

bradscholars

Computational Approaches for Time Series Analysis and Prediction. Data-Driven Methods for Pseudo-Periodical Sequences.

Item Type	Thesis
Authors	Lan, Yang
Rights	<p>http://creativecommons.org/licenses/by-nc-nd/3.0/>
The University of Bradford theses are licenced under a http://creativecommons.org/licenses/by-nc-nd/3.0/>Creative Commons Licence.</p>
Download date	2026-05-09 20:32:53
Link to Item	https://bradscholars.brad.ac.uk/handle/10454/4317.2



University of Bradford eThesis

This thesis is hosted in [Bradford Scholars](#) – The University of Bradford Open Access repository. Visit the repository for full metadata or to contact the repository team



© University of Bradford. This work is licenced for reuse under a [Creative Commons Licence](#).

Computational Approaches for Time Series Analysis and Prediction

Data-Driven Methods for Pseudo-Periodical Sequences

Yang Lan

Submitted for the Degree
of Doctor of Philosophy

Department of Computing
School of Computing, Informatics & Media
University of Bradford

2009

Acknowledgements

Writing a thesis is an arduous process, and no any author can produce a good one only by himself. I would like to thank deeply all of those who helped me with this thesis.

As always, my sincerest and special thanks go to my supervisor, Dr. Daniel Neagu, for his dedication to my studies. Not only his guidance and encouragement, but also his friendship, advices and discussions marked my way during my research years in Bradford.

To my dear parents, Shuangjun Lan and Yingzhang Yang, who with love supported me during studies and encouraged me to give my best, I can say that I am the proudest member of our family in this world.

To my sweet wife, Lina Wang, who has always been my strongest support throughout years: without her faith and patience, I could have not finished this thesis.

28 July 2009

Bradford, UK

Abstract

Time series data mining is one branch of data mining. Time series analysis and prediction have always played an important role in human activities and natural sciences. A *Pseudo-Periodical* time series has a complex structure, with fluctuations and frequencies of the times series changing over time. Currently, *Pseudo-Periodicity* of time series brings new properties and challenges to time series analysis and prediction.

This thesis proposes two original computational approaches for time series analysis and prediction: *Moving Average of n^{th} -order Difference (MANoD)* and *Series Features Extraction (SFE)*. Based on data-driven methods, the two original approaches open new insights in time series analysis and prediction contributing with new feature detection techniques. The proposed algorithms can reveal hidden patterns based on the characteristics of time series, and they can be applied for predicting forthcoming events.

This thesis also presents the evaluation results of proposed algorithms on various pseudo-periodical time series, and compares the predicting results with classical time series prediction methods. The results of the original approaches applied to real world and synthetic time series are very good and show that the contributions open promising research directions.

Keywords: Time Series, Time Series Analysis and Prediction, n^{th} -order Difference, Similarity, Feature Extraction

Declaration

Some parts of the work presented in this thesis have been published in the following articles:

R Zhang, S Zhang, **Y Lan** and J Jiang, *Network Anomaly Detection Using One Class Support Vector Machine*, IAENG International Conference on Data Mining and Applications (ICDMA), March 2008, Newswood Limited, volume I, pp.452-456, ISBN: 978-988-98671-8-8, Hong Kong, China.

Y Lan and D Neagu, *A New Time Series Prediction Algorithm based on Moving Average of n^{th} -order Difference*, The Sixth International Conference on Machine Learning and Applications (ICMLA), December 2007, IEEE Computer Society Press, pp.248-253, ISBN: 0-7695-3069-9, Cincinnati, Ohio, USA.

Y Lan and D Neagu, *Applications of Moving Average of n^{th} -order Difference Algorithm for Time Series Prediction*, The Third International Conference on Advanced Data Mining and Applications (ADMA), August 2007, Springer Verlag LNCS, volume 4632/2007, pp.264-275, ISBN: 978-3-540-73870-1, Harbin, China.

Y Lan and D Neagu, *A New Algorithm Based on the Average Sum of n^{th} -order Difference for Time Series Prediction*, The Sixth annual UK Workshop on Computational Intelligence (UKCI), September 2006, Proceedings pp.183-189, University of Leeds.

V Zharkova, S Zarkov and **Y Lan**, *Active Longitudes and Latitudes in Sunspot and Plage Occurrences in the Cycle 23 and their Magnetic Field Variations*, SCOSTEP Eleventh Quadrennial Solar Terrestrial Physics Symposium "Sun, Space Physics and Climate", March 2006, Rio de Janeiro, Brazil.

S Zharkov and **Y Lan**, *Data Analysis in Solar Feature Catalogues*, Proceedings of the Sixth Informatics Workshop, March 2005, pp.205-207, ISBN: 1-85143-2205, University of Bradford.

Table of Contents

Acknowledgements	I
Abstract	II
Declaration	III
1 Introduction	1
1.1 Data Mining on Time Series	2
1.2 Motivation	4
1.3 Purpose of Research	8
1.4 Methodology	10
1.5 Overview of This Thesis	13
1.6 Summary	14
2 Time Series Analysis and Prediction	16
2.1 What a Time Series is?	17
2.1.1 Examples of Time Series	18
2.1.1.1 Time Series in Economy and Finances	18
2.1.1.2 Time Series in Nature	19
2.1.1.3 Time Series in Demography	20

2.1.1.4	Time Series in Production Process Control . . .	21
2.1.1.5	Binary Equivalent Time Series	22
2.1.1.6	Points Process Time Series	23
2.2	Pseudo-Periodical Time Series	24
2.2.1	Earthquakes Time Series	25
2.2.2	Flu Trends Time Series	27
2.2.3	Nile River Flooding Time Series	28
2.2.4	Sunspot Number (Monthly Average) Time Series	29
2.2.5	Synthetic Pseudo-Periodical Time Series	31
2.3	Time Series Analysis	32
2.4	Time Series Prediction	34
2.4.1	Linear Regression Method	36
2.4.2	Auto-Regression Moving Average Method	37
2.5	Summary	39

3 A Time Series Prediction Algorithm based on Moving Average of n^{th} -order Difference **41**

3.1	Introduction	42
3.2	n^{th} -order Difference	44
3.3	Moving Average of Data Series	46
3.4	The Prediction Algorithm	47

3.4.1	Implementation of the Prediction Algorithm	47
3.4.2	Finding Suitable Parameters for Increasing Precision of the Prediction Algorithm	56
3.5	Case Studies	60
3.6	Summary	64
4	A Time Series Prediction Algorithm based on Series Features Extraction	67
4.1	Introduction	68
4.1.1	Epistemology	68
4.1.2	A Priori and A Posteriori Knowledge	69
4.1.3	Features, Patterns and Model	70
4.1.4	The Methodology of Series Features Extraction Ap- proaches	72
4.2	Time Series Data Classification based on a Combination Rule of Successive Neighbouring Data Points	74
4.2.1	Data Classification for a Generic Data Sequence Set . .	74
4.2.2	Combination of Time Series Data Points	75
4.2.3	Optimizing the Categorization	81
4.3	The Approach of Series Features Extraction Algorithm for Time Series Analysis and Prediction	98

4.3.1	Eigenvector	98
4.3.2	Transformation of Time Series	102
4.3.3	Feature Extraction and Pattern Recognition from His- torical Values	104
4.3.4	Filtering the Returned Matches	107
4.3.5	Computing the Prediction	112
4.4	Case Studies	112
4.5	Summary	116
5	Evaluation	117
5.1	Implementation of Classical Methods and Proposed Prediction Algorithms	118
5.2	Evaluation of Prediction Results	120
5.2.1	Testing Time Series Case Studies for Evaluation	120
5.2.2	Prediction Results Comparison	121
5.2.2.1	Measures for Results Evaluation	121
5.2.2.2	Linear Regression: Prediction Results	122
5.2.2.3	Auto-Regression Moving Average: Prediction Results	125
5.2.2.4	Moving Average of n^{th} -order Difference: Pre- diction Results	128

5.2.2.5	Series Features Extraction: Prediction Results	131
5.2.2.6	Results Comparison	133
5.3	Summary	138
6	Conclusions	140
6.1	Summary of Research	141
6.2	Original Contributions	143
6.3	Future Work	144
6.4	Final Remarks	146
Appendix A		147
Appendix B		151
References		154

List of Figures

1.1	Methodology for Time Series Data Mining (From Original Database to Integrated System Deployment)	12
2.1	Beveridge Wheat Price Index Time Series (Source: Time Series Data Library [Hyndman, 2009]) X Coordinate Axis Lists the Time Intervals and Y Coordinate Axis Illustrates the Wheat Price Index Value.	18
2.2	Monthly Precipitation Time Series in Southwestern Mountain, West Virginia, U.S. (Source: Time Series Data Library [Hyndman, 2009]) X Coordinate Axis Lists the Index of Time Intervals for Time Series and Y Coordinate Axis Illustrates the Monthly Precipitation.	19
2.3	Rates of Proportion of Church of England Marriages Time Series, England, (Source: Time Series Data Library [Hyndman, 2009]) X Coordinate Axis Lists the Time Intervals of Time Series and Y Coordinates Axis Illustrates the Values of Rates of Proportion.	20

2.4	Production Process Control Time Series from Ishikawa (Source: Time Series Data Library [Hyndman, 2009]) X Coordinate Axis Lists the Index of Time Intervals for the Time Series and Y Coordinate Axis Illustrates the values of Production Process Control.	21
2.5	An Example for (generated) Realization of a Binary Equivalent (Processes) Time Series. X Coordinate Axis Lists the Index of Time Intervals for the Time Series and Y Coordinate Axis Illustrates the values (0 or 1) of Binary Equivalent Time Series.	22
2.6	An Example for (generated) Realization of a Point Process Time Series. X Coordinate Axis Lists the Index of Time Intervals for the Time Series and Y Coordinate Axis Illustrates the Events Series (with the red “×”).	23
2.7	An Example of Earthquakes Time Series Data Set (Richter Magnitude Scale (RMS) by Index); X Coordinate Axis Lists the Index of Time Intervals for the Time Series and Y Coordinate Axis Illustrates the Values of RMS.	26
2.8	An Example of Flu Trends (Influenza Rates) in United States Time Series Data Set; X Coordinate Axis Lists the Index of Time Intervals and Y Coordinate Axis Illustrates the Influenza Rates Values.	28

2.9	An Example of Nile River Flow Time Series Data Set; X Coordinate Axis Lists the Index of Time Intervals and Y Coordinate Axis Illustrates the Flow Values of Nile River.	29
2.10	An Example of (Monthly Average) Sunspot Number Time Series Data Set; X Coordinate Axis Lists the Index of Time Intervals and Y Coordinate Axis Illustrates the Monthly Average Sunspot Number Values.	30
2.11	An Example of Synthetic Pseudo-Periodical Time Series Data Set, X Coordinate Axis Lists the Values of Variable \bar{t} (see eq.(2.5) with 100000 values) and Y Coordinate Axis Illustrates the values of \bar{y} (see eq.(2.5) with 100000 values).	31
3.1	The Monthly Average Values of Sunspot Number Time Series for 600 Months; X Coordinate Axis Lists the Index of Time Intervals (600 Months) and Y Coordinate Axis Illustrates the Values of Monthly Average Sunspot Number.	49
3.2	First-Order Difference (D_m^1) of Monthly Average Values of Sunspot Number Time Series; X Coordinate Axis Lists the Values of m with 600 Samples and Y Coordinate Axis Illustrates the First-Order Difference Values for D_m^1	49

3.3	The Moving Average (E_m^1) of First-order Difference (D_m^1) of Monthly Average Values of Sunspot Number Time Series; X Coordinate Axis Lists the Values of m with 600 Samples and Y Coordinate Axis Illustrates the Moving Average Values for E_m^1	49
3.4	A Map of Moving Average of n^{th} -order Difference's Limit (see eq.(3.19)) for Sunspot Number Time Series Data Set; X Coordinate Axis Lists the Variable $m \in [1, 100]$, Y Coordinate Axis Illustrates the Variable $n \in [1, 100]$ and Z Coordinate Axis Shows the Values of Moving Average.	54
3.5	Analysis and Prediction for Error (ε) with Artificial Neural Network with E_m^n (1000 samples and $n = 10$) and E_{m+1}^n (1000 samples and $n = 10$). There is a linear correlation relationship between the two variables, E_m^n and E_{m+1}^n , therefore, the error (ε) for next term prediction can be approximated by ANN. . .	57
3.6	The Manhattan Distance Value Series, Θ_m^n , where $n = 1$ and X Coordinate Axis Lists the Values of $m \in [1, 600]$ and Y Coordinate Axis Illustrates the Values for Distance.	59
3.7	The Value Map of Matrix: Θ for Sunspot Number Time Series Data Set (where $m \in [1, 500]$ and $n \in [1, 20]$)	60

3.8	The Initial Monthly Average Sunspot Number Time Series and Prediction Results by Algorithm MANoD (where $m = 12$ and $n = 12$); X Coordinate Axis Lists the Index of Time Intervals (1200 Values) and Y Coordinate Axis Illustrates the Original (in blue with 1200 values) and Prediction Values (in purple with 600 values).	61
3.9	Sunspot Number Time Series Prediction Errors (600 data values) by MANoD; X Coordinate Axis Lists the Index of Time Intervals (600 values) and Y Coordinate Axis Illustrates the Prediction Error Values ($\ Prediction - Original\ $) for 600 Values.	62
3.10	The Initial Global Earthquakes' Richter Magnitude Scale (RMS) Time Series and Prediction Results by Algorithm MANoD; X Coordinate Axis Lists the Index of Time Intervals (1351 Values) and Y Coordinate Axis Illustrates the Original (in blue with 1351 values) and Prediction Values (in purple with 676 values).	62
3.11	Global Earthquakes' Richter Magnitude Scale Time Series Prediction Errors by MANoD; X Coordinate Axis Lists the Index of Time Intervals (676 values) and Y Coordinate Axis Illustrates the Prediction Error Values ($\ Prediction - Original\ $) for 676 Values.	63

