i}{\sum_i \sum_j FN_{ij} + \sum_i TP_i} \times 100\% \tag{6.1}$$

The *F*-measure (Larsen, B. et al. 1999) is defined for C_i as Eqn. (6.2).

$$F\text{-measure}(C_i) = \frac{2 \times \text{Recall}(C_i)\text{Precision}(C_i)}{\text{Recall}(C_i) + \text{Precision}(C_i)}, \tag{6.2}$$

where $Precision\ (C_i) = \frac{TP_i}{TP_i + \sum_j FN_{ji}}$ evaluates how many instances were correctly predicted in C_i and $Recall(C_i) = \frac{TP_i}{TP_i + \sum_j FN_{ij}}$ evaluates how many of real instances were correctly captured. Given these definitions, the *F*-measure takes on values in the interval [0, 1]; the closer its value is to 1, the better the classification quality it reflects. In our study, we used the *average of F-measure* to evaluate the final classification result which could be calculated as $\frac{\sum_i F\text{-measure}(C_i)}{n}$.

For the purpose of performance evaluation, 85% of the data were selected randomly as training data and the rest 15% as testing data. We iterated the classification process to obtain the final average value of the above measurements.

6.3.5 Experimental Result

To evaluate the performance of FMTSC, we conducted tests using the above datasets (the experimental processes have already been described in Section 6.3.3.2)

6.3.5.1 Channel Selection

As for the *GDD Data*, according to some studies, the most notable electrode sites for gender discrimination by human are found to be located at the back of the brain (Inoue, Takuya. Et al. 2001)—see Figure 6.9 (a). To determine whether these are indeed the most important sites for gender discrimination, we selected different EEG data channels, designated numbers 1 to 8, that had been collected from the electrodes located at the back of the skull and used FMTSC to construct a classifier for the classification of the data. The experimental results are shown in Table 6.15.

Table 6.15 Classification Results from the GDD Dataset Collected From Different Combinations of Channels Located at the Back Part of the Brain

Channels in Back Brain	Accuracy	F-measure	Channel subset
1	63.33%	0.623	{ C_{B2} }
2	60%	0.598	{ P_7 O_Z }
3	63.64%	0.624	{ P_5 P_4 O_Z }
4	78.95%	0.77	{ P_Z P_{O2} C_{B2} O_2 }
5	76.47%	0.757	{ P_3 P_{O4} C_{B1} O_2 O_1}
6	**83.33%**	**0.822**	**{ P_{O5} P_{O3} C_{B1} P_{O4} C_{B2} P_{O6}}**
7	81.48%	0.805	{ P_z P_{O5} P_{OZ} P_{O4} C_{B1} C_{B2} O_1}

8	80%	0.78	{ P_5 P_1 P_2 P_4 P_{O5} P_{OZ} C_{B1} C_{B2} }

As shown in the table below, the combination of the set of the six channels, {$\mathbf{P_{O5}}$ $\mathbf{P_{O3}}$ $\mathbf{C_{B1}}$ $\mathbf{P_{O4}}$ $\mathbf{C_{B2}}$ $\mathbf{P_{O6}}$} yields the highest accuracy of 83.33%.

In order to determine whether the EEG signal obtained from channels located at the back part of the brain can better distinguish between genders when compared with the front part (see Figure 6.10), we performed additional experiments.

The average classification accuracy and the *F*-measure were computed and shown in Table 6.16. It is noted that the classification accuracy has dropped to around 57% with the *F*-measure being around 0.36. Hence, when comparing with these results with those obtained from the channels located in the back part of the brain, the classification accuracy went down by 23% from 83% and, even more significantly, the F-measure went down from 0.82 to 0.36. We therefore conclude that the channels in the front part of the brain are indeed less useful when the gender discrimination task is to be performed. These results are consistent with those reported in (Inoue, Takuya. Et al. 2001).

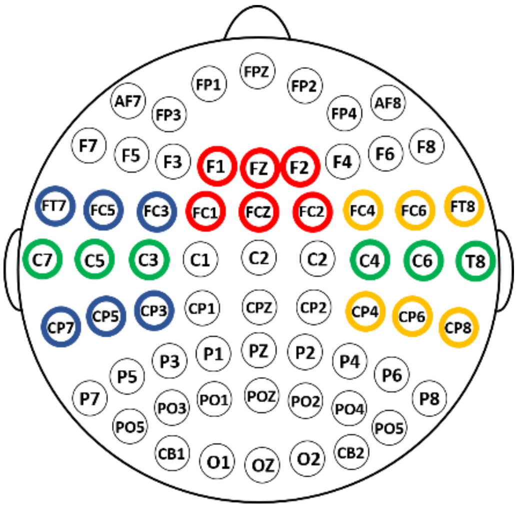

Figure 6.10 Combinations of Channels in the Front Area of Brain for Gender EEG Data

Table 6.16 Experimental Result with the GDD Dataset Using Channels Combinations from the Front Brain

Channels in the Front Brain	**Accuracy**	**F-measure**
$\{F_1, Fz, F_2, F_{C1}, F_{Cz}, F_{C2}\}$	53.85%	0.35
$\{F_{T7}, F_{C5}, F_{C3}, C_{P7}, C_{P5}, C_{P3}\}$	48%	0.324
$\{F_{T8}, F_{C6}, F_{C4}, C_{P8}, C_{P6}, C_{P4}\}$	60.87%	0.378
$\{C_7, C_5, C_3, C_4, C_6, T_8\}$	67.9%	0.371
Average	**57.66%**	**0.36**
Channels in the Back Brain	**83.33%**	**0.822**

For the *ALC Data*set, there have been some attempts to determine if some of the channels of EEG data are more useful in allowing us to differentiate between subjects classified as “alcoholics” and “control”. Some of the results are reported in (Andrzejak, R.G. et al. 2012). These results show that the EEG data collected from 7 of the 64 channels are more significant when it comes to differentiating alcoholics from the control group—see Figure 6.9(b).

In order to determine whether FMTSC can also identify these 7 channels as the more important and significant channels in differentiating between alcoholics and control, we performed a number of experiments to compare the classification accuracies for different datasets obtained through different combinations of channels including i) the data obtained from three channels located at the center part of the brain (3 channel-central), ii) the data obtained from four channels located at the back part of the brain (4 channel-back), and iii) the data obtained from the seven channels (7 channels) as shown in Figure 6.9(b).

The results from the experiments are shown in Table 6.17. It is clear that when the data from all the seven channels are selected, the classification accuracy is the highest at 93.18% with the F-measure being 0.931. These results confirm those reported in (Andrzejak, R.G. et al. 2012).

Table 6.17 Experiments with the ALC Dataset Using Different Combinations of Data Drawn from Different Channels

Data Set 2	3 channels-central	4 channels-back	7 channels
Accuracy	83.87%	86.67%	**93.18%**
F-measure	0.819	0.867	**0.931**

For further comparison, we randomly selected different subsets of data representing different combinations of channels (shown in Figure 6.11). The classification results presented in Table 6.18 confirm that the set of channels found in (Andrzejak, R.G. et al. 2012) are indeed the best set of channels to collect data from if the objective is to identify alcoholics. Using FMTSC, we determined the average classification to be 93.18% on average using the same seven channels as recommended in (Andrzejak, R.G. et al. 2012). The differences can be very significant if a different combination of channels is used. The classification accuracy can be 20% lower (73.49% as opposed to 93.18%) if the wrong combination of channels are selected. As for the *F*-measure, it can also be about 0.15 lower (0.78 as opposed to 0.93).