3.12	The Synthetic Pseudo-Periodical Time Series Source Values and Prediction Results by Algorithm MANoD; X Coordinate Axis Lists the Index of Time Intervals (100000 Values) and Y Coordinate Axis Illustrates the Original (in blue with 100000 values) and Prediction Values (in purple with 50000 values).	64
3.13	The Synthetic Pseudo-Periodical Time Series Prediction Errors by MANoD; X Coordinate Axis Lists the Index of Time Intervals (50000 values) and Y Coordinate Axis Illustrates the Prediction Error Values ($ \text{Prediction} - \text{Original} $) for 50000 Values.	64
4.1	Difference of 3 Successive Neighbouring Data in a 3-Dimensional Space	78
4.2	Earthquake RMS Testing Time Series, X Coordinate Axis Lists the Index of Time Intervals and Y Coordinate Axis Illustrates Earthquakes RMS Values.	83
4.3	Foreign Exchange Rates Testing Time Series; X Coordinate Axis Lists the Index of Time Intervals and Y Coordinate Axis Illustrates Foreign Exchange Rates Values.	83
4.4	Nile River Low Flows Testing Time Series; X Coordinate Axis Lists the Index of Time Intervals and Y Coordinate Axis Illustrates Nile River Low Flows Values.	83

4.5	Sunspot Number Testing Time Series; X Coordinate Axis Lists the Index of Time Intervals and Y Coordinate Axis Illustrates Sunspot Number Values.	84
4.6	Synthetic Pseudo-Periodical Testing Time Series; X Coordinate Axis Lists the Index of Time Intervals and Y Coordinate Axis Illustrates Synthetic Pseudo-Periodical Time Series Data Set Values.	84
4.7	Difference of 3 Successive neighbouring Data in a 3-Dimensional Space	89
4.8	A Sample Time Series of Flu Trends in United States; X Coordinate Axis List the Index of Time Intervals and Y Coordinate Axis Illustrates the Flu Trends Time Series Data Set Values. .	103
4.9	Patterns Matched (Each matched string starts with “×”); X Coordinate Axis Lists the Index of Time Intervals and Y Coordinate Axis the Flu Trends Time Series Data Set Values. . .	106
4.10	Unequal Scaling (Eigenvalues: $\lambda_1, \lambda_2, \dots, \lambda_k$) Sets Comparison for Matched Patterns from Historical Time Series (each “o” on each color line represents one of a series values of eigenvalues (λ)); X coordinate axis presents the index of eigenvalue, and Y coordinate axis indicate their values).	108

4.11	The Comparison of Eigenvalues' Determinants; X Coordinate Axis Lists the Index of Eigenvalues Series and Y Coordinate Axis Illustrates the Determinant Values for each Eigenvalues Series.	110
4.12	The Initial Flu Trends Time Series Values and Prediction Results by SFE Algorithm; X Coordinate Axis Lists the Index of Time Intervals (315 Values) and Y Coordinate Axis Illustrates the Original (in blue with 315 values) and Prediction Values (in red with 157 values).	113
4.13	The Flu Trends Time Series Prediction Errors by SFE Algorithm; X Coordinate Axis Lists the Index of Time Intervals (315 values) and Y Coordinate Axis Illustrates the Prediction Error ($ \text{Prediction} - \text{Original} $) Values with 157 Values.	113
4.14	The Initial Foreign Exchange Rates (GBP to USD) Time Series Values and Prediction Results by SFE Algorithm; X Coordinate Axis Lists the Index of Time Intervals (2295 Values) and Y Coordinate Axis Illustrates the Original (in blue with 2295 values) and Prediction Values (in red with 1148 values).	114
4.15	The Foreign Exchange Rates (GBP to USD) Time Series Prediction Errors by SFE Algorithm; X Coordinate Axis Lists the Index of Time Intervals (2295 values) and Y Coordinate Axis Illustrates the Prediction Error ($ \text{Prediction} - \text{Original} $) Values with 1148 Values.	114

4.16	The Initial U.S. Interests Rates Time Series Values and Prediction Results by SFE Algorithm; X Coordinate Axis Lists the Index of Time Intervals (582 Values) and Y Coordinate Axis Illustrates the Original (in blue with 582 values) and Prediction Values (in red with 291 values).	115
4.17	The U.S. Interests Rates Time Series Prediction Errors by SFE Algorithm; X Coordinate Axis Lists the Index of Time Intervals (582 values) and Y Coordinate Axis Illustrates the Prediction Error ($ \text{Prediction} - \text{Original} $) Values with 291 Values.	115
5.1	Prediction Results for Earthquakes (Richter Magnitude Scale) Time Series by LR Method; X Coordinate Axis Lists the Index of Time Intervals (1351 Values) and Y Coordinate Axis Illustrates the Original (in blue with 1351 values) and Prediction Values (in green with 676 values).	123
5.2	Prediction Results for Flu Trends in United States (Influenza Rates) Time Series by LR Method; X Coordinate Axis Lists the Index of Time Intervals (315 Values) and Y Coordinate Axis Illustrates the Original (in blue with 315 values) and Prediction Values (in green with 158 values).	123

5.3	Prediction Results for Nile River Flow Time Series by LR Method; X Coordinate Axis Lists the Index of Time Intervals (360 Values) and Y Coordinate Axis Illustrates the Original (in blue with 360 values) and Prediction Values (in green with 180 values).	124
5.4	Prediction Results for Sunspot Number Time Series by LR Method; X Coordinate Axis Lists the Index of Time Intervals (1200 Values) and Y Coordinate Axis Illustrates the Original (in blue with 1200 values) and Prediction Values (in green with 600 values).	124
5.5	Prediction Results for Synthetic Pseudo-Periodical Time Series by LR Method; X Coordinate Axis Lists the Index of Time Intervals (100000 Values) and Y Coordinate Axis Illustrates the Original (in blue with 100000 values) and Prediction Values (in green with 50000 values).	125
5.6	Prediction Results for Earthquakes (Richter Magnitude Scale) Time Series by ARMA Method; X Coordinate Axis Lists the Index of Time Intervals (1351 Values) and Y Coordinate Axis Illustrates the Original (in blue with 1351 values) and Prediction Values (in cyan with 676 values).	125

5.7	Prediction Results for Flu Trends in United States (Influenza Rates) Time Series by ARMA Method; X Coordinate Axis Lists the Index of Time Intervals (315 Values) and Y Coordinate Axis Illustrates the Original (in blue with 315 values) and Prediction Values (in cyan with 158 values).	126
5.8	Prediction Results for Nile River Flow Time Series by ARMA Method; X Coordinate Axis Lists the Index of Time Intervals (360 Values) and Y Coordinate Axis Illustrates the Original (in blue with 360 values) and Prediction Values (in cyan with 180 values).	126
5.9	Prediction Results for Sunspot Number Time Series by ARMA Method; X Coordinate Axis Lists the Index of Time Intervals (1200 Values) and Y Coordinate Axis Illustrates the Original (in blue with 1200 values) and Prediction Values (in cyan with 600 values).	127
5.10	Prediction Results for Synthetic Pseudo-Periodical Time Series by ARMA Method; X Coordinate Axis Lists the Index of Time Intervals (100000 Values) and Y Coordinate Axis Illustrates the Original (in blue with 100000 values) and Prediction Values (in cyan with 50000 values).	127

5.11	Prediction Results for Earthquakes (Richter Magnitude Scale) Time Series by MANoD Method; X Coordinate Axis Lists the Index of Time Intervals (1351 Values) and Y Coordinate Axis Illustrates the Original (in blue with 1351 values) and Prediction Values (in purple with 676 values).	128
5.12	Prediction Results for Flu Trends in United States (Influenza Rates) Time Series by MANoD Method; X Coordinate Axis Lists the Index of Time Intervals (315 Values) and Y Coordinate Axis Illustrates the Original (in blue with 315 values) and Prediction Values (in purple with 158 values).	129
5.13	Prediction Results for Nile River Flow Time Series by MANoD Method; X Coordinate Axis Lists the Index of Time Intervals (360 Values) and Y Coordinate Axis Illustrates the Original (in blue with 360 values) and Prediction Values (in purple with 180 values).	129
5.14	Prediction Results for Sunspot Number Time Series by MANoD Method; X Coordinate Axis Lists the Index of Time Intervals (1200 Values) and Y Coordinate Axis Illustrates the Original (in blue with 1200 values) and Prediction Values (in purple with 600 values).	130

5.15	Prediction Results for Synthetic Pseudo-Periodical Time Series by MANoD Method; X Coordinate Axis Lists the Index of Time Intervals (100000 Values) and Y Coordinate Axis Illustrates the Original (in blue with 100000 values) and Prediction Values (in purple with 50000 values).	130
5.16	Prediction Results for Earthquakes (Richter Magnitude Scale) Time Series by SFE Method; X Coordinate Axis Lists the Index of Time Intervals (1351 Values) and Y Coordinate Axis Illustrates the Original (in blue with 1351 values) and Prediction Values (in red with 676 values).	131
5.17	Prediction Results for Flu Trends in United States (Influenza Rates) Time Series by SFE Method; X Coordinate Axis Lists the Index of Time Intervals (315 Values) and Y Coordinate Axis Illustrates the Original (in blue with 315 values) and Prediction Values (in red with 158 values).	132
5.18	Prediction Results for Nile River Flow Time Series by SFE Method; X Coordinate Axis Lists the Index of Time Intervals (360 Values) and Y Coordinate Axis Illustrates the Original (in blue with 360 values) and Prediction Values (in red with 180 values).	132

5.19 Prediction Results for Sunspot Number Time Series by SFE Method; X Coordinate Axis Lists the Index of Time Intervals (1200 Values) and Y Coordinate Axis Illustrates the Original (in blue with 1200 values) and Prediction Values (in red with 600 values).	133
5.20 Prediction Results for Synthetic Pseudo-Periodical Time Series by SFE Method; X Coordinate Axis Lists the Index of Time Intervals (100000 Values) and Y Coordinate Axis Illustrates the Original (in blue with 100000 values) and Prediction Values (in red with 50000 values).	133

List of Tables

2.1	An Example of Earthquakes Time Series Database (Time Format: YYYY.MM)	26
2.2	An Example of Flu Trends Time Series Data Set (Time Format: DD/MM/YY)	27
2.3	An Example of Nile River Flow Time Series Data Set (Time Format: YYYY.MM)	29
2.4	An Example of (Monthly Average) Sunspot Number Time Series Data Set (Time Format: YYYY.MM)	30
3.1	Definitions of Various Difference Operators (where $a \in A$, $A = \{a_t\}$, $t \in \mathbf{N}$)	44
3.2	Pseudo-code for Algorithm of Moving Average based on n^{th} -order Difference (MANoD)	55
4.1	The Definition of <i>a priori</i> and <i>a posteriori</i> Knowledge	69
4.2	Categorization of Definition for Combination (3 Successive Data) Rule of Grouping 13	79
4.3	The 3D Sub-domain Correspondence of Grouping 13 Cases	80
4.4	Five Samples of Time Series for Testing	82
4.5	Statistical Results of Combination Rule for Grouping 13	85

4.6	Definition of Classes for Combinations based on 3 Successive Values in Grouping 07	88
4.7	The 3D Sub-domain Correspondence for Grouping 07 classes .	90
4.8	Correspondence between groups of Grouping 13 and Grouping 07 approaches	92
4.9	Experiments Results Comparison Between Combination Rules Grouping 13 and Grouping 07	95
4.10	Original and Transformed for Flu Trends Time Series	103
4.11	The First Half of Initial and Combination Data Series of Flu Trends in U.S.	106
4.12	Pseudo-codes for Time Series Prediction Algorithm based on Series Features Extraction	111
5.1	Testing Time Series Details	121
5.2	Prediction Results Comparison	134
5.3	Prediction Results Comparison (cont'd)	135
5.4	Prediction Results Comparison (cont'd)	136

Chapter 1

Introduction

Contents

1.1	Data Mining on Time Series	2
1.2	Motivation	4
1.3	Purpose of Research	8
1.4	Methodology	10
1.5	Overview of This Thesis	13
1.6	Summary	14

1.1 Data Mining on Time Series

Data mining is a process of analyzing data sets, which is used to discover regularities, to find unknown relationships and to understand the organizational essences from data. Data mining methods have been widely used in business (insurance, banking, retail), science research (astronomy, medicine) and government security (detection of crimes, prevention of disease) [Hand et al., 2001].

Data mining on time series is one important branch of data mining. Due to the specificity of “time”, the significance of “time-stamped” data sets can be explored and discovered by time intervals. As time series data sets are ubiquitous in everyday life, time series data mining becomes an important and active research topic nowadays.

Historical data mining approaches are designed to process “static” data sets, i.e. the indices of data sets are independent variables and they are unrelated to others attributes of the data sets. However, modern approaches indicate that there are certainly many other cases for which sequential data measurements associated with the time interval(s) present significant information about the “time-stamped” data sets. In other words, the “time” exists in the time series database as a dependant but relevant variable of data observations.