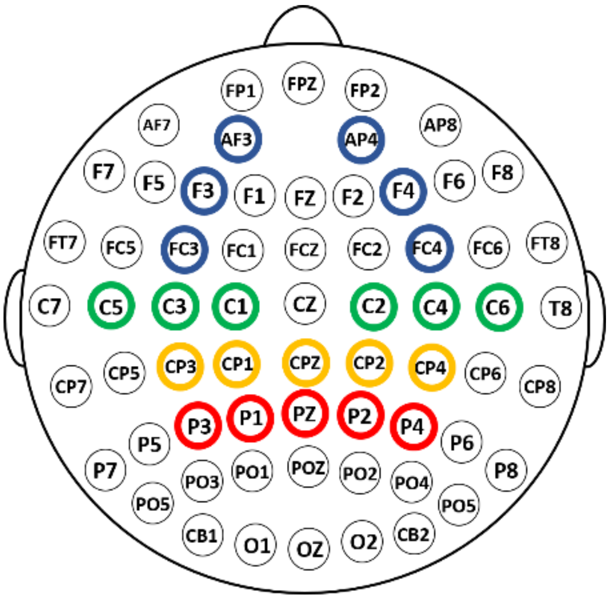

Figure 6.11 Other Channels Combinations for ALC Data Set

Table 6.18 Experiments with the ALC Dataset Using Other Channel Combination

Channels From Other Area	Accuracy	F-measure

$\{F_1,F_3,F_2,F_4,F_{C3},F_{C4}\}$	73.49%	0.78
$\{C_5,C_3,C_6,C_8,C_{P5},C_{P8}\}$	78.75%	0.81
$\{C_{P3}, C_{P1}, C_{PZ}, C_{P2}, C_{P4}\}$	83.64%	0.82
$\{P_5,P_3,P_1,P_Z,P_2,P_4,P_6\}$	82.37%	0.81
Average	**79.56%**	**0.805**
Proposed Channels	**93.18%**	**0.931**

As for the *EPL Data* collected from only two channels, Pz and Fz, the channel selection experiments were not performed. However, tests were still performed to determine how accurate FMTSC could be in extracting temporal patterns for classification. The results presented in Table 6.19 show that both the classification accuracy (92.86%) and the *F*-measure (0.925) were accurate. This confirms the effectiveness of FMTSC.

Table 6.19 Experiments with the EPL Dataset Using the Fz and Pz Channels

Dataset 3	**2 channels (Fz & Pz)**
Accuracy	92.86%
F-measure	0.925

6.3.5.2 Fuzzy Multi-Channel EEG Data Classification

To further evaluate the performance of the FMTSC Classifier, we performed additional experiments. The results are shown in Table 6.20.

Table 6.20 Comparison of Experimental Results from Different Algorithms

Dataset	**Evaluation**	**Original Data**	**CAR +SVM**	**ICA-SCC**	**No-Fuzzy**	**FMTSC**
GDD Data	**Accuracy**	45.16%	75%	56.76%	**85.71%**	**85.76%**
	F-measure	0.31	0.75	0.54	**0.813**	**0.846**
ALH Data	**Accuracy**	55.56%	78.86%	62.7%	**85.27%**	**89.33%**
	F-measure	0.49	0.767	0.55	**0.843**	**0.891**
EPL Data	**Accuracy**	38.09%	62.25%	52.94%	**84.47%**	**90.34%**
	F-measure	0.28	0.64	0.35	**0.894**	**0.9**

From the above table (for the *GDD Data*) we note that the classification accuracy was only 45.16% when the original raw data without any pre-processing with denoising algorithm were used. When the CAR+SVM algorithm was used, the classification accuracy was 75%, the *F*-measure being 0.75. The classification accuracy was found to be around 56.76% with an average *F*-measure of 0.54 when the ICA-SCC algorithm was used with only one single channel being considered by this algorithm. As for FMTSC, regardless of whether the data were just discretized or fuzzified, the patterns that it has managed to discover in the multi-channel EEG datasets seem to be quite meaningful. Based on this observation, the classification accuracy measure at 85.71% or 85.76% and the *F*-measure at 0.813 and 0.846 are all quite high.

Similar observations were noted for the *ALC Data*. This confirms the ability of FMTSC to identify even crisp patterns quite accurately. Even without the fuzzification of the EEG data, it is able to achieve a classification accuracy of 85.27% with a high *F*-measure of 0.843. This is a performance superior that is achieved by the CAR+SVM approach (it had performed with a classification accuracy of 78.86% and an *F*-measure value of 0.767). Relatively speaking, the classification accuracy of ICA-SCC is worse as it performed the classification task with an accuracy of 62.7% and an *F*-measure of 0.551. While fuzzifying data, the use of FMTSC can improve the classification accuracy to 89.33% and the *F*-measure value to 0.891.

As for the *EPL* dataset, the performance of FMTSC when the data are fuzzified are also the best; it has a classification accuracy as high as 90.34% and an *F*-measure value of 0.9. These experimental results show that the performance of FMTSC can be significantly better when the fuzzy intra-channel/inter-channel temporal patterns are considered for classification.

To make the results even more clear, the comparison results are summarized in Figure 6.12. The figure presents the average of classification accuracies for all classification methods; each group stands for one dataset. The five different bars in each group represent the classification accuracies while classifying EEG signals using i) the original EEG data, ii) the SVM classifier based on CAR pre-processed EEG signals, iii) the ICA-based Single Channel Classifier, iv) the proposed feature extraction method without fuzzification, and v) the proposed FMTSC classifier. The comparison results show that the last two bars have achieved the highest classification accuracies. Note that the bars show the classification accuracies achieved by proposed algorithm for extracting features from multi-channel EEG signals without and with a fuzzy system, respectively. In addition, FMTSC has sometimes been able to achieve better classification results than the method without fuzzification. We can conclude from this that, when FMTSC is used, the performance of classification result is generally comparable and, in some cases, even turns out to be the best.

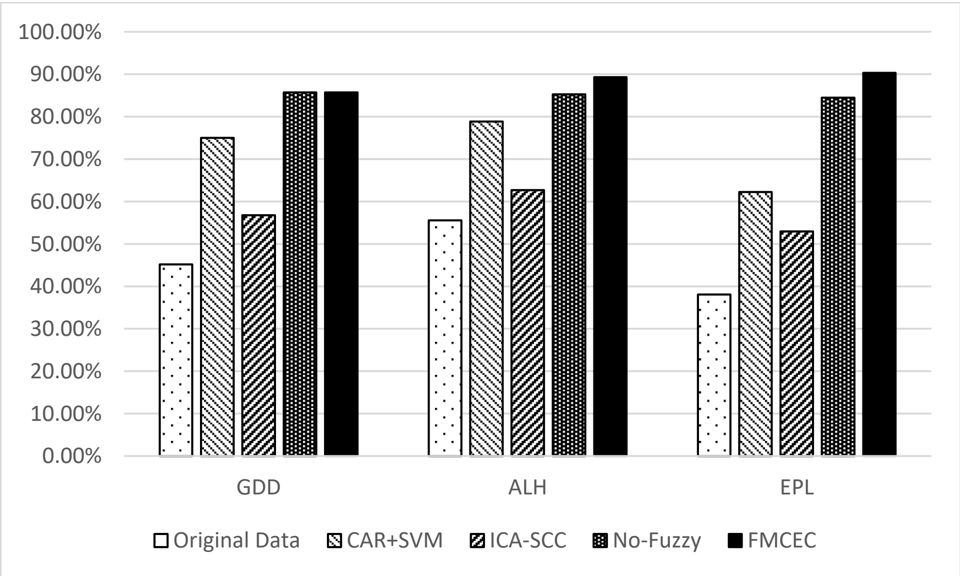

Figure 6.12 Comparison of Average Classification Accuracy from Different Classification Methods