One of the best-known examples is the data sequence of Sunspot Number [Wikipedia., 2004] observations: the values at the end of one time period

affect the value(s) at the beginning of next period, such as: sunspot's size, intensity, location, and so on. Moreover, a popular pre-process procedure investigates the inter-relationship between data observations and their corresponding meta-data (as the labels of time), for example, a form of "date created", or "date modified", or other time related attributes creation; consequently, the "dynamic" data sets' analysis and process, particularly in time series, are key tasks of data mining.

Statistical methods of time series analysis apply to multi-variables sequences of data observations, which could result in a single dimension numeric variable analysis or a multi-dimension data set analysis. For example, most data sequences from real world include several numerical and nominal attributes, each of them could be treated as an individual sequence, which are not only dependent on a single dimension (e.g. time dimension) but also on other attributes sequences; therefore, these attributes may help defining sub-sets or super-sets of the time series. Specifically, to make the time series data mining progress more logical and effective, adding the time dimension to a most overriding data values sequence for producing a time series is the primary task.

Because there are concealed regularities in ir-regular periodical or periodical-like time series, establishing a model to reveal hidden patterns based on the characteristics of a time series and applying it for predicting the forthcoming events is a difficult progress, especially for a pseudo-periodical time series. As a result, the challenges of data mining on time series are to define the most efficient representation of time series data set in order to establish a

regression analysis model and procedure, to address the inflection point (peak and valley) in time domain, to make classification/clustering of time series data set, to predict/forecast the forthcoming data values and to account an interpretation for the reliability of time series analysis and prediction approaches.

1.2 Motivation

Time series analysis and prediction have always played an important role in human activities and natural sciences. Since ancient times, people traced the agricultural crops' seasonal growth time series to forecast the harvest. Chinese used time series of relative position of stars to predict astronomical events as early as 20th century B.C., and recorded the apparent path time series of the sun to estimate the obliquity of the ecliptic about 1000 B.C.. Economists evaluate the impact of economic models to human society based on time series of economical and financial factors.

There are many motivations/objectives possible for time series analysis and prediction, but they are mainly divided into four classes: Description, Explanation, Prediction and Control.

Description: for a given time series data set, the first step of analysis is normally to illustrate it in a figure, which is the brief description of the measurements for the essential nature of time series. Some of time series' figures will show "obvious" characteristics, e.g. data sequence's trend, seasonality,

cyclical fluctuations, etc. Other time series' figures denote a more complex structure, such as: “non-regular” cyclical periods with “ir-regular” components (noise). Providing an appropriate graphical description of the target time series is a good start for empirical analysis and getting the sense of the data collection.

Explanation: the time is the primary variable in time series; if there were two or more variables besides the time domain in the time series, an explanation would be required to indicate which variable is dependant on time. For example, studying the sea level in time may find that the temperature variable is the most dependent upon the time, while sea level is indirectly dependent time via temperature's influence. Meanwhile, explaining the development of predicted time series based on the existing time series (*Regression Analysis*) will give a deeper understanding for the mechanism of time series data generation.

Prediction: for the “analyzed” time series, people like to know the future values before they happen, and this is the basic task of economic and financial analysis. On various occasions, there is a close connection between “Prediction” and “Control”; for example, if a factory production is changing away from the target, an appropriate rectification of the production chain will be adopted.

Control: if a time series analysis focuses on monitoring the progression of time series and/or handling the direction of the progression, the motivation of time series analysis is to control this time series's development itself.

In a time series data set with complex structure, variations and fluctuations of the data values and their occurrences' frequency changes over time, these variations and fluctuations may imply the nature and fundamental features of the data sequences typically. Successive events with same significance may recur in a certain time interval of (periodical) time series, whereas for time series, the time interval also evolves with difference as time elapses, it emerges a phenomenon similar to a time series cyclic period. This kind of data sequence variation phenomenon, "Pseudo-Periodicity of Time Series", brings a new challenge to time series analysis and prediction.

As a result, research on pseudo-periodical time series requires consideration of the characteristics of pseudo-periodicity. The non-determinacy of time intervals between two events is the most important feature of a pseudo-periodical time series. Moreover, the events themselves' variations and fluctuations at least do not intensify the complexity but increase the difficulties for patterns recognition.

A pseudo-periodical time series could be looked upon as being a composite of random and periodical time series. Therefore, the classical analysis techniques could be not able to cover completely this complicated architecture, because the approaches aim to classify the features and patterns based either on statistical information on *a priori* knowledge what extracted from the initial time series data sets, however, the periodical and random time series are difficult to be managed together, because one expresses an appearance of what contains a certain cyclic period, the other manifests a state with random fluctuations either on value's and time's domain.

Consequently, extracting information, learning knowledge and understanding essences from a pseudo-periodical time series based on data-driven methods require to create an appropriate model for representing a general time series. Whereafter, this model is able to adapt an abstract pseudo-periodicity and to predict the forthcoming data accurately.

Hence, the motivations for this research have emerged:

- It is a really interesting and challenging research task that from a series of patterns, which represent features of the target time series, to establish one effective model. Among this modelling progress, the model should be a “global” summary of the original time series and a pattern should be a “local” one.
- To develop new algorithms for knowledge representation, extraction and mining applied to pseudo-periodical time series.
- To propose new feature detection techniques designed as data-driven methods for pseudo-periodical time series.
- To produce an accurate analysis method and prediction approaches with wide applications.
- To conduct experiments and study the performances of proposed approaches and compare the predicting results with classical methods applied on pseudo-periodical time series from various sources.

1.3 Purpose of Research

The purpose of this thesis is to propose original algorithms for time series data mining, and their approaches in pre-processing, analysis, classification, prediction and interpretation of relationship between data sets in time domain and value domain.

The goal of the research project (*Computational Approaches for Time Series Analysis and Prediction*) is to create and develop novel approaches applicable to different types of time series data sets, to integrate technologies of Data Mining, Machine Learning, Regression Analysis, Knowledge Discovery, Predictive Analytic, Classification and Clustering, Features Extraction, Pattern Recognition and Modelling in time series. An integrated prototype will possess the ability to explore information and extract knowledge from time series, to understand the regularities of changes in data, and to capture the trend for future values prediction.

The purpose of this research is a challenging task, to investigate, filter and define relationships between different pseudo-periodical time series descriptor values for further prediction and validation; to introduce automated prediction tools applicable to the main features; to develop theoretical models and data-driven methods for pseudo-periodical sequences and apply them for various time series' analysis and prediction. In one sentence, the aim is to propose and develop new automated feature detection and extraction techniques, then apply them to various pseudo-periodical time series for analysis and prediction.

This thesis also includes experimental work, which was conducted to study performances of proposed algorithms and compare with classical methods on several different types of pseudo-periodical time series, such as: monthly average of sunspot number time series [NGDC, 2006], global earthquakes' Richter magnitude scale time series [NGDC, 2006], flu trends (influenza rates) in United States time series [GoogleTrends, 2009], Nile river flooding (flow level) time series [Hyndman, 2009], synthetic pseudo-periodical time series [KDDArchive, 2007a].

Therefore, the goals of research are:

- to extend regression analysis approaches for time series analysis and prediction;
- to design computational approaches with data-driven methods for pseudo-periodical time series;
- to develop original time series analysis models and prediction algorithms;
- to test and implement the proposed approaches on various pseudo-periodical time series data sets;
- to compare the prediction and research of such classical methods in terms of flexibility and performance;

1.4 Methodology

All researches in time series analysis and prediction start with one same goal, which is to understand and describe the target time series.

The possible problems may arise because of the high dimensionality of time series data sets. These lead to pre-process time dimension series extraction, then data values sequences dependent on the time domain. This procedure of multi-dimensionality decreasing helps researchers to presume a tentative solution of simplifying the complex structure of initial data collection.

After time series data set is initially generated, the problem of searching features, identifying patterns and establishing models for time series requires adequate algorithms. It may issues a series of regression analysis progress step by step forward for constructing and completing a most suitable model for predicting.

While the regression in progressing, the feedbacks from constructed model continually return with system improvements. Therefore, the model's completing itself is also a process of regression. It could be able to identify and solve a possible current problem of analyzing.

Then, the next stage is the time series prediction. This procedure provides a model for short-term predicted values. If keeping monitoring the current gathered data measurement, it is able to step in a long-term prediction progress.

Implementation and Evaluation of the model will be a cyclic research progress for examining the model and then the predicted results. By using same defined environment, such as: the pre-specified data series's length for regression analysis and data normalization processing, the model revision and adjusted prediction process will deliver forecast solution. In this context, two classical methods widely applied in various domains: *Linear Regression (LR)* and *Auto-Regression Moving Average (ARMA)* will be used in this thesis to compare the performance of the proposed approaches.

Finally, a prototype was created; this system gained flexibility to handle different time series data sets from both natural and statistical databases and to predicting the future values before they happen.

A relevant issue about increasing the precision of prediction is the revision of the “approach towards” progress, which the progresses of regression analysis may not always deliver improvements back to the time series analysis model. In fact, it might be inevitable in case of applying the prototype for a long-term prediction based on existing invariant data sequence, however, it could give an appropriate and efficient solution that iteratively revising and improving the analysis and prediction models with operations of inserting the real forthcoming values, when they become available to measure, into the range of known data series.

Fig 1.1 illustrates the methodology for time series data mining.

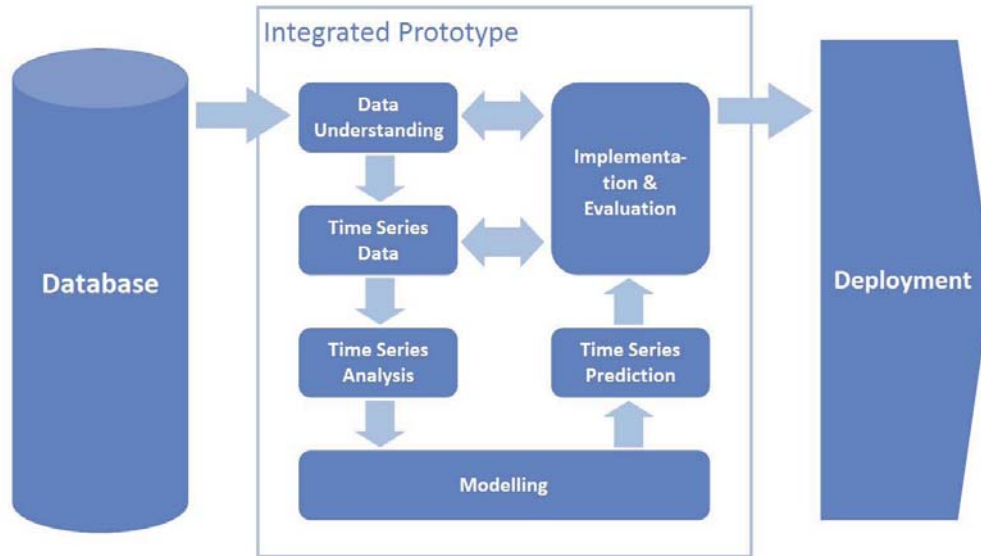


Figure 1.1: Methodology for Time Series Data Mining (From Original Database to Integrated System Deployment)

- **Data Understanding:**
To understand the data set and mining objective(s);
- **Time Series Data:**
Time Series data set initializing, e.g. necessarily data pre-processing and normalization;
- **Time Series Analysis:**
Analyze the imported time series to identify relationships and patterns;
- **Modelling:**
To construct an efficient model for prediction;
- **Time Series Prediction:**
To apply the model to predict the forthcoming values;
- **Implementation & Evaluation:**
Model Implementation and Evaluation via reviews of time series and improvement of the model;
- **Deployment:**
To offer new insights back into (or carry on) the prototype;

1.5 Overview of This Thesis

This thesis is organized in three parts and a series of chapters as follows:

Part I of this thesis contains two chapters, which are to introduce the background of Time Series Data Mining and to present the definition of Time Series and Pseudo-Periodical Time Series. Chapter 1 presents the background of Time Series Data Mining, motivation, purpose of research, methodology and overview of thesis. Chapter 2 presents the definition of time series, pseudo-periodical time series and illustrates their examples; then presents the concepts of time series analysis and time series prediction, and introduces two classical time series prediction approaches: “Linear Regression (LR)” and “Auto-Regression Moving Average (ARMA)”.

Part II of this thesis presents two successful time series prediction algorithms that I proposed in my research in two separate chapters. Chapter 3 presents the time series prediction algorithm based on Moving Average of n^{th} -order Difference (MANoD); also gives the definitions of n^{th} -order difference and moving average of a data series; then discusses the increasing precision of prediction; and presents case studies on the application of proposed MANoD algorithm for sunspot number, earthquakes and synthetic pseudo-periodical time series. Chapter 4 presents a time series prediction algorithm based on Series Features Extraction (SFE). The introduction section states the concepts of epistemology, *a priori* and *a posteriori* knowledge, also the methodology of series features extraction approaches. The following section introduces time series data classification based on a combination rule and op-

timizing the categorization to improve the performance of classification. The next section represents the concept of eigenvector for clustering and filtering the patterns recognized from the transformed time series. Case studies show the prediction results for flu trends, foreign exchange rates and interest rates in United States time series.