In summary, we have successfully applied FMTSC to analyze several real cases related to EEG signal classification. We have shown that FMTSC can discover fuzzy temporal patterns between and within EEG signals collected from different channels using fuzzy measures. FMTSC tested with three real-world multi-channel EEG data. A comparison of the effectiveness of the proposed algorithm with that of the others shows that the experimental result using FMTSC is more realistic and superior. Experiments have shown that the proposed algorithm is able to achieve a classification accuracy that is higher than that achievable by other traditional methods. FMTSC is useful both from an algorithmic and an Electroencephalographic perspective. The following conclusions pertaining to the algorithm are worth mentioning:

i) Because a probabilistic measure is used for discovering fuzzy temporal patterns, the algorithm is able to work effectively even when the data covered are unequal and contain missing or erroneous values.

ii) The classification accuracy is improved when fuzzy measures are used. Specifically, with respect to application, it

   (a) is able to discover both intra-channel and inter-channel fuzzy temporal patterns among EEG signals,

   (b) is able to determine the most significant time interval for discovering fuzzy temporal patterns, and

   (c) can be used to solve a range of EEG problems beyond classification, e.g., the selection of significant channel combinations.

Chapter 7.

Conclusion

7.1 Summary

This thesis has presented a Multivariate Time Series Classifier (MTSC) capable of discovering patterns implicit in a multivariate time series and then classifying them. Unlike many existing methods, it is able to handle multivariate time series consisting of either continuous or discrete data, or both. Since MTSC can perform its tasks without requiring any special assumptions about data models, it is generic and application-independent. Given that it performs class analysis by discovering patterns within each time series independently of the others, it can also handle time series of different lengths. MTSC's performance has been tested using both artificial and real data. The results have shown that it is a promising algorithm for multivariate time series analysis.

After validating MTSC in analyses multivariate time series arising in a range of contexts, we have extended it into a Fuzzy Multivariate Time Series Classifier (FMTSC) using fuzzy set concepts.

In addition, in order to improve the classification performance, we have developed an unsupervised attribute clustering algorithm (UACA) for feature selection. The UACA can handle datasets without class label information. Since the proposed feature selection method is generic, it can perform its tasks without requiring any special assumptions concerning data models. In addition, the UACA does not need to discretize continuous data into discrete data before attribute clustering. For performance evaluation, it has been tested using several synthetic datasets and six UCI datasets drawn from the real world. Classification results obtained with different classifiers have shown that the algorithm holds high promise with regard to feature selection. The classification results arising from feature selection by UACA have even outperformed the classification results obtained by using whole datasets in some cases.

The analysis of multivariate time series has also been tested in various applications. In order to classify multi-channel EEG signals, FMTSC first discovers the fuzzy temporal patterns existing between and within EEG signals collected from different channels using fuzzy measures. The proposed algorithm has been tested against three sets of real-world multi-channel EEG datasets. A comparison of the effectiveness of the proposed algorithm with that of the others has shown that the experimental results on FMTSC use are more realistic and superior. It has been able to obtain a higher classification accuracy than that observed with other traditional methods. It is clear that FMTSC is useful both from an algorithmic and an Electroencephalographic perspective.

On the other hand, the FMTSC developed in this thesis has been motivated by the lack of an algorithm suitable for dealing effectively with multivariate time series data meant for stock analysis. Making use of the concept of fuzzy sets, FMTSC enables association analysis, clustering and creating portfolio determination stock analyses. It is able to discover intra- and inter-fuzzy temporal associations in multivariate time series data for stock analysis. It can discover patterns even when the lengths of the time series are different and even when the data are noisy and missing. Data from New

York Stock Exchange and Hong Kong Stock Exchange have been used to test the algorithm. Results have shown that the proposed algorithm holds high promise in the context of analyzing multivariate time series data. Experimental results have indicated that FMTSC can be an effective approach for stock analysis. Using FMTSC, it has been confirmed that the technique is able to discover several interesting associations implicit in stock data for predicting not only daily price movements but also semi-weekly, or weekly price movements. We have also been able to determine the number of days of look-back it takes to most accurately predict stock movements ahead of time. The clustering analysis using FMTSC has demonstrated its usefulness in the application of stock market analysis by predicting the stock price, improving the classification into industrial categories of stocks, and creating portfolios for helping investors in practice.

In summary, from an algorithmic point of view, three methodologies, MTSC, FMTSC and UACA are proposed. MTSC can discover intra-/inter-temporal patterns and classify MTS even when the data covered are unequal and contain missing or erroneous values. The classification accuracy is improved when fuzzy measures are used. And finally, an unsupervised feature selection method can reduce the size of feature space without considering class label information and setting the number of cluster manually.

From an applicational point of view, for stock market analysis, i) FMTSC can indeed discover significant associations in stock data and the discovered associations can predict not only daily price movements but also semi-weekly, or weekly price movements; ii) the proposed method's usefulness for conducting a stock market analysis to improve the industrial categories of stocks and create portfolios for helping investors; for EEG it i) can discover both intra-channel and inter-channel fuzzy temporal patterns among EEG signals, ii) can determine the most significant time interval for fuzzy temporal patterns, iii) is used to solve a range of EEG problems for not just the classification but also the selection of significant channel combinations.

7.2 Future Work

For future work, we intend to investigate into the possibility of improving MTSC in the aspect of algorithmic and in the aspect of application.

From the viewpoint of the algorithm:

i) In order to improve the classification performance, we intend to apply the "big data" concept to reduce the time consumption of the proposed algorithm. This will mean developing a distributed algorithm so that it can be executed on a massively parallel system.

ii) We need to extend the proposed algorithm into a multi-labels classification so that MTS belongs to different classes and overlapping classes can be discovered.

iii) We need to discover motifs or high order patterns instead of intra-/inter-temporal patterns.

iv) We need to analyze spatial and temporal data through addition spatial information for multivariate time series.

From the viewpoint of application:

i) We can include more applications to make comparisons, considering different types of multivariate time series have different temporal patterns.

ii) The proposed algorithm can be combined with graph mining, text mining to solve more practical problem, such as social network analysis.

Reference

Aggarwal C. C., On effective classification of strings with wavelets. In KDD ’02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 163–172, 2002.

Agrawal, R. and Srikant, R. Fast algorithms for mining association rules. In Proc. 20th int. conf. very large data bases, VLDB, vol. 1215, pp. 487-499, 1994.

Agrawal R., Ghosh S., Imielinski T., Iyer B., and Swami A., “An interval classifier for database mining applications”. In Proc. of the VLDB Conference. August 1992, pp. 560-573,1992.

Aguera P. E. , Jerbi K. , Caclin A. , and Bertrand O. , “ELAN: A Software Package for Analysis and Visualization of MEG, EEG, and LFP Signals,” Computational Intelligence and Neuroscience, 2011, no.5, pp.1-11.

Aha D.W., Kibler D., and Albert M. K. , “Instance-based learning algorithms. Machine learning” Machine Learning, vol. 6 no.1, pp.37-66, 1991.

Akin M., and Kiymik M. K. . “Application of periodogram and AR spectral analysis to EEG signals”. Journal of Medical Systems, 24(4), pp.247–256, 2000

Alhaddad M. J., and Abdulaziz K. “Common Average Reference (CAR) Improves P300 Speller”. International Journal of Engineering and Technology (IJET) – (2) 3, pp.453-464, 2012.

Al-Radaideh, Q. A., Assaf, A. A., & Alnagi, E. (2013). Predicting stock prices using data mining techniques. In The International Arab Conference on Information Technology (ACIT’2013).

Andrzejak R.G., Schindler K., Rummel C.. “Nonrandomness, nonlinear dependence, and nonstationarity of electroencephalographic recordings from epilepsy patients”. Phys. Rev. E, 86, 046206, 2012

Argiddi R. V. and Apte S. S., “Future trend prediction of Indian IT stock market using association rule mining of transaction data,” International Journal of Computer Application, vol. 39, pp. 30-34, 2012.

Asuncion A., and Newman D. J. , “UCI Machine Learning Repository”, School of Information and Computer Science, University of California, Irvine, CA, http://archive.ics.uci.edu/ml/, 2007.

Au W. H. ,Chan K. C. C , Wong A. K. C, and Wang Y. , “Attribute clustering for grouping, selection, and classification of gene expression data”. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB), vol. 2 no.2, pp.83-101, 2005.

Barrett, D.J. Change communication: Using strategic employee communication to facilitate major change. Corporate Communication: An International Journal. 7(4): 219-231, 2002

Batal, I., Sacchi, L., Bellazzi, R., and Hauskrecht, M.. Multivariate time series classification with temporal abstractions, 2009

Bell A.J. , and Sejnowski. T.J. An information-maximization approach to blind separation and blind deconvolution, Neural computation, 7(6), pp.1129–1159, 1995

Bellman R. E. , Adaptive Control Processes. Princeton University Press. 1961.

Beste A., Leventhal D., and Williams J., Lu Q. The Markowitz Model. Selecting an Efficient Investment Portfolio". Lafayette College, Mathematics REU Program, 2002

Bezdek, J. C. Pattern recognition with fuzzy objective function algorithms. New York: Plenum Press, 1998

Blankertz B., Lemm S., Treder M. S. , Haufe S. , and M üller K.-R. Single-trial analysis and classification of ERP components – a tutorial Neuroimage, vol. 56, no.2, 2011 pp.814-25.

Blekas K. , Fotiadis D. I., and Likas A. . Motif-based protein sequence classification using neural networks. Journal of Computational Biology, 12(1):64–82, 2005.

Bolotnikov, S. "Usage of Fuzzy Classification Algorithms in Brain-Computer Interfaces." Bachelor's thesis, Univeristy of Tartu, 2014.

Burke, W.W. and Litwin, G.H. . A causal model of organizational performance and change, Journal of Management. 18(3): 523-545, 1992.

Cai, D., He, X., Han, J.: Orthogonal Laplacianfaces for Face Recognition. IEEE Transactions on Image Processing. vol. 15 no.11, 2006, pp.3608-3614

Cai F,, Le-Khac N.A., Kechadi M.T.. "Clustering approaches for financial data analysis: a survey". Proceedings of the 8th International Conference on Data Mining. 2012. pp.105-111.

Cao D.,Ma N., Liu Y. , and Guo J. "A Feature Selection Algorithm for Continuous Attributes Based on the Information Entropy". Journal of Computational Information Systems, vol. 8, no. 4, pp.1467-1475, 2012.

Chan K. C. C. and Wong A. K. C., "A statistical technique for extracting classificatory knowledge from databases," in G. Piatetsky-Shapiro and W.J. Frawley (eds.) Knowledge Discovery in Databases. AAAI/MIT Press, pp. 107–123, 1991.

Chan, K.C.C., Wong, A.K.C. & Chiu, D.K.Y. Learning sequential patterns for probabilistic inductive prediction. IEEE Transactions on Systems, Man and Cybernetics. 24(10): p1532-1547. 1994

Chandrashekar G., and Sahin F., "A survey on feature selection methods". Computers and Electrical Engineering, vol.40, no.1, pp.16-28, 2014.

Chandwani D., Saluja M.S. , "Stock Direction Forecasting Techniques: An Empirical Study Combining Machine Learning System with Market Indicators in the Indian Context". International Journal of Computer Applications, vol 92, no.11, 2014.

Chang Y W, Hsieh C J, Chang K W, et al. Training and testing low-degree polynomial data mappings via linear SVM. The Journal of Machine Learning Research, 2010, 11: 1471-1490.

Cheng, C. H., Chen, T. L., & Chiang, C. H. (2006). Trend-weighted fuzzy time-series model for TAIEX forecasting. Lecture Note in Computer Science, 4234, 469–477.

Chiappa S. and Barber. D. Generative independent component analysis for EEG classification. In European Symposium on Artificial Neural Networks, pp.297–302, 2005

Chu H.H., Chen T.L., Cheng C.H., and Huang C.C.,. "Fuzzy dual-factor time-series for stock index forecasting". Expert systems with applications, vol. 36, no. 1, pp. 165-171, 2009.

Clampitt and Downs. Employee perceptions of the relationship between communication and productivity: a field study. Journal of Business Communication. vol. 30, no.1: pp.5-28, 1993

Cleary J. G. and Trigg. L. E. K*: An instance-based learner using and entropic distance measure. In International Conference on Machine Learning, pages 108–114, 1995.

Cohen, S., Intrator, N.: A Study Of Ensemble Of Hybrid Networks With Strong Regularization. In: 4th international conference on Multiple Classifier Systems.pp.227–235, 2003

Collingsworth, B., Menezes, R. and Martins, P. Assessing organizational stability via network analysis. In 2009 IEEE Symposium on Computational Intelligence for Financial Engineering Proceedings, pages 43-50. IEEE, April 2009.

Combarro E. F., Montanes E., Diaz I. , Ranilla J. , and Mones R. , "Introducing a family of linear measures for feature selection in text categorization". Knowledge and Data Engineering, IEEE Transactions on, vol.17, no.9, pp.1223-1232, 2005.

Corona C. and Randhawa, R. S. The auditors slippery slope: An analysis of reputational incentives. Management Science. vol.56 no.6 pp. 924-937, 2010.

Cortes C., VapnikV.: Support-Vector Networks. Machine Learning, 20 (3), 273-297,1995.

Coyle, D., McGinnity, T., & Prasad, G. . A multiclass brain-computer interface with sofnn-based prediction preprocessing. In Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on, pp. 3696-3703. IEEE, 2008.

Coyle, D., Prasad, G.&McGinnity, T., Faster self-organizing fuzzy neural network training and a hyperparameter analysis for a brain{computer interface. Systems, Man, and Cyber-netics, Part B: Cybernetics, IEEE Transactions on, 39(6):1458-1471, 2009.

De Andrés, J., Lorca, P., de Cos Juez, F. J., & Sánchez-Lasheras, F. "Bankruptcy forecasting: A hybrid approach using Fuzzy c-means clustering and Multivariate Adaptive Regression Splines (MARS)." Expert Systems with Applications vol. 38, no. 3, pp. 1866 - 1875, 2011.

Deng, Zhaohong, et al. "Minimax probability TSK fuzzy system classifier: a more transparent and highly interpretable classification model." Fuzzy Systems, IEEE Transactions on 23(4) pp: 813-826, 2005.

Devadoss A.V.,and LigoriT.A.A.. Stock prediction using artificial neural networks. International Journal of Data Mining Techniques and Applications, 2, 283-291, 2013.

M. Deshpande and G. Karypis. Evaluation of techniques for classifying biological sequences. In PAKDD '02: Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, pages 417–431, 2002.