Part III, the last part of this thesis, presents the evaluation of proposed algorithms, compares the predicting results with the classical time series methods, and presents the conclusions of this thesis. Chapter 5 reviews and discusses the classical time series approaches: “Linear Regression (LR)” and “Auto-Regression Moving Average (ARMA)”; then uses five different testing time series data from various natural and statistical domains for evaluating the proposed time series prediction algorithms. At the end of chapter 5, the comparison of both, the classical and proposed algorithms, is also discussed. Chapter 6 concludes the thesis and discusses the summary of research, original contributions and future work; at the end of chapter 6, I state the final remarks on my research work.

1.6 Summary

This chapter introduced data mining on time series and presented motivation, purpose and methodology of research. Then, an overview of this thesis’ structure has been provided at the end of this chapter.

The next chapter will introduce the definitions of Time Series and Pseudo-

Periodical Time Series, and list their examples. Then, chapter 2 will present Time Series Analysis and its classical methods. Time Series Prediction and its objectives will be also discussed.

Chapter 2

Time Series Analysis and Prediction

Contents

2.1	What a Time Series is?	17
2.2	Pseudo-Periodical Time Series	24
2.3	Time Series Analysis	32
2.4	Time Series Prediction	34
2.5	Summary	39

2.1 What a Time Series is?

A *Time Series* is a sequence of data points, measured typically at successive time or spaced over time intervals; or having the output arranged according to time intervals. In this context, a time series is an associative data array of numbers indexed in chronological order.

Time series exist ubiquitously, stock's selling/buying prices and economists trace the market for analysis of stability of price; meteorological observations of the wind, temperature, precipitation, etc; demographers monitor rates of annual births and deaths of a defined population; manufacturers survey production to improve quality assurance; geologists continuously observe the shaking and trembling of the earth for predicting the next earthquakes; electroencephalogram tracks brain waves in order to prevent cerebral diseases; electrocardiogram traces heart waves to record and study cardiac health.

A large number of different notations are in use for time series, however, two common notations specify a time series A indexed by natural numbers, where the a_1, a_2, a_3, \dots and a_t are the measurements of time series:

$$A = \{a_1, a_2, a_3, \dots\} \quad \text{or} \quad A = \{a_t\} \quad t \in \mathbf{N} \quad (2.1)$$

or in a Vector Space, a time series A has ordered elements a_i what consist of a time-stamp t_i and their values v_i , where the $i \in \mathbf{N}$:

$$\vec{A} = \langle a_1 = \langle t_1, v_1 \rangle, a_2 = \langle t_2, v_2 \rangle, a_3 = \langle t_3, v_3 \rangle, \dots \rangle \quad (2.2)$$

2.1.1 Examples of Time Series

There are mainly two kinds of time series data, *Continuous Time Series*, which the observations occur at every instant of time; *Discrete Time Series*, which the observations spaced by (often uniformed) time intervals [Easton and McColl, 2008].

2.1.1.1 Time Series in Economy and Finances

There are many well-known time series in economy, for example, daily stock and share prices, monthly import and export total amounts, yearly corporation profits, and so on [Chatfield, 2003]. Fig 2.1 shows the “Beveridge Wheat Price Index (1819-1869)” time series, which records averaged wheat prices about 50 locations in western and central Europe. Both economists and historians are particularly interested in this sequence, which is available in many places [Chatfield, 2003], and it has been proven that it has a period cycle existing obviously (about 15.3 years) [Beveridge, 1921].

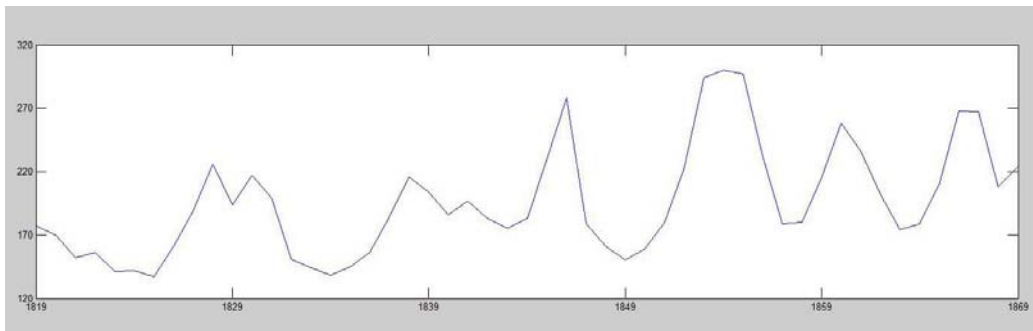


Figure 2.1: Beveridge Wheat Price Index Time Series (Source: Time Series Data Library [Hyndman, 2009]) X Coordinate Axis Lists the Time Intervals and Y Coordinate Axis Illustrates the Wheat Price Index Value.

2.1.1.2 Time Series in Nature

In the nature, there are many different types of time series (named as their sources, Meteorology, Oceanography, Geophysics, etc) and these time series are exhibited in the physical forms. These kinds of time series are recorded continuously and they can produce a continuous trace rather than observations at discrete time intervals [Chatfield, 2003]. The observations of Precipitation for example can be taken as a continuous variable (or convert it to a series in discrete for special requirements, e.g. daily, monthly, seasonally, etc), so analysts are able to survey atmospheric activities.

Fig 2.2 shows the monthly average of precipitation in West Virginia, US. Based on the historical data set, the *Return period* of precipitation can be addressed. Additionally, the intensity of a storm can be predicted for any return period and storm duration, from the charts based on historic data for a given location.

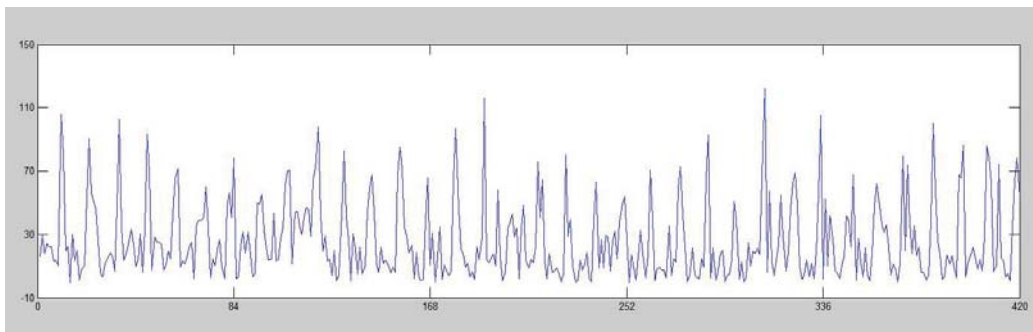


Figure 2.2: Monthly Precipitation Time Series in Southwestern Mountain, West Virginia, U.S. (Source: Time Series Data Library [Hyndman, 2009]) X Coordinate Axis Lists the Index of Time Intervals for Time Series and Y Coordinate Axis Illustrates the Monthly Precipitation.

2.1.1.3 Time Series in Demography

Demography is the analysis of population features. Demographic analysis can be applied to whole societies or to groups defined by criteria such as education, nationality, religion and ethnicity. In academia, demography is often regarded as a branch of either anthropology, economics, or sociology [Hinde, 1998].

The information of demographic time series measurements always play an important role in studies of the characteristics of human populations. Meanwhile, these information measurements normally involve several variables and the correlation between two variables is not due to any causal relationship, but each of them is correlated with another one. For example, Fig 2.3 shows a time series rates of (annual) Proportion of Church of England Marriages per 1000 of all marriages (e.g. Marriages/1000), England (1866-1911) [Yule, 1926]; the rates as a variable either relate to Church and also the married population.

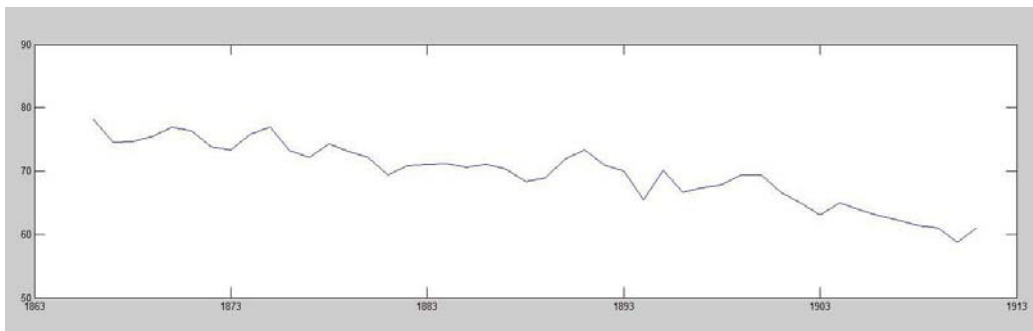


Figure 2.3: Rates of Proportion of Church of England Marriages Time Series, England, (Source: Time Series Data Library [Hyndman, 2009]) X Coordinate Axis Lists the Time Intervals of Time Series and Y Coordinates Axis Illustrates the Values of Rates of Proportion.

2.1.1.4 Time Series in Production Process Control

During the production process control, the manufacturers survey the eligible production processing to improve quality assurance [Chatfield, 2003]. A storage unit receives the measurements from process unit as a message queue and then it provides a product processing log to compare the quantity of the pre-specified target quality level.

The monitored measurements could be plotted against the time, and if the measurements do not reach the target quality level, appropriate corrective action should be executed to control the production processing [Chatfield, 2003]. Fig 2.4 shows an example for a “quality control” time series data (125 successive measurements represent 25 days products processing) from Ishikawa [Ishikawa, 1986] (the red line denotes the pre-specified target quality level (assurance)).

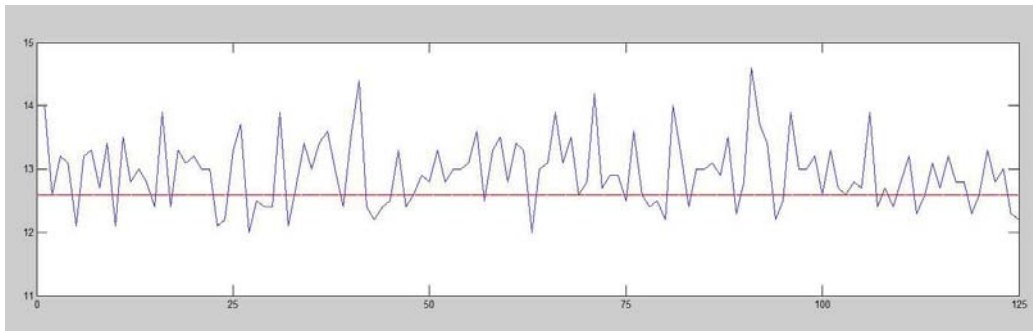


Figure 2.4: Production Process Control Time Series from Ishikawa (Source: Time Series Data Library [Hyndman, 2009]) X Coordinate Axis Lists the Index of Time Intervals for the Time Series and Y Coordinate Axis Illustrates the values of Production Process Control.

2.1.1.5 Binary Equivalent Time Series

When the observations can only be taken from two values (usually 0s and 1s), they form a special kind of time series, for example, in computer science, the statement of position of a switch (either “on” and “off”) could be recorded respectively as 1 and 0 [Chatfield, 2003]. These kinds of time series data named *Binary Equivalent Time Series* or *Binary Processes Time Series* and occur in many situations e.g. in the communication of telegraphy, computer network, etc.

These binary equivalent time series are assumed underlying the continuous time series data set with discrete data values. Moreover, this kind of time series has a *pseudo-periodicity* property: the patterns are represented either on the value’s domain or on frequency’s domain. Fig 2.5 shows a (generated) realization of a binary equivalent (processes) time series observations.

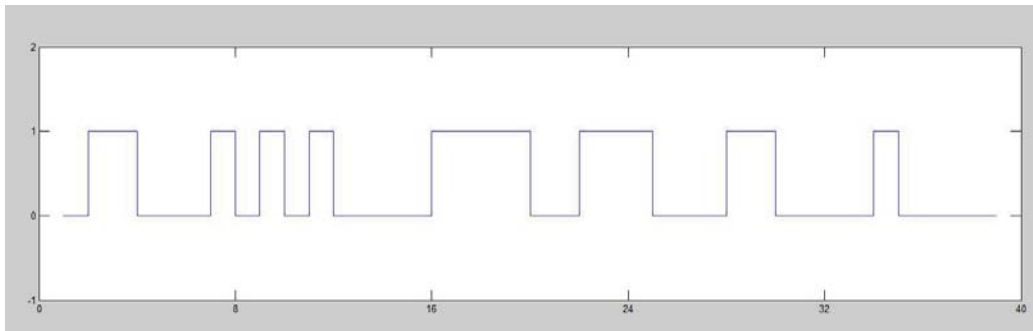


Figure 2.5: An Example for (generated) Realization of a Binary Equivalent (Processes) Time Series. X Coordinate Axis Lists the Index of Time Intervals for the Time Series and Y Coordinate Axis Illustrates the values (0 or 1) of Binary Equivalent Time Series.

2.1.1.6 Points Process Time Series

When there are recorded random incidents taking place over time, the measurements may form a completely different type of time series data set. For example, the dates of serious traffic accidents, earthquakes, or the dates of major railway disasters, etc. A series of events in this type is usually named *Point Process*, the time series is named *Points Process Time Series* [Chatfield, 2003].

Both the distribution of events in particular time period and time intervals between events are important for events detecting and analyzing. Box and Jenkins [Box and Jenkins, 1976] [Box et al., 1994] discussed the analysis methods of these kinds of points process time series. Fig 2.6 shows a example for (generated) realization of a point process time series, where each of “×” represents an event what randomly happens through time .

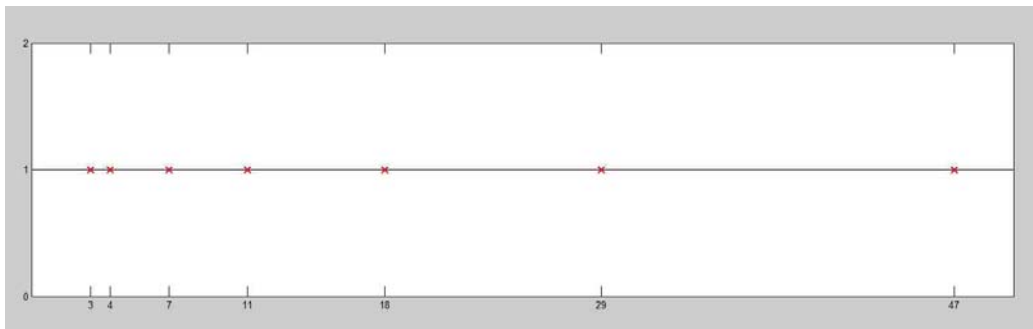


Figure 2.6: An Example for (generated) Realization of a Point Process Time Series. X Coordinate Axis Lists the Index of Time Intervals for the Time Series and Y Coordinate Axis Illustrates the Events Series (with the red “×”).

2.2 Pseudo-Periodical Time Series

A *pseudo-periodical* time series is a time series of which values recur over un-certain time intervals.

There are three elements (data measurements) a_t, a_{t+p}, a_{t+q} in a time series A , $t, t + p$ and $t + q$ are time-stamps of the three data values, p and q are two separate and unequal time intervals, then:

$$a_t \cong a_{t+p} \cong a_{t+q} \quad (2.3)$$

where $t \in \mathbf{N}$, $p \in \mathbf{N}$, $q \in \mathbf{N}$, $p \neq q$, $p/q = k > 1$, $k \in \mathbf{Z}$.

For real applications of time series, there are values showing a pattern of pseudo-periodical time series, where values show a repetition over a finite time interval. A consequence for periodical and pseudo-periodical time series is that for a finite value v and initial values, the series values are bounded. In this context, any data point (a_t) in a measurable pseudo-periodical time series is also a finite value, where the time series $A = \{a_1, a_2, \dots, a_v\}$ is a finite algebraic set, and $v \geq 1$, $t \in [1, v]$ and $v \in \mathbf{N}$

$$a_t \in [\min(a_1, a_2, \dots, a_v), \max(a_1, a_2, \dots, a_v)] \quad (2.4)$$

For an evaluation of time series prediction algorithms, which will be proposed in chapter 3 and 4, five different Testing Time Series (TTS) data sets from various domain are introduced as following.

2.2.1 Earthquakes Time Series

This original time series has been generated by the National Geophysical Data Center (NGDC). NGDC provides stewardship, products and services for geophysical data describing the solid earth, marine and solar-terrestrial environment, as well as earth observation from space [NGDC, 2006]. Its databases currently contain more than 300 digital and analog data catalogues, which include Land, Marine, Satellite, Snow, Ice, Solar-Terrestrial Subjects.

NGDC acquires, processes and analyzes technical data on the earthquake hazards, and disseminates the data in many useable formats, which mainly focus on “Richter Magnitude Scale (RMS)” of earthquakes. For example, Significant Earthquake Database contains information on more than 5000 destructive earthquakes from 2150 B.C. to present; Earthquake Slide Sets NGDC offers fourteen 35mm slide sets depicting earthquake damage throughout the world; Earthquake Intensity Database contains and felt reports for over 22000 U.S. earthquakes from 1638 to 1985; Worldwide Strong Motion Data Archive contains more than 15000 digitized and processed accelerograph records over 60 years; The Seismograph Station Bulletins Database contains more than 500000 microfiche pages from seismograph station bulletins for the years 1900 to 1965.

The measure of Richter Magnitude Scale (RMS) assigns a single number to quantify the amount of seismic energy released, therefore, the time series data set consists of RMS numbers is a pseudo-periodical time series due to

the limit of the RMS number. Table 2.1 presents an example organization of earthquakes time series data set, which are observation of global earthquakes from January 1001 A.D. to August 2006); Fig 2.7 shows the Richter Magnitude Scale (RMS) measurements of earthquakes.

Table 2.1: An Example of Earthquakes Time Series Database (Time Format: YYYY.MM)

Index	Time (A.D.)	Location	RMS	Longitude	Latitude
1	1001.01	China	6.2	34.300	109.000
2	1001.01	Italy	7.0	42.000	13.500
...
671	1500.01	China	6.9	24.500	103.000
672	1500.01	Hawaii	6.8	19.000	-155.500
...
1350	2006.08	Argentina	5.6	-33.131	-68.707
1351	2006.08	France	4.3	44.000	6.800

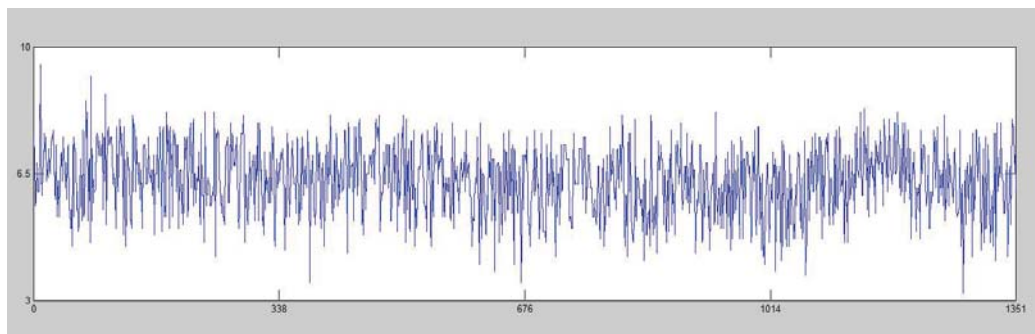


Figure 2.7: An Example of Earthquakes Time Series Data Set (Richter Magnitude Scale (RMS) by Index); X Coordinate Axis Lists the Index of Time Intervals for the Time Series and Y Coordinate Axis Illustrates the Values of RMS.

2.2.2 Flu Trends Time Series

Each week, millions people around the world catch flu. Historical research indicate that there is a close relationship between two peaks of flu condition occurrences, and the flu season in time has a pseudo-period. United States Center for Disease Control and Prevention [USCDC, 2009] currently hold a surveillance system to monitor the flu's situation. Based on the reported measurements and influenza-like (ILI) estimates of sickness in United States, the set of Flu Trends in United States is an ordered data sequence set by time interval; and since there are flu season occurrences, Flu Trends in United States is a pseudo-periodical time series.

Table 2.2 presents an sample of Flu Trends time series which are weekly influenza rates records: $(\text{influenza}/\text{entire population}) \times 100\%$ from 01st June 2003 to 07th June 2009 from United States Center for Disease Control and Prevention. Fig 2.8 shows a time series measurements of the Flu Trends rates in United States.

Table 2.2: An Example of Flu Trends Time Series Data Set (Time Format: DD/MM/YY)

Index	1	2	...	200	201	...	314	315
Time	01/06/03	08/06/03	...	25/03/09	01/04/09	...	31/05/09	07/06/09
Flu Rates	0.509%	0.546%	...	1.485%	1.289%	...	0.780%	0.739%

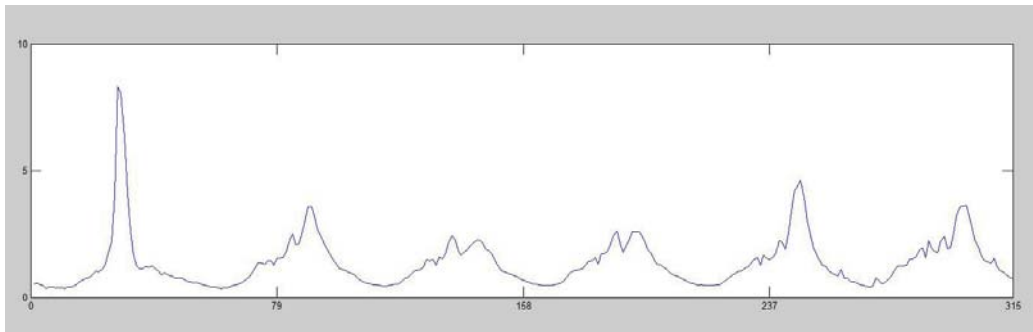


Figure 2.8: An Example of Flu Trends (Influenza Rates) in United States Time Series Data Set; X Coordinate Axis Lists the Index of Time Intervals and Y Coordinate Axis Illustrates the Influenza Rates Values.

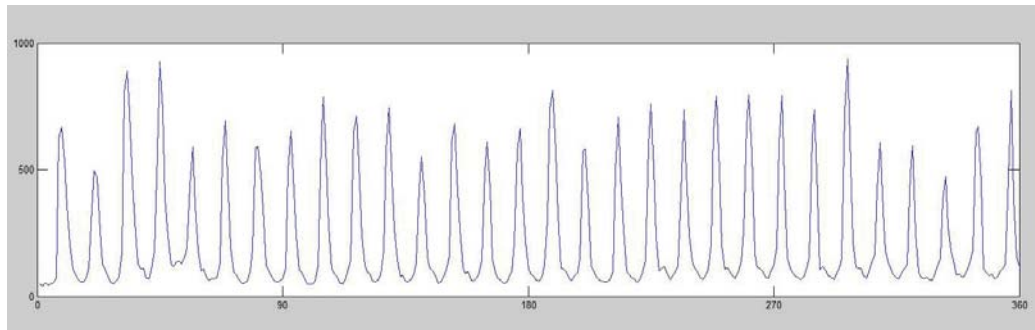
2.2.3 Nile River Flooding Time Series

From millions years ago to present, the river Nile played a major role in politics and social life and the Nile still supports much of the population living along its banks, in otherwise inhospitable regions of the Sahara. The river is flooding every summer, depositing fertile silt on the plains. When Nile flooded and inundated annually, the river water made the landing surrounding it extremely fertile and providing food for general population. The Egyptians knew their life is related with the Nile's waters, even the whole of the structure of Egypt's society. As a result of flooding annually (around several months every year), the Nile river's flow shows a pseudo-periodical time series.

Table 2.3 presents an example of Nile River Flow time series data set from January 1900 A.D. to December 1930 A.D.; Fig 2.9 shows a time series measurements of the Nile River Flow.

Table 2.3: An Example of Nile River Flow Time Series Data Set (Time Format: YYYY.MM)

Index	1	2	...	180	181	...	359	360
Time	1900.01	1900.02	...	1914.06	1914.07	...	1929.11	1929.12
Flow	48.710	40.714	...	109.032	89.3548	...	153.333	119.355

**Figure 2.9:** An Example of Nile River Flow Time Series Data Set; X Coordinate Axis Lists the Index of Time Intervals and Y Coordinate Axis Illustrates the Flow Values of Nile River.

2.2.4 Sunspot Number (Monthly Average) Time Series

Early research showing that sunspots have a cycle period start in modern times with George Ellery Hale: he has found that the sunspot period cycle is 22 years, because the magnetic polarity of sunspots reverses after 11 years. Rudolf Wolf proposed in 1849 in Zürich to count sunspot numbers by what is nowadays called: “Wolf Number” or “International Sunspot Number” using numeric values related to sunspots’ number and size, their location and

instrumentation used. Based on sunspots characterization and observations, Time Series Sunspot Data Set is an ordered data set of sunspot numbers based on observation, which can be treated as a tracking record of solar activities. Form the point of view of *Pseudo-Periodical Time Series*, sunspot number data set is a pseudo-periodical time series, since there is not a fixed value of period cycle, but a series of period cycle's values with average of about 22 years.

Table 2.4 present an example of (monthly average) Sunspot Number time series data set, which from January 1901 A.D. to December 2000 A.D. Fig 2.10 shows the time series measurements of Sunspot Number.

Table 2.4: An Example of (Monthly Average) Sunspot Number Time Series Data Set (Time Format: YYYY.MM)

Index	1	2	...	600	601	...	1199	1200
Time	1901.01	1901.02	...	1950.12	1951.01	...	2000.11	2000.12
Sunspot Number	0.2	2.4	...	54.1	59.9	...	106.8	104.4

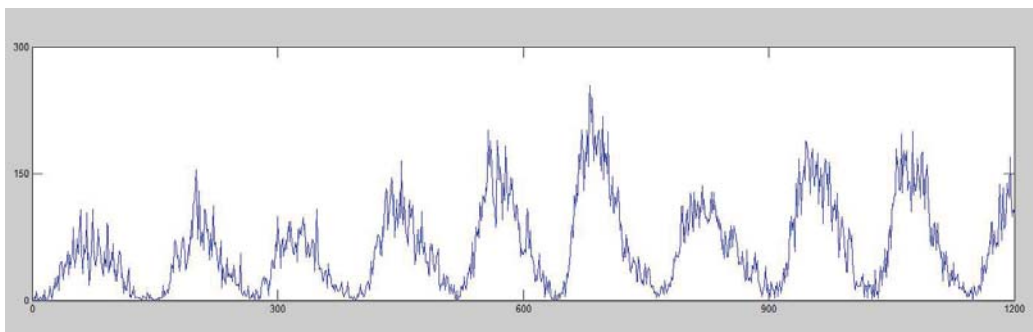


Figure 2.10: An Example of (Monthly Average) Sunspot Number Time Series Data Set; X Coordinate Axis Lists the Index of Time Intervals and Y Coordinate Axis Illustrates the Monthly Average Sunspot Number Values.