Dimitrova, E.S., McGee, J.J., Laubenbacher R.C.: Discretization of Time Course Data. Journal of Computational Biology. 17(6),853–868. 2010.

Ding S., Zhu H., Jia W., Su C.: A Survey On Feature Extraction for Pattern Recognition, Artificial Intelligence Review, 37(3), 169-180, 2012

Dolphin, R.R. .The strategic role of investor relations. Corporate Communication: An International Journal. vol.9, no.1: pp.25-42, 2004

Down, C.W., Phillip, G.C. & Pfeiffer, A.L. Communication and organizational outcome. In G. Goldhaber & G. Barnett (Eds.), Handbook of Organizational Communication. Norwood, NJ: Ablex, 1998

Duan, W., Gu, B. & Whinston, A. Do online reviews matter? - An empirical investigation of panel data. Decision Support Systems. vol. 45 no.4: pp.1007-1016, 2008

Dunham, M. H. . Data Mining: Introductory and Advanced Topics. Prentice-Hall, 2002

Durbin R, , Eddy S. R., Krogh A,, and Mitchison G.. Chapter 3. Markov Chain and Hidden Markov Model. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids, pages 47–65. Cambridge University Press, 1998.

Duskin O. and Feitelson D. G. . Distinguishing humans from robots in web search logs: preliminary results using query rates and intervals. In WSCD09: Proceedings of the 2009 workshop on Web Search Click Data, pages 15–19, 2009.

Esmael B, Arnaout A, Fruhwirth R K, et al. Multivariate time series classification by combining trend-based and value-based approximation. Computational Science and Its Applications–ICCSA 2012. Springer Berlin Heidelberg, 2012: 392-403.

Fayyad U. and Uthurusamy R., "Data Mining and Knowledge Discovery in Databases", Communications of the ACM, vol.39, no.11, pp.24-27, 1996.

Fernandez, L.. A diversified portfolio: Joint management of non-renewable and renewable resources offshore. Resource and Energy Economics, 27, pp.65–82, 2005

Fingleton, B. Models of Category Counts, Cambridge, England: Cambridge University Press, 1984

Fiol-Roig, G., Miró-Julià, M., & Isern-Deyà, A. P. Applying data mining techniques to stock market analysis. In Trends in practical applications of agents and multiagent systems (pp. 519-527). Springer Berlin Heidelberg. 2010

Fu T., "A review on time series data mining," Eng. Appl. AI, vol. 24, pp. 164–181, 2011.

Fulcher, Ben D., and Nick S. Jones. Highly comparative feature-based time-series classification. Knowledge and Data Engineering, IEEE Transactions on26.12 pp. 3026-3037, 2014.

Garrett D., Peterson D. A., Anderson C. W. , Thaut, M. H. " Comparison of Linear, Nonlinear, and Feature Selection methods for EEG signal classification". IEEE transactions on neural system and rehabilitation engineering, vol.11 no. 2, 2003 pp141-144.

Gahirwal, M. "Inter Time Series Sales Forecasting". International Journal of Advanced Studies in Computers, Science and Engineering vol. 2 no.1 pp.55-66, 2013.

Gavrilov M. ,Anguelov D.,Indyk P., and Motwani R.. "Mining the Stock Market: Which measure is best?" Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, pp.487-496, 2000.

Gilbert, E. & Karahalios, K. Widespread worry and the stock market. The 4th International AAAI Conference on Weblogs and Social Media (ICWSM). 59-65, 2010

Godfrey M.D. , Granger C.W.J., Morgenstern O.. "The Random-Walk Hypothesis of Stock Market Behavior." Kyklos, vol. 17, no. 1 pp.1-30, 1964.

Gu Z.L., Shi L. C. and Lu B.L.. “Evidence of Rapid Gender Processing Revealed by ERSP”. Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE. IEEE, 2011: 7360-7363.

Guo, Chonghui, Hongfeng Jia, and Na Zhang. "Time series clustering based on ICA for stock data analysis." Wireless Communications, Networking and Mobile Computing, 2008. WiCOM'08. 4th International Conference on. IEEE, 2008.

Gupta R., Mittal A., and Singh K.. A time-series-based feature extraction approach for prediction of protein structural class. EURASIP J. Bioinformatics Syst. Biol., 2008:2:1–2:7, January 2008.

Gutierrez D., Garcia-Nocetti F., and Solano-Gonzalez J., “Classification of multichannel EEG data using length/energy transforms” In Proc. Computational Advances in Multi-Sensor Adaptive Processing, 1st IEEE International Workshop, 2005, pp.221-224.

Guyon I., Weston J., Barnhill S., and Vapnik V., “Gene selection for cancer classification using support vector machines”. Machine learning, vol 46. pp. 389-422, 2002.

Guyon I. , and Elisseeff A., “An introduction to variable and feature selection”. The Journal of Machine Learning Research, no.3, pp. 1157-1182, 2003.

Haberman, S. J. The analysis of residuals in cross-classified tables. Biometrics. 29: 205-220, 1973

Han J., Kamber M.. “Data Mining: Concepts and Techniques” (Second Edition). Publish by Elsevier Inc. 2007.

Hansen, G.S., & Wernerfelt, B. Determinants of firm performance: the relative importance of economic and organizational factors. Strategic Management Journal. 10(5): 399-411., 1989

Hargreaves, C., & Hao, Y. Prediction of stock performance using analytical techniques. Journal of Emerging Technologies in Web Intelligence. 5(2):136-142., 2013

Hayashi, A., Mizuhara, Y., Suematsu, N.: Embedding time series data for classification, In: MLDM’05, 356–365, (2005)

Haykin S.: Neural Networks: A Comprehensive Foundation. Prentice Hall, Upper Saddle River, (1998)

He H., Chen J., Jin H., and Chen S., “Stock trend analysis and trading strategy,” in Proceedings of Joint Conference on Information Sciences, 2006.

He X. and Niyogi P., "Locality Preserving Projections," Advances in Neural Information Processing Systems 16, Vancouver, British Columbia, Canada, 2003.

Higgins & Bannister (1992). How corporate communication of strategy affects share price. Long Range Planning. 25(3): 27-35.

Hong T.P., Wang, P. C. and Lee, Y. C. “An effective attribute clustering approach for feature selection and replacement”. Cybernetics and Systems: An International Journal, vol.40, no.8, pp. 657-669, 2009.

Hosni S.M., Gadallan, M. E. Bahgat, S. G. Wahab, M. S. A. “Classification of EEG Signals Using Different Feature Extraction Techniques for Mental-Task BCI”. In Proc. of ICCES '07, 2007, pp.220-226.

Huan L. and Hiroshi M.. Feature Selection for Knowledge Discovery and Data Mining, volume 454 of Springer International Series in Engineering and Computer Science. Springer, Berlin / Heidelberg, 1998.

Huang, C.F. Chen, Y. C. Chen, A. P. “An association mining method for time series and its application in the stock prices of TFT-LCD industry” Advances in Data Mining. Springer Berlin Heidelberg, pp.117-126, 2005.

Ijegwa, A.D. Rebecca, V.O. Olusegun, F. and Isaac, O.O. “A Predictive Stock Market Technical Analysis Using Fuzzy Logic”. Computer and Information Science, vol. 7 no.3, p1, 2014.

Takuya Inoue and Shigeo Abe. Fuzzy support vector machines for pattern classification. In Neural Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on, volume 2, pages 1449-1454. IEEE, 2001.