2.2.5 Synthetic Pseudo-Periodical Time Series

The pseudo-periodical synthetic time series data set has been taken from Knowledge Discovery in Database Archive (KDD Archive), University of California, Irvine [KDDArchive, 2007b]. KDD Archive is an online repository of large data sets which encompasses a wide variety of data types, analysis tasks, and application areas. This time series data set is designed for testing indexing schemes in time series databases. The data appears highly periodical, but never repeats exactly itself in a specific (time) interval. This feature is designed to challenge the indexing tasks. This time series data set [KDDArchive, 2007a] is generated by independent calls of the mathematical function, where $0 \leq \bar{t} \leq 1$:

$$\bar{y} = \sum_{i=3}^7 \sin \left(2\pi (2^{2+i} + \text{rand}(2^i)) \bar{t} \right) \quad (2.5)$$

The function $\text{rand}(x)$ produces a random integer between 0 and x . Fig 2.11 shows 100000 measurements of the Synthetic pseudo-periodical time series.

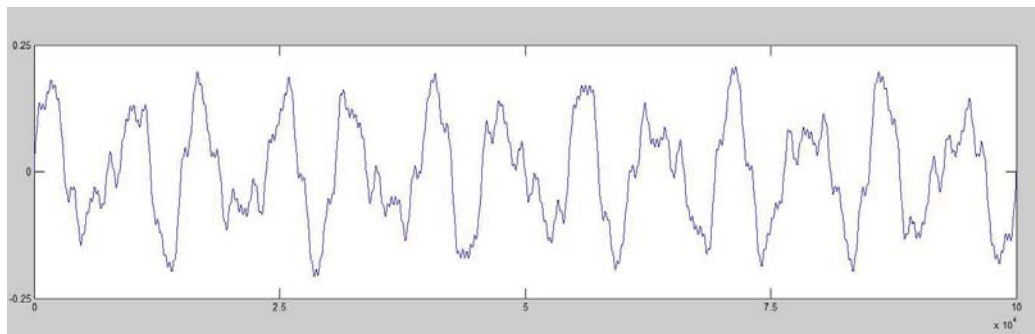


Figure 2.11: An Example of Synthetic Pseudo-Periodical Time Series Data Set, X Coordinate Axis Lists the Values of Variable \bar{t} (see eq.(2.5) with 100000 values) and Y Coordinate Axis Illustrates the values of \bar{y} (see eq.(2.5) with 100000 values).

2.3 Time Series Analysis

The objective of *Time Series Analysis* is to analyze the collected data in order to discern whether there are some patterns over time. Meanwhile, to account for the evolving nature of data surveillance, time series analysis is an alternative for monitoring cases and identifying events' occurrence.

Time series analysis also attempt to understand the underlying context of the whole data sequence, i.e. where did the time series come from, or what generated or formed data, etc; and then to follow the discovered regularities to make a prediction/forecast.

Normally, the strategies of time series analysis intend to establish a model to distinguish the situation from ordinary data (an *a priori* analysis) firstly, and then from data analyzed to describe results or its context (an *a posteriori* analysis). There are also additional possibilities to transfer the analysis results into another corresponding model, for example, from time domain into frequency domain.

A time series analysis model is attempting to represent the essential aspects of time series. Like mathematical models, time series analysis models may have many forms, i.e. statical/dynamical systems, statistical models, abstract models, etc.

Furthermore, time series analysis models will generally reflect the fact that observations close together in time will be more closely related than observations further apart. In addition, time series analysis models will often make

use of the natural one-way (non-reversible) ordering of time so that measurements in the time series data sets for a given time will be expressed as deriving in some way from past values, rather than from future values.

An accurate mathematical model will be close enough or match the actual existing data. As a result, defining validation approaches (e.g. distance methods) to measure the model via mathematical and logical methods is a suitable tool for model assessment.

The classical methodologies for time series analysis include “Linear Regression Model (LR)”, which constructs a bridge formula between a given time series data set and predicted value(s) [Cohen et al., 2002]. Linear Regression is a form of regression analysis, consequently, the function with established regression coefficients can be treated as the skeleton of the original time series data set. The line in “linear” model may not be a straight line, but rather than the way in which the regression coefficients occur in the regression equation.

The Box-Jenkins methodology, also called *Auto-Regression Moving Average* (ARMA), consists of two parts: an *Auto-Regressive* (AR) model and a *Moving Average* (MA). Auto-Regression Moving Average is usually referred to as a $ARMA(p,q)$ model where p and q represent the order of the models $AR(p)$, $MA(q)$. [Box and Jenkins, 1976] [Box et al., 1994]

There are other methods designed for time series analysis, such as: a variation of ARMA, *Auto-Regressive Integrated Moving Average* (ARIMA) [Mills, 1990] [Percival and Walden, 1993]; commonly applied on modelling financial

time series, *Auto-Regressive Conditional Heteroscedasticity* (ARCH) [Engle, 1982]; and its' generalized version, *Generalized Auto-Regressive Conditional Heteroskedasticity* (GARCH) [Bollerslev, 1986].

2.4 Time Series Prediction

Time Series Prediction is an application of time series analysis models. The objective is to use a model to forecast the future events based on collected measurements or known previous events; and to address the future values before they could be measured. A popular example is to predict tomorrow's price of a stock based on today and previous performance.

From the mathematical point of view, time series prediction represents the use of a mathematical equation to predict future data points based on known previous data sequence measurements.

$$a_{t+1} = f(a_1, a_2, a_3, \dots, a_{t-1}, a_t) \quad (2.6)$$

where given the first t measured data points of time series $A = \{a_t\}$, $t \in \mathbf{N}$, the target of prediction is to use a developed model to address the forthcoming data point a_{t+1} (*Short Term Prediction*).

If it is required to indicate a further future value over a long time interval for some particular research cases, that is used to predict the m steps forward

based on existing time series $A = \{a_t\}$ (*Long Term Prediction*):

$$a_{t+m} = f(a_1, a_2, a_3, \dots, a_{t-1}, a_t) \quad (2.7)$$

where $t \in \mathbf{N}$, $m \in [1, k]$, $k > 1$.

Indeed, the long term prediction could consist of a series of successive progresses of short term prediction, where in the each step it involves the predicted result from the last step (or actual values just happened).

One of the most important activities of human civilization is to record observations, then to forecast the forthcoming events and undiscover future. There are obviously numerous reasons to trace and analyze the time series data set.

To gain a better understanding of the data generating approaches, the time series analysis model (as the function f in eq.(2.6) and eq.(2.7)) should be not a *Black Box*, although many research results indicated that the Black Box (“a Heuristic algorithm” [Pearl, 1984] [Goodman and Hedetniemi, 1977] [Aho et al., 1983]) methods also obtained good results. That is because the approaches are procedure-emphasized for time series analysis and prediction like the Neural Network methodology or other statistical methods. A clear explanation should be produced on what is changing with time. In this thesis, both proposed algorithms for time series prediction are presented as *White Box** algorithms.

*White Box, In contrast to a Black Box, the inner components or logic are available for inspection, it makes the (sub-)system easier to understand. [Beizer, 1995]

Generally, a time series analysis model is to summarize the initial time series data sets and present the knowledge of nature. Time series prediction approaches assume the unknown future values existing and involve them to expand the time series analysis model. Thus, An optimal framework for time series analysis and prediction should be independent and no matter what distribution of the target time series as the system's input.

2.4.1 Linear Regression Method

Linear regression is a form of regression analysis in which the relationship between one or more independent variables and dependent variable, is modelled by a least squares function, called a linear regression equation.

$$\begin{aligned}\hat{Y} &= \alpha X + \varepsilon \\ &= \alpha_0 + \sum_{i=1}^P \alpha_i X_i + \varepsilon\end{aligned}\tag{2.8}$$

where the values of \hat{Y} are predicted values from the data sequence X ; and X_1, X_2, \dots, X_P are known and measured values of the initial data sequence; The disturbance term ε is added to eq.(2.8) relationship to capture the influence of everything else on \hat{Y} other than X .

A linear regression equation with one independent variable represents a straight line when the predicted value (i.e. the dependent variable from the regression equation) is plotted against the independent variable: this is called a simple linear regression.

However, note that generally "linear" does not refer to a straight line, but rather to the way in which the regression coefficients occur in the regression equation. The results are subject to statistical analysis [Edwards, 1976] [Chatterjee and Hadi, 2006].

The most common form of linear regression is Least Squares Fitting (LSF), this method was first described by Carl Friedrich Gauss around 1794 [Björck, 1996]. Least squares fitting of lines and polynomials are both forms of linear regression, which the first and main objective of regression analysis is to best-fit the data by estimating the parameters of the model.

A Linear Regression model is widely used in many domains, such as: technical analysis, biological, behavioral and social sciences to describe possible relationships between (independent/dependent) variables.

2.4.2 Auto-Regression Moving Average Method

Auto-Regression Moving Average (ARMA), so called "Box-Jenkins model", is typically applied to time series analysis and prediction. An ARMA model commonly used in the study of long-term tracking in many domains, such as: long-term time series analysis research of natural disasters prevention, consumer behavior, seasonal marketing price, finance scale prediction, and so on.

An ARMA model consists of two parts: an Auto-Regression (AR) part and a Moving Average (MA) part. This model is usually referred to as the

ARMA(p, q) model, where p is the order of the AR(p) part and q is the order of the MA(q) part, as following defined:

Definition 2.1 - Auto-Regression (AR) Model:

Auto-Regression model provides a way to express the prediction of the following value in the initial time series by using previous finite number of values affected by white noise and AR(p) of order p is defined by [Box and Jenkins, 1976] [Box et al., 1994]:

$$X_t = \sum_{i=1}^p \alpha_i X_{t-i} + \varepsilon_t \quad (2.9)$$

where α_i is the auto-regression coefficient, X_t is the series under investigation, p is the length of the filter, which is commonly less than the length of the series, and the ε_t is a white noise process with a zero mean and variance σ^2 .

Definition 2.2 - Moving Average (MA) Model:

The notation MA(q) refers to the Moving Average model of an order q [Box and Jenkins, 1976] [Box et al., 1994]:

$$X_t = \sum_{i=1}^q \beta_i \varepsilon_{t-i} + \varepsilon_t \quad (2.10)$$

where the term β_i is the moving average coefficient, X_t is the series under investigation, q is the length of filter, which is commonly less than the length of the series, and ε_t represents the error (noise) terms.

Definition 2.3 - Auto-Regression Moving Average Model:

Auto-Regression Moving Average (ARMA) model contains an infinite AR(p) model with p auto-regression terms and a finite MA(q) model with q moving average terms [Box and Jenkins, 1976] [Box et al., 1994]:

$$X_t = \sum_{i=1}^p \alpha_i X_{t-i} + \sum_{i=1}^q \beta_i \varepsilon_{t-i} + \varepsilon_t \quad (2.11)$$

where the error term ε_t are generally assumed to be independent identically-distributed random variables (i.i.d.) sampled from a normal distribution with zero mean: $\varepsilon_t \sim N(0, \sigma^2)$ where σ^2 is the variance.

Definition 2.4 - Non-Linear Regression:

In statistics, nonlinear regression is a form of regression analysis in which observational data are modeled by a function which is a nonlinear combination of the model parameters and depends on one or more independent variables. The data are fitted by a method of successive approximations. [Seber and Wild, 2003] [Bethea, 1995]

2.5 Summary

This chapter has given the definition of *Time Series* and introduced several typical examples of time series, such as: economy and finance time series, Nature time series, Demography time series, Production Process Control time series, Binary Equivalent time series and Points Process time series.

Also, this chapter has given the definition of *Pseudo-Periodical Time Series*.

Then, this chapter has also presented the *Time Series Analysis* and its classical models, such as: Linear Regression (LR) and Auto-Regression Moving Average (ARMA) and their derivatives.

In this chapter, it has been discussed *Time Series Prediction* and its objectives.

The next chapter will introduce an original time series prediction algorithm based on moving average of n^{th} -order difference and its performance.

Chapter 3

A Time Series Prediction

Algorithm based on Moving

Average of n^{th} -order Difference

Contents

3.1	Introduction	42
3.2	n^{th} -order Difference	44
3.3	Moving Average of Data Series	46
3.4	The Prediction Algorithm	47
3.5	Case Studies	60
3.6	Summary	64

3.1 Introduction

Time series prediction proposes algorithms for which previous data sequence (mainly finite observation sequences of data points related to uniform time intervals) are used to generate models to forecast the future data points of the series. It is widely applied to different format time series data sets in various domains (as described in Chapter 2). From a procedural perspective, using computational approaches may first require mathematical analysis to describe and breakdown the initial time series problem into simpler sub-problems for further computational modelling.

A historical main constraint in using mathematical series models for prediction was the fact that the performance of the model is related to the length of data series, but nowadays is not anymore an issue from neither computational nor data storage and processing points of view. However, most machine learning methods face the difficulty of requiring *a priori* knowledge about the problem at hand.

On the other hand, results of some traditional methods applied in time series analysis can not satisfy the demand of specific applications. We intend to address these drawbacks for the restricted problem of pseudo-periodical series with limited boundaries by a two-step approach: we propose hereby a new algorithm to approximate the time series terms using the moving average of n^{th} -order difference of already known values and intend to address later the problem of error of approximation by a hybrid model.

Therefore future work is proposed to identify as accurately as possible a general approximation by use of a supervised-learning model to forecast a further approximation error if found necessary.

We propose an algorithm for efficient mining of pseudo-periodical time series. Applications to sunspot number time series prediction, earthquake time series prediction and synthetic pseudo-periodical time series are added to explain the generality of the proposed algorithm, by exploring some interesting properties related to moving average of first-order difference for bounded time series.

A further generalization to the use of the sum of n^{th} -order difference to increase forecast performances and a hybrid approach to combine the results of the moving average of n^{th} -order difference of time series with a supervised-learning model of the error of value approximation are also proposed.

We study the possibility that pre-processing of time series combined with *a priori* knowledge and hybrid models can increase prediction performances for time series, even for mining noisy data. The results highlight our proposed algorithm's efficiency in mining bounded pseudo-periodical patterns in time series with direct applications in sunspot number time series prediction, earthquake time series prediction and synthetic pseudo-periodical time series prediction.

3.2 n^{th} -order Difference

A Difference Operator involves the difference between successive values of a function of a discrete variable. A discrete variable is the one that is defined or of interest only for values that differ by a (or some) finite amount.

A difference operator is a mathematical operator, which maps a function f to another function whose values are the corresponding finite differences. Table 3.1 shows several types of difference operator:

Table 3.1: Definitions of Various Difference Operators (where $a \in A$, $A = \{a_t\}$, $t \in \mathbf{N}$)

Operator	Definition
Forward Difference Operator:	$\Delta_h[f](a) = f(a + h) - f(a)$
Backward Difference Operator:	$\nabla_h[f](a) = f(a) - f(a - h)$
Central Difference Operator:	$\delta_h[f](a) = f(a + \frac{1}{2}h) - f(a - \frac{1}{2}h)$

Definition 3.1 - Forward Difference Operator:

Forward Difference is a finite difference, which defined for a given functional f with real values as:

$$\Delta f(a) = f(a + 1) - f(a) \quad (3.1)$$

$\Delta f(a)$ is also named “First-order Difference” (or Simply Difference) of $f(a)$.

The same principle, the “Second-order Difference” is defined as:

$$\begin{aligned}\Delta^2 f(a) &= \Delta f(a+1) - \Delta f(a) \\ &= (f(a+2) - f(a+1)) - (f(a+1) - f(a)) \\ &= f(a+2) - 2f(a+1) + f(a)\end{aligned}$$

The higher order differences are obtained by repeated operations of the forward difference operator, such as: n^{th} -order Difference is defined as:

$$\Delta^n f(a) = \sum_{i=0}^n (-1)^{n-i} C_n^i f(a+i) \quad (3.2)$$

where $C_n^i = \binom{n}{i} = \frac{n!}{i!(n-i)!}$, $0 \leq i \leq n$ is the Binomial Coefficient.

The forward differences are useful in solving ordinary differential equations by single-step predictor-corrector methods. For instance, a forward difference above predicts the value of P_{t_i} from the derivative $[f](P_{t_{i-1}})$ and from value of $P_{t_{i-1}}$. If the data values are equally spaced with the step size h , the truncation error of the forward difference approximation has the order of $O(h)$ [Flajolet and Sedgewick, 1995].

The backward differences are useful for approximating the derivatives if data in the future are not available yet. Moreover, the data in the future may depend on the derivatives approximated from the data in the past. If the data values are equally spaced with the step size h , the truncation error of the backward difference approximation has the order of $O(h)$ [Flajolet and

Sedgewick, 1995].

In order to make a good forward prediction, we propose the time series prediction algorithm based on Moving Average of n^{th} -order Difference (MANoD) [Lan and Neagu, 2007b] [Lan and Neagu, 2007a] [Lan and Neagu, 2006], which uses the Forward Difference Operator as follows.

3.3 Moving Average of Data Series

In statistics, a Moving Average, also named a Rolling Average, is used to analyze a set of data points by creating a series of averages (Arithmetic Mean) of different subsequences of the full data terms [Chou, 1975]. As a result, a moving average is a series of numbers (data) instead of a single one value.

Definition 3.2 - Moving Average (for Time Series):

(Cumulative) Moving Average is a way of smoothing by averaging n terms of the time series. In mathematics and statistics, moving average is used as a generic smoothing operation or an example of a convolution. As a result, (cumulative) moving average is un-weighted (or weighted moving average, which all weights equal 1) mean of previous m data points (stream) in the initial time series:

$$CMA_m^n = \frac{P_1 + P_2 + P_3 + \cdots + P_m}{m} \quad (3.3)$$

where P_1, \dots, P_m is the values of time series, $m = 1, 2, 3, \dots, n = 1$.

Same principle, the sequence ${}^n CMA_m$ means n -Moving Average with m data points input.

$$CMA_m^n = \frac{1}{m} \left(\sum_{i=1}^{n+i-1} P_i + \sum_{i=2}^{n+i-1} P_i + \sum_{i=3}^{n+i-1} P_i + \dots + \sum_{i=m}^{n+i-1} P_i \right)$$

where $n \geq 1$. We consider the n -Moving Average of m data values of time series in order to deduce a simple form for CMA_{m+1}^n . There is a brute-force method to calculate the cumulative moving average based on all stored data, or simply update the average every time a new data point (P_{m+1}) arrives, where $CMA_0 = 0$:

$$\begin{aligned} CMA_{m+1} &= CMA_i + \frac{1}{i+1} (P_{m+1} - CMA_m) \\ &= \frac{1}{m} \sum_{j=1}^m \sum_{i=1}^{n+j-1} P_i \end{aligned}$$

3.4 The Prediction Algorithm

3.4.1 Implementation of the Prediction Algorithm

Considering the equations on the last two sections for same length n of data input, we can find out that the same rule applies for $n + 1$: therefore, based on the induction principle (Peano) eq.(3.2) is valid for any natural value of n . If $f(a)$, with n , a natural number, generates a discrete series a_m , then the

previous result (eq.(3.2)) can be written as:

$$\begin{aligned}\Delta^n f(m) &= \Delta^{n-1} f(m-1) - \Delta^{n-1} f(m) \\ D_m^n &= D_{m+1}^{n-1} - D_m^{n-1}\end{aligned}\tag{3.4}$$

where D_m^n means $\Delta^n f(m)$.

The proof for eq.(3.2), a n^{th} -order Difference equals the difference of two lower differences ($(n-1)^{th}$ -order) is presented in the Appendix A.

The n^{th} -order difference is used in the binomial transform of a function usually, also the Newton forward difference equation and the Newton series [Flajolet and Sedgewick, 1995]. These are very useful prediction relationships with the main drawback of difficult numerical evaluation, because there is a very rapid growing of the binomial coefficients for a large value of n (the length of recursion). In order to avoid a complex calculus and also to provide a relationship for time series analysis and prediction, the main idea of the algorithm MANoD relates to the fact that applying the difference operator generates another series from the initial original series featuring the property of pseudo-periodicity.

Since the initial original time series data set is bounded, the new series generated by the difference operator is also bounded and its average converges to zero (see the definition of “*Pseudo-Periodical Series*” eq.(2.3)). The following paragraph provide a proof for the result above, and exemplifies with the monthly average of Sunspot Number data set case of 600 months values (see Fig 3.1, Fig 3.2 and Fig 3.3).

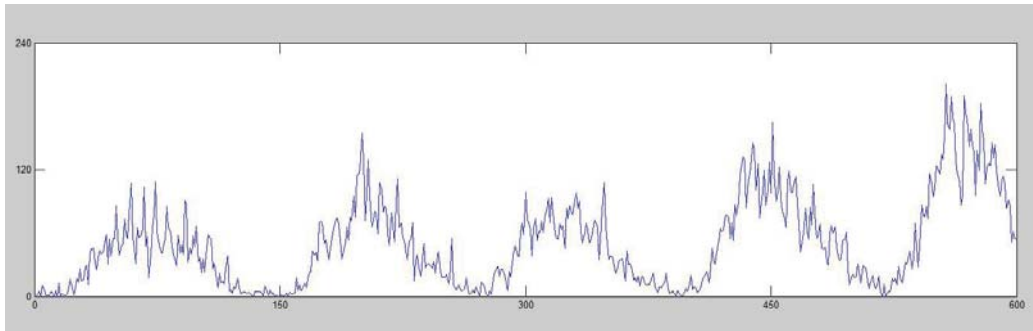


Figure 3.1: The Monthly Average Values of Sunspot Number Time Series for 600 Months; X Coordinate Axis Lists the Index of Time Intervals (600 Months) and Y Coordinate Axis Illustrates the Values of Monthly Average Sunspot Number.

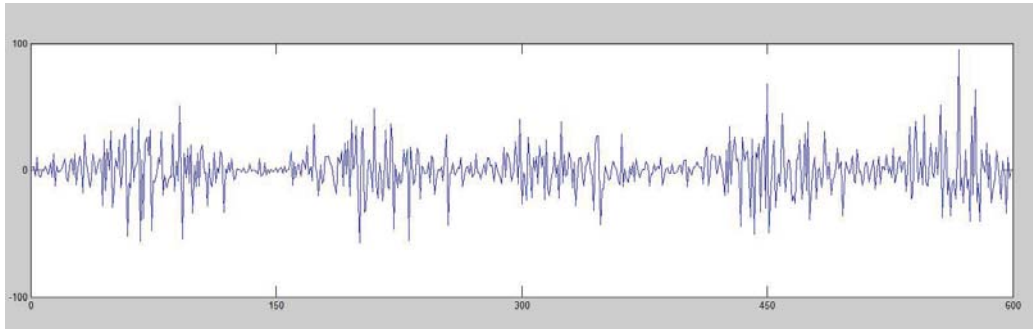


Figure 3.2: First-Order Difference (D_m^1) of Monthly Average Values of Sunspot Number Time Series; X Coordinate Axis Lists the Values of m with 600 Samples and Y Coordinate Axis Illustrates the First-Order Difference Values for D_m^1 .

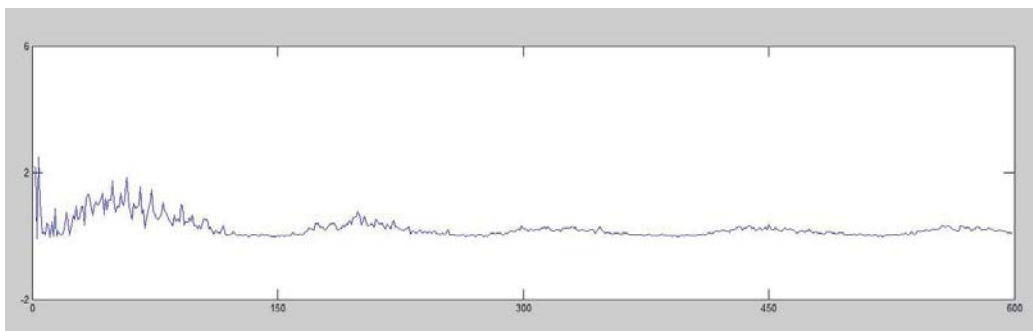


Figure 3.3: The Moving Average (E_m^1) of First-order Difference (D_m^1) of Monthly Average Values of Sunspot Number Time Series; X Coordinate Axis Lists the Values of m with 600 Samples and Y Coordinate Axis Illustrates the Moving Average Values for E_m^1 .

D_m^1 represents the First-order Difference of initial original time series data set $A = \{a_m\}$ as represented in Fig 3.2, and the first-order difference time series shows a pseudo-periodical bounded shape with amplitude modulated in time dimension. The moving average of first-order difference time series for initial original data set a_m can then be constructed as:

$$D_m^1 = a_{m+1} - a_m, \quad m \geq 1 \quad (3.5)$$

$$E_m^1 = \frac{1}{m}(D_1^1 + D_2^1 + \dots + D_m^1) = \frac{1}{m} \sum_{i=1}^m D_i^1 \quad (3.6)$$

Then limit of moving average time series E_m^{1*} (for an easy calculation it can be consider as the following):

$$\lim_{m \rightarrow \infty} E_m^1 = \lim_{m \rightarrow \infty} \frac{1}{m} \sum_{i=1}^m D_m^1 \quad (3.7)$$

Therefore, based on eq.(3.6) and eq.(3.7):

$$\begin{aligned} \lim_{m \rightarrow \infty} E_m^1 &= \lim_{m \rightarrow \infty} \frac{1}{m} \sum_{i=1}^m (a_{i+1} - a_1) \\ &= \lim_{m \rightarrow \infty} \frac{1}{m} ((a_{m+1} - a_m) + (a_m - a_{m-1}) + \dots + (a_2 - a_1)) \\ &= \lim_{m \rightarrow \infty} \frac{1}{m} (a_{m+1} - a_1) \\ &= \lim_{m \rightarrow \infty} \frac{a_{m+1}}{m} - \lim_{m \rightarrow \infty} \frac{a_1}{m} \end{aligned} \quad (3.8)$$

*if no other specific instruction, E_m^n means ${}^1E_m^n$ in this thesis, which is 1-moving average for n^{th} -order difference series with m data points input

For a large m , since a_1 is a limit value, the second term in eq.(3.8), becomes negligible. Also a_{m+1} is a limited value given the initial original constraints on the bounded time series what considered of interest and therefore the first term in eq.(3.8) has a null limit also:

$$\lim_{m \rightarrow \infty} E_m^1 = \left(\lim_{m \rightarrow \infty} \frac{a_{m+1}}{m} - \lim_{m \rightarrow \infty} \frac{a_1}{m} \right) \rightarrow 0 \quad (3.9)$$

Indeed, it is easily seen that the result (eq.(3.9)) is verified by the practical example in Fig 3.3.