Jafarzadehm H.,Sadeghzadeh, M. “Improved Apriori Algorithm Using Fuzzy Logic”. International Journal of Advanced Research in Computer Science and Software Engineering. vol.4 no.6 pp.439-447, 2014.

John G.H., and Langley P. , “Estimating continuous distributions in Bayesian classifiers”. In Proceedings of the Eleventh conference on Uncertainty in artificial intelligence, Morgan Kaufmann Publishers Inc, August 1995, pp.338-345, 1995.

Kalpakis K., Gada D., Puttagunta, V., Distance measures for effective clustering of ARIMA time-series, Proceeding of the 2001 IEEE International Conference on Data Mining, San Jose, CA, Novermber 29- December 2, 2001. Pp.273-280

Kaplan, A.Y. and Shishkin, S. L. . “Application of the change-point analysis to the investigation of the brain's electrical activity”. Chapter 7 in: B.E.Brodsky, B.S.Darkhovsky. Nonparametric Statistcal Diagnosis: Problems and Methods. Kluwer Academic Publishers, Dordrecht (the Netherlands), 2000. P. 333-388.

Katz, P. The effects of group longevity on project communication and performance. Administrative Science Quarterly. vol. 27 no.1 pp: 81-104, 1982.

Katz R., and Tushman, M. Communication Patterns, Project Performance and Task Characteristics, Organizational Behavior and Human Performance no. 23, pp:139-162, 1979.

Keerthi, S. S., Shevade, S. K. , Bhattacharyya, C. , and Murthy, K. R. K.. Improvements to platt’s smo algorithm for svm classifier design. Neural Comput., 13:637–649, March 2001.

Kenji, K., and Rendell, L. A. . "The feature selection problem: Traditional methods and a new algorithm." In AAAI, vol. 2, pp. 129-134. 1992.

Keogh, E., and Kasetty, S.. On the need for time series data mining benchmarks: a survey and empirical demonstration. In KDD ’02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 102–111, 2002.

Kubler A., Nijboer F., Mellinger J., Vaughan T. M.,Pawelzik H. ,Schalk G. , McFarland D. J. , Birbaumer N. , and Wolpaw J. R. “Patients with ALS can use sensorimotor rhythms to ioerate a brain-computer interface” Neurology, 64 pp.1775-1777, 2005

Kumar M. and Thenmozhi M. . "Forecasting stock index movement: A comparison of support vector machines and random forest." Indian Institute of Capital Markets 9th Capital Markets Conference Paper. 2006.

Kumar M. and Kalia A., “Mining of emerging pattern: Discovering frequent itemsets in a stock data”. International Journal of Computer Technology and Applications, no. 2, pp. 3008-3014, 2011.

Kwon, O.-W. and Lee T.-W.. Phoneme recognition using ica-based feature extraction and transformation. Signal Process, 84, June 2004.

Lal, T. N., Schröder, M., Hinterberger, T., Weston, J., Bogdan, M., Birbaumer, N., & Schölkopf, B., Support vector channel selection in BCI.Biomedical Engineering, IEEE Transactions on, vol. 51 no.6, pp.1003-1010, 2004

Lameira P. Pair trading: Clustering based on principal component analysis. Work Project. NOVA-School of Business and Economics, 2015.

Larsen, B., Aone, C.: Fast And Effective Text Mining Using Linear-Time Document Clustering, In: 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, pp.16-22, 1999.

Laxman S. and Sastry P. S., “A survey of temporal data mining,” Sadhana, vol. 31, no. 2, pp. 173–198, 2006.

Lee, S., Kwon, D., and Lee, S. , Minimum distance queries for time series data. Journal of Systems and Software, vol. 69 no. 1-2 pp.105-113, 2004.

Lemieux V, Rahmdel P S, Walker R, et al. Clustering Techniques and Their Effect on Portfolio Formation and Risk Analysis, Proceedings of the International Workshop on Data Science for Macro-Modeling. ACM, 2014: 1-6.

Lewis D.D., Naive (bayes) at forty: The independence assumption in information retrieval. In ECML' 98: The 10th European Conference on Machine Learning, pages 4–15, 1998.

Li B., Chan, K.C.C., and Ou C.X., . "Public Sentiment Analysis in Twitter Data for Prediction of a Company's Stock Price Movements, " Proceedings of the IEEE International Conference on e-Business Engineering, ICEBE 2014:232-239, 2014

Li, C., Khan, L., Prabhakaran, B.: Feature Selection For Classification of Variable Length Multi-Attribute Motions. In: V.A. Petrushin, L. Khan (Eds.), Multimedia Data Mining and Knowledge Discovery, Springer-Verlag, chapter 7, (2007)

Li G. , Hu X. , Shen X. , Chen X. , and Li, Z. "A novel unsupervised feature selection method for bioinformatics datasets through feature clustering". In Granular Computing, 2008. GrC 2008. IEEE International Conference, IEEE, pp. 41-47, 2008.

Liao T.W. . "Clustering of time series data - a survey". Pattern Recognition no. 38, 2005, pp.1857-1874

Lofhede J. , Lofgren N. , Thordstein M. , Flisberg A. , Kjellmer I. , and Lindecrantz K. , "Classification of burst and suppression in the neonatal electroencephalogram". Journal of neural engineering, vol. 5 no.4, pp.402-410, 2008

Lotte F.. The use of fuzzy inference systems for classification in eeg-based brain-computer interfaces. In 3rd International Brain- Computer Interfaces Workshop and Training Course, 2006.

Lotte F., Congedo M., Lecuyer A., Lamarche F. , and Arnaldi B.. "A Review of Classification Algorithms for EEG-based Brain-Computer Interfaces" Journal of neural engineering, vol. 14 no.2, pp.1-13, 2007

Lu, H. Ha J. , and Feng L. , "Stock movement prediction and N-dimensional intertransaction association rules," Proceedings of SIGMOD Workshop Research Issues on Data Mining and Knowledge Discovery, pp. 12:1-12:7, 1998.

MA, P.C.H., Chan, K.C.C., Chiu, D.K.Y.: Clustering and re-clustering for pattern discovery in gene expression data. Journal of Bioinformatics and Computational Biology, 3(2), 281-201, 2005

MA, P.C.H., Chan, K.C.C., "Incremental fuzzy mining of gene expression data for gene function prediction", IEEE Transactions on Biomedical Engineering, vol.58, no.5, pp.1246-1252, 2011.

Markowitz H.. "Portfolio selection". Journal of Finance, vol.7, pp.77–91, 1952.

Marston, C. and Straker, M.. Investor relations: An European survey, Corporate Communication: An International Journal. vol.6 no.2 pp.82-93, 2001

McQueen J. “Some Methods for Classification and Analysis of Multivariate Observations,” Proc. Fifth Berkeley Symp. Math. Statistics and Probability, pp. 281-297, 1967.

Miller, G., Social scientists wade into the Tweet stream. Science. vol.333 no. 6051 pp.1814-1815, 2011

Mitra P., Murthy C. A. and Pal, S. K., “Unsupervised Feature Selection Using Feature Similarity”. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.24, no. 3, 2002.

Mueller A., Candrian G. , Kropotov J. D. , Ponomarev V. A., and Baschera G. M. . “Classification of ADHD patients on the basis of independent ERP components using a machine learning system”, Nonlinear Biomedical Physics, 4(Suppl 1), S1, 2010

Murph P.M., Aha, E.W.: UCI Repository of Machine Learning Databases,1999. url=http://archive.ics.uci.edu/ml/datasets/EMG+Physical+Action+Data+Set

Musetti, A. T. Y. "Clustering methods for financial time series.“ Swiss Federal Institute of Technology (2012).

Nair B.B., Dharini N. M. , and Mohandas V. P., “A Stock Market Trend Prediction System Using A Hybrid Decision Tree-neuro-fuzzy System”. Advances in Recent Technologies in Communication and Computing (ARTcom). pp. 381-385, 2010.

Nair B.B., Sai S.G. , Naveen A. N. , Lakshmi A. , Venkatesh G. S. , and Mohandas V. P. , “A GA-artificial neural network hybrid system for financial time series forecasting”. In V. Das, G. Thomas, and F. Gaol (Eds.), Lecture notes in computer science-communications in computer and information science vol.147, pp. 499-506, 2011.

Nair B. B., Patturajan M., Mohandas V. P., and Sreenivasan R. R. “Predicting the BSE Sensex: Performance comparison of adaptive linear element, feed forward, and time delay neural networks”. International Conference on Power, Signals, Controls and Computation pp.1-5, 2012.

Nanda S. R., Mahanty B., Tiwari M. K. “Clustering Indian stock market data for portfolio management”. Expert Systems with Applications, vol. 37, no. 12, pp.8793-8798, 2010.

Nandish M., Stafford M. , Hemanth K. P. , and Faizan A. , “Feature extraction and classification of EEG signal using neural network based techniques”, International Journal of Engineering and Innovative Technology (IJEIT), 2(4),pp. 1-5, October, 2012.

Nicholis S. C., and Sumpter D. J. T. . “A dynamical approach to stock market fluctuation”。 International Journal of Bifurcation and Chaos, vol.21 no.12, pp.3557 -564, 2011.

Niedermeyer E., and Silva F. H. L., “Electroencephalography: Basic Principles, Clinical Applications, and Related Field,” in Lippincott Williams & Wilkins, 2005

Oh, K. J., Kim, T. Y., and Min, S. Using genetic algorithm to support portfolio optimization for index fund management. Expert Systems with Applications, 28, 371–379, 2005

Olszewski, R.T..ECG Database,2004 url = http://www.cs.cmu.edu/~bobski/

Ostermark, R. A fuzzy control model (FCM) for dynamic portfolio management. Fuzzy Sets and Systems, 78, 243–254, 1996

Osteyee D.B. And Good I.J., “Information, Weight of Evidence, the Singularity between Probability Measures and Signal Detection”. Berlin,Germany:Springer-Verlag, vol.376, 1974.

Ou P., Wang H. . “Prediction of Stock Market Index Movement by Ten Data Mining Techniques.” Modern Applied Science. vol.3, no.12, 2009.

Ozkan, I., Türksen, I. B., & Canpolat, N. A currency crisis and its perception with fuzzy C-means. Information Sciences, 178, 1923–1934, 2008

Pathak, K., Mane, S. & Srivastava, J. Who thinks who knows who? Socio-cognitive analysis of email networks. Proceedings of the 6th IEEE International Conference on Data Mining. 18-22 December, Hong Kong, China. 466-477, 2006.

Powers, D.M.W. . "Evaluation: From Precision, Recall and F-Measure to Roc, Informedness, Markedness & Correlation". Journal of Machine Learning Technologies 2 (1): 37–63, February 27, 2011.

Quinlan, J.R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993

Ramoni M.,Sebastiani P. , Cohen P. , “Multivariate clustering by dynamics”, In Proc. of AAAI-2000, pp.633-638, 2000.

Rani, S., and Sikka, G. Recent techniques of clustering of time series data: a survey. International Journal of Computer Applications, Volume 52 – NO.15, August 2012.

Ravisankar P. , Ravi V., Rao G. R. , and Bose I. , “Detection of financial statement fraud and feature selection using data mining techniques”. Decision Support Systems, vol. 50, no. 2, pp.491-500, 2011.

Reshef D.N., Reshef Y. A., Finucane H. K., Grossman S. R., McVean G. , Turnbaugh P. J. , Lander E. S. , Mitzenmacher M. , and Sabeti P. C. , “Detecting novel associations in large datasets”. Science, vol. 334, no. 6062, pp.1518-1524, 2011.

Reshef, D., Reshef, Y., Mitzenmacher, M., and Sabeti, P.. Equitability analysis of the maximal information coefficient, with comparisons. arXiv preprint arXiv: 1301.6314, 2013.

Rodrgueza, J.J., Alonsob, C.J., Maestro, J.A.: Support Vector Machines of Interval-Based Features For Time Series Classification, Knowledge-Based Systems. 18, pp.171–178, 2005

Rosenberg, A.. Automatic detection and classification of prosodic events. Thesis Columbia University, 2009.

Roy P., Kumar R., Sharma S., “A Fuzzy Logic Model to forecast stock market momentum” International Journal of Advanced Engineering Research and Studies. vol. 98 no. 101, 2015

Royo R.C., Guijarro F., Michniuck K., “Stock market trading rule based on pattern recognition and technical analysis: forecasting the DJIA index with intraday data.” Expert System with Application. vol.42, no.14, pp. 5963-5975, 2015.

Saigo H., Vert J.-P., Ueda N. , and Akutsu T.. Protein homology detection using string alignment kernels. Bioinformatics, 20(11):1682–1689, 2004.

Sands, T. M., Tayal, D., Morris, M. E., and Monteiro, S. T. “Robust stock value prediction using support vector machines with particle swarm optimization” Evolutionary Computation (CEC), IEEE Congress on. IEEE, pp.3327-3331, 2015.

Sawant A A, and Chawan P M.. Study of Data Mining Techniques Used for Financial Data Analysis. International Journal of Engineering Science and Innovative Technology. vol.2 no.3 pp.503-509, 2013

She, R., Chen, F., Wang, K., Ester, M., Gardy, J. L., & Brinkman, F. S.,. Frequent-subsequence-based prediction of outer membrane proteins. In KDD ’03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 436–445, 2003.

Spiegel S., Kunegis J., and Li F. . Hydra: a hybrid recommender system [cross-linked rating and content information]. In CIKM-09: Proceeding of the 1st ACM international workshop on Complex networks meet information and knowledge management (CNIKM-09), pages 75–80, New York, NY, USA, 2009. ACM.

Spiegel, S., Jain, B. J., De Luca, E., and Albayrak, S.. Pattern recognition in multivariate time series." PIKM 2011, 2011, pp.27-33.

Spiegel, S., Clausen, J., Albayrak, S., and Kunegis, J., Link prediction on evolving data using tensor factorization. In PAKDD-2011: Proceeding of ACM

International Workshop on Behavior Informatics (BI-2011), New York, NY, USA, 2011. ACM.

Srisawat A., “An application of association rule mining based on stock market”. The 3rd International Conference on Data Mining and Intelligent Information Technology Applications (ICMIA). pp.259-262, 2011.

Srivastava P. K. , Desai D. K., Nandi S. , and Lynn, A. M. HMM-ModE-Improved classification using pro- file hidden Markov models by optimising the discrimination threshold and modifying emission probabilities with negative training sequences. BMC Bioinformatics, 8(104), 2007.

Subasi A. , Alkan A. , Koklukaya E. , and Kiymik M. K. . “Wavelet neural network classification of EEG signals by using AR model with MLE preprocessing”. Neural Networks, 18, pp.985-997, 2005

Sudjianto A. and Wasserman G.. A nonlinear extension of principal component analysis for clustering and spatial differentiation. IIE transactions, 28(12):1023-1028, 1996.

Suleiman A. B. R. , Fatehi T. A. H. , "Feature extraction techniques of EEG signal for BCI applications", Faculty of Computer and Information Engineering Deparment College of Electronics Engineering, University of Mosul, Iraq, 2007

Tan P. , Steinbach M. , and Kumar V., “Introduction to Data Mining” Addison-Wesley, 2005.

Tan P-N. , Steinbach M. , and Kumar V., Introduction to Data Mining, Addison Wesley, 2006, pp. 150-172.

Tang D. , Wang M. , Zheng W. , and Wang H. , “RapidMic: Rapid Computation of the Maximal Information Coefficient”. Evolutionary bioinformatics online, vol.10, pp.11-16, 2014

Theodorus Z. , and Dimitrus K. , “Short Term Prediction of Foreign Exchange Rates with a Neural-Network Based Ensemble of Financial Technical Indicators”. International Journal on Artificial Intelligence Tools. vol. 22 no. 3, pp.220-241, 2013.

Tian L. , Erdogmus D. , Adami A. , Pavel M. , and Mathan S. , “Salient EEG. Channel selection in brain computer interfaces by mutual information maximization”. In Proc. IEEE-EMBS, pp. 7064–7067, 2006

Ting J. , Fu T.-C. , and Chung F. -L , “Mining of stock data: intra-and inter-stock pattern associative classification”. Threshold, vol. 5, no. 100, pp. 95-99, 2006.

Todd, W. "Time-series clustering and association analysis of financial data." Elect. Eng. Res. Lab., Univ. Texas, Austin, Tech. Rep. CS 8980, 2002.

Topaloglou, N., Vladimirou, H., and Zenios, S. A., A dynamic stochastic programming model for international portfolio management. European Journal of Operational Research, no.185, pp.1501–1524, 2008

Tung A.K.H. , Lu. H. ,Han J. and Feng L. , “Efficient mining of intertransaction association rules”. IEEE Transactions on Knowledge and Data Enginning, vol. 15, pp. 43-56, 2003.

Vamsidhar E., Varma K.V.S.R.P., Rao P.S. , and Satapati R. . Prediction of rainfall using backpropagation neural network model. International Journal on Computer Science and Engineering. vol.2, no.4 pp.1119-1121, 2010

Wang, L. Mehrabi,M.G. Kannatey-AsibuJr .,E. , Hidden Markov model-based wear monitoring in turning, J. Manufacturing Sci. Eng. 124 (2002) p651–658

Wang X. Z., Smith K. and Hyndman, R. “Characteristic-Based Clustering for Time Series Data,” Journal on Data Mining and Knowledge Discovery, vol. 13, no. 3, pp. 335-364, 2006.

Wang, Y. and Wong, A.K.C. From association to classification: Inference using weight of evidence. IEEE Transactions on Knowledge and Data Engineering. 15(3), 2003, pp.764-767.

Wang Y., Choi I.-C. “Market Index and Stock Price Direction Prediction using Machine Learning Techniques: An empirical study on the KOSPI and HSI”. Science Direct Elsevier 2013.

Webb G. I. , “Decision tree grafting from the all-tests-but-one partition”. In IJCAI, Jyly 1999, pp. 702-707, 1999.

Wei L. and Keogh, E. . Semi-supervised time series classification. In KDD ’06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 748–753, 2006.

Weigend A. Data mining in finance: Report from the post-nncm-96 workshop on teaching computer intensive methods for financial modeling and data analysis[C]//Fourth International Conference on Neural Networks in the Capital Markets, NNCM. 1997, 1996: 399-411.

Weng, X.: Classification of Multivariate Time Series Using Supervised Neighborhood Preserving Embedding. In: 25th Control and Decision Conference (CCDC), Guiyang, pp.957-961, 2013

Weng, X., Shen, J.: Classification of Multivariate Time Series Using Locality preserving projection. Knowledge-based System. 21(7), 581-587 (2008)

Wolpaw, J. R. Birbaumer, N. McFarland, D. J.Pfurtscheller, G. and Vaughan, T. M. “Brain-computer interface for communication and control,” Clinical Neurophysiology, 113(6), pp.767-791, 2002

Wong, A.K.C., Wang, C.C.: DECA–A Discrete-Valued Ensemble Clustering Algorithm. IEEE Trans on Pattern Analysis and Machine Intelligence, PAMI-1(4), 342-349, 1979

Wong, D.K. Guimaraes M. P., Uy, E. T., Suppes P. .“Classification of individual trials based on the best independent component of EEG-recorded sentences”. Neurocomputing, 61, pp.479-484, 2004

Wong, D. K. Guimaraes, M. P. and Uy, E. T. “Multichannel Classification of Single EEG Trials with Independent Component Analysis”, Advances in Neural Netowrks – ISNN, pp.541-547, 2006

Wu D. , Liu Y., Gao G., C3M: A Classification Model for Multivariate Motion Time Series, Proceedings of 2009 World Congress on Computer Science and Information Engineering, 483-489, 2009.

Wu K P, Wu Y P, Lee H M. Stock Trend Prediction by Using K-Means and AprioriAll Algorithm for Sequential Chart Pattern Mining[J]. Journal of Information Science and Engineering, 2014, 30(3): 653-667.

Wu, M-C. Lin S-Y. and Lin, C-H. (2006). An effective application of decision tree to stock trading. Expert Systems with Applications.131:270–274

Xing, Z., Pei, J., Keogh, E.,: A Brief Survey On Sequence Classification. ACM SIGKDD Explorations Newsletter, ACM Press, New York, pp.40-48 (2010)

“Yahoo! Financial.” http://finance.yahoo.com.”

Yakhnenko O. , Silvescu A., and Honavar V. . Discriminatively trained markov model for sequence classification. In ICDM ’05: Proceedings of the Fifth IEEE International Conference on Data Mining, pages 498–505, 2005.

Yang, K. and Shahabi, C. (2004). A pca-based similarity measure for multivariate time series. In ACM International Workshop On Multimedia Databases: Proceedings of the 2 nd ACM international workshop on Multimedia databases, volume 13, pages 65-74.

Yates, K., Effective employee communication linked to greater shareholder returns, Watson Wyatt Study finds - November 2003. (http://www.watsonwyatt.com/render.asp?catid=1&id=12092).

Ye, J. Janardan, R. and Li. Q. Gpca: an efficient dimension reduction scheme for image compression and retrieval. In KDD ’04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 354–363, New York, NY, USA, 2004. ACM Press.

Zeithaml, V.A., Berry, L.L., & Parasuraman, A. (1988). Communication and control processes in the delivery of service quality. Journal of Marketing. 52(2): 35-48.

Zhang, H., Ho, T., Huang, W.: Blind Feature Extraction for Time Series Classification Using Haar Wavelet Transform, In: 2nd International Conference On Advances In Neural Network, ISNN'05, pp.605–610 2005.

Zhao, X. Deng, W. and Shi, Y. "Feature Selection with Attributes Clustering by Maximal Information Coefficient". Procedia Computer Science, no. 17, pp.70-79, 2013.

Zhao Y. ,Hong W. , Xu, Y. and Zhang, T. . "Multichannel epileptic EEG classification using quaternions and neural network". In Proc. PCSPA, 2010, pp.568-571.

Zhu D., Wu C., Qin W.: Multivariate statistic analysis and software SAS. Southeast University Press, Nanjing, 1999.

Zhuang D.E.H., Li G.C.L. and Wong, A.K.C. "Discovery of temporal associations in multivariate time series." IEEE Transactions on Knowledge and Data Engineering 26.12,2014. pp.2969-2982.