Based on the result in eq.(3.9) as depicted in Fig 3.3, we can state that: *given a time series $A = \{a_i\}$, $i = 1, 2, 3, \dots, m$, let the first-order difference be $D = \{D_m^1\}$, $i = 1, 2, 3, \dots, m$ (Time Series Analysis), the aim is to determine the value for further value a_{m+1} (Time Series Prediction) based on previous data measurements (and some negligible error).*

The series of cumulative moving average for first-order difference is easy to calculate:

$$\begin{aligned} E_{m-1}^1 &= \frac{1}{m-1} (D_1^1 + D_2^1 + \dots + D_{m-1}^1) \\ &= \frac{1}{m-1} \sum_{i=1}^{m-1} D_i^1 \end{aligned} \quad (3.10)$$

Since $E_m^1 \rightarrow 0$ for a large value of m , then:

$$E_m^1 = E_{m-1}^1 + \varepsilon \quad (3.11)$$

Where $\varepsilon > 0$ is a negligible error for a large value of m ; and replacing in eq.(3.11) E_m^1 and E_{m-1}^1 by using the result in eq.(3.10):

$$\frac{1}{m} \sum_{i=1}^m D_i^1 = \frac{1}{m-1} \sum_{i=1}^{m-1} D_i^1 + \varepsilon \quad (3.12)$$

And therefore, based on eq.(3.6):

$$\frac{1}{m} \left(\sum_{i=1}^{m-1} D_i^1 + D_m^1 \right) = \frac{1}{m-1} \sum_{i=1}^{m-1} D_i^1 + \varepsilon$$

Thus, the D_m^1 can be presented as:

$$D_m^1 = m \left(\frac{1}{m-1} \sum_{i=1}^{m-1} D_i^1 + \varepsilon \right) - \sum_{i=1}^{m-1} D_i^1 \quad (3.13)$$

At the same time, because the first-order difference $D_m^1 = a_{m+1} - a_m$ and put it into eq.(3.13):

$$a_{m+1} = a_m + m \left(\frac{1}{m-1} \sum_{i=1}^{m-1} D_i^1 + \varepsilon \right) - \sum_{i=1}^{m-1} D_i^1 \quad (3.14)$$

For simplicity, the equation above can be simplified to:

$$\begin{aligned} a_{m+1} &= a_m + \frac{m}{m-1} \sum_{i=1}^{m-1} D_i^1 + m\varepsilon - \sum_{i=1}^{m-1} D_i^1 \\ &= a_m + \frac{1}{m-1} \sum_{i=1}^{m-1} D_i^1 + m\varepsilon \end{aligned} \quad (3.15)$$

And replacing the difference operator D as defined from eq.(3.6)

$$\begin{aligned} a_{m+1} &= a_m + \frac{1}{m-1}(a_m - a_1) + m\varepsilon \\ &= \frac{1}{m-1}(ma_m - a_1) + m\varepsilon \end{aligned} \quad (3.16)$$

On the whole, the prediction precision for the forthcoming value a_{m+1} depends on the n^{th} -order difference D_m^n (for the example above, it is the simplest instance that is processed by the first-order difference and the 1-moving average).

In addition, the result in the eq.(3.16) is obtained by considering the moving average series of first-order difference, and it suggests a practical way to approximate the prediction of the forthcoming value a_{m+1} based on the “current” data a_m and the first measurement value a_1 . For a longer period term, although the error value could be negligible (see eq.(3.9) and eq.(3.11) and Fig 3.3), the accuracy of prediction may still be affected.

At the same time, the cumulative moving average E_m^n can be expressed in term of n^{th} -order difference as:

$$E_m^n = \frac{1}{m}(D_{m-1}^{n+1} - D_1^{n-1}) \quad (3.17)$$

The above equation is proven by mathematical induction (see Appendix B for the whole proof).

Consequently, for a higher order difference, the cumulative moving average

series for a large value L is (where L is a large but finite value):

$$\lim_{m \rightarrow L} E_m^n = \lim_{m \rightarrow L} \left(\frac{1}{m} \sum_{k=1}^m D_k^n \right) = \lim_{m \rightarrow L} \left(\frac{1}{m} (D_{m+1}^{n-1} - D_1^{n-1}) \right) \quad (3.18)$$

Since the n^{th} -order difference series D_m^n is bounded, then there is an existing real limited number C for which $|D_{m+1}^{n-1} - D_1^{n-1}| \leq C$. As a result,

$$\lim_{m \rightarrow L} E_m^n = \lim_{m \rightarrow L} \left(\frac{1}{m} \sum_{k=1}^m D_k^n \right) = \lim_{m \rightarrow L} \left(\frac{1}{m} (D_{m+1}^{n-1} - D_1^{n-1}) \right) \rightarrow \frac{C}{L} \quad (3.19)$$

and Fig 3.4 shows a map of the limit of the moving average series E_m^n , where $\{m | 1 \leq m \leq 100\}$ and $\{n | 1 \leq n \leq 100\}$ for a simple demonstration:

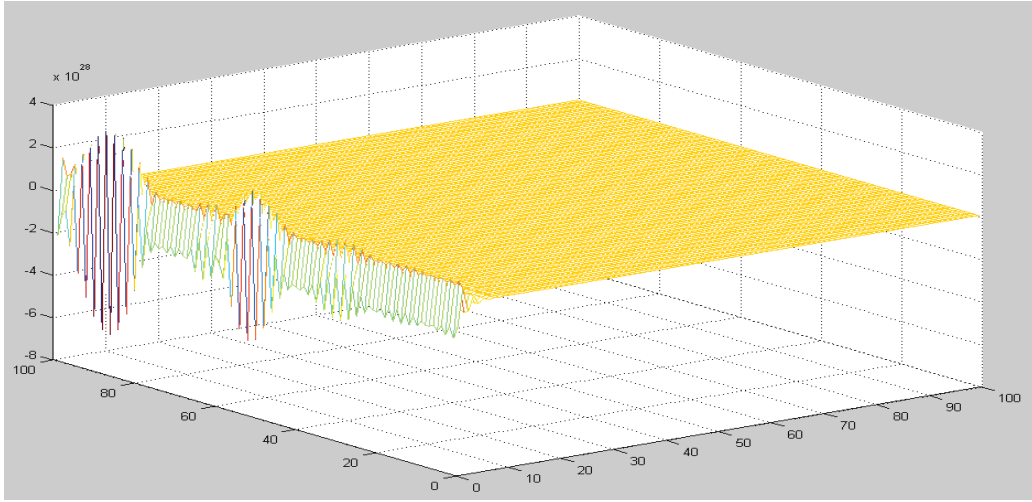


Figure 3.4: A Map of Moving Average of n^{th} -order Difference's Limit (see eq.(3.19)) for Sunspot Number Time Series Data Set; X Coordinate Axis Lists the Variable $m \in [1, 100]$, Y Coordinate Axis Illustrates the Variable $n \in [1, 100]$ and Z Coordinate Axis Shows the Values of Moving Average.

The time series prediction algorithm of moving average based on n^{th} -order Difference (MANoD) below implements the results described above for a general time series input A (Table 3.2)

Table 3.2: Pseudo-code for Algorithm of Moving Average based on n^{th} -order Difference (MANoD)

```

INPUT: An Initial General Time Series Data Set;
METHOD: Moving Average based on nth-order Difference (MABOND);
OUTPUT: Predicted Time Series Data;

```

```

01. // Input the time series data set;
02. SET A[ ] to READ(An Initial Original Time Series Data Set)
03. // L records the size of data sequence;
04. SET L to the length of A[ ]
05. // Calculate the nth-order difference D[ ] of A[ ]
06. SET counter to 0
07. WHILE counter < L-n
08.     SET D[counter] from CALCULATE difference of A[ ]
09. ENDWHILE
10. // Compute the moving average E[ ] of D[ ];
11. SET counter to 0
12. FOR each of D[ ]
13.     SET sumTemp to 0
14.     FOR each term of D[0] to D[counter]
15.         SET sumTemp to sum of term of D[0] to D[counter]
16.     ENDFOR
17.     SET E[counter] to divide sumTemp by counter
18.     INCREASE counter
19. ENDFOR
20. // Get the error value by using ANN;
21. GET error from ANN(E[ ])
22. // Give two values Ln and Lm for Finding Function inputs
23. SET Ln and Lm
24. FOR n = 0 to Ln
25.     FOR m = 0 to Lm
26.         SET F[n][m] to COMPUTE Finding Function result
27.     ENDFOR
28. ENDFOR
29. SET Do[n][m] to COMPUTE from A[ ]
30. GET Theta[n][m] to ||F[n][m] - Do[n][m]||
31. GET (m,n) from find(Theta == min(Theta[n][m]))
32. GET the prediction value based on (m,n)
33. OUTPUT(predicted value)

```

3.4.2 Finding Suitable Parameters for Increasing Precision of the Prediction Algorithm

From the demonstration at the last sections, there are two formulae on the difference operators:

$$D_m^n = \sum_{i=0}^n (-1)^{n-i} C_n^i a_{m+i} = D_{m+1}^{n-1} - D_m^{n-1} \quad (3.20)$$

$$\sum_{j=1}^m D_j^n = D_{m+1}^{n-1} - D_1^{n-1} \quad (3.21)$$

Based on the above results, the moving average becomes:

$$E_m^n = \begin{cases} E_{m-1}^n + \varepsilon & \text{if } \varepsilon \neq 0 \\ E_{m-1}^n & \text{if } \varepsilon \cong 0 \end{cases} \quad (3.22)$$

Then, take eq.(3.20) and eq.(3.21) into eq.(3.22):

$$\begin{aligned} E_m^n &= E_{m-1}^n + \varepsilon \\ &\Downarrow \\ D_{m+1}^{n-1} &= \frac{m}{m-1}(D_m^{n-1} + (m-1)\varepsilon) - \frac{1}{m-1}D_1^{n-1} \end{aligned} \quad (3.23)$$

Next, let $n = n - 1$ into above equation, so that:

$$D_{m+1}^n = \frac{m}{m-1}(D_m^n + (m-1)\varepsilon) + \frac{-1}{m-1}D_1^n \quad (3.24)$$

Thus, the two coefficients in eq.(3.24) can be treated as two special “weights” related to two difference terms of the same order difference level, and they depend on the “start” and the “end” period’s series values, D_m^n and D_1^n .

For increasing the prediction precision of MANoD algorithm, the accuracy, when there $\varepsilon \neq 0$ in eq.(3.22), is proposed to be approximated by Artificial Neural Network (ANN) [Minsky and Papert, 1969] (via “Back-Propagation” Method [Werbos, 1994]) for predicting for the next period, error ε in cumulative moving average of n^{th} -order difference algorithm (see eq.(3.22) and eq.(3.24)).

The moving averages E_m^n and E_{m-1}^n are the inputs of a Feed-Forward Artificial Neural Network (FFANN) with three layers. The trained network is able to get moving average value E_{m+1}^n for further error approximation (see Fig 3.5 for the details), which is used for the 2-inputs and 1-output ANN Back-Propagation training algorithm for 1000 epochs.

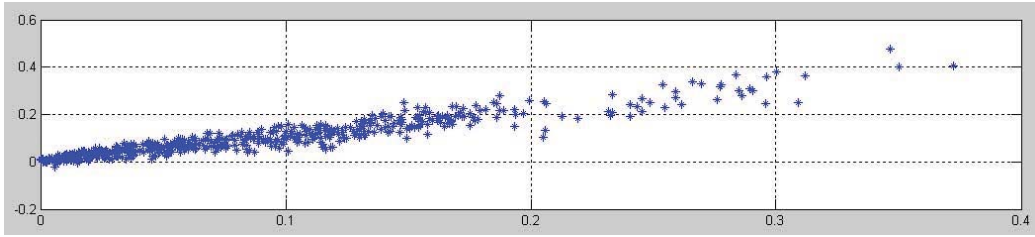


Figure 3.5: Analysis and Prediction for Error (ε) with Artificial Neural Network with E_m^n (1000 samples and $n = 10$) and E_{m+1}^n (1000 samples and $n = 10$). There is a linear correlation relationship between the two variables, E_m^n and E_{m+1}^n , therefore, the error (ε) for next term prediction can be approximated by ANN.

The target of increasing precision for the algorithm’s prediction is to find

suitable values for m (the length value of term input) and n (order level value of difference).

As m is also involved into the prediction of the forthcoming value to be approximated, the choice of parameters (m and n) is key-problem for accuracy and refinement.

Since the second “weight” is a negative value, and its condition number is so high in distribution, eq.(3.24) is not a “normal” weighted function but *ill-conditioned function* in Short Selling Framework [Yuille, 2009]. As a result, with $m \rightarrow \infty$, the function’s variance increase and the variation of function solution(s) could be bigger.

Therefore, the predicted precision may not be good enough. Thus, for a given $k \in [1, m]$, where m is the length of the initial time series data set and ε is unknown yet, based on the eq.(3.24), let:

$$F(k) = \frac{k}{k-1}(D_k^n + (k-1)\varepsilon) + \frac{-1}{k-1}D_1^n \quad (3.25)$$

Then calculate D_{m+1}^n from the original time series data set (eq.(3.2)); and next, obtain Θ representing their Manhattan Distance:

$$\Theta(k) = \|F(k) - D_{k+1}^n\| \quad (3.26)$$

From the result values of array $\Theta(k)$ from the above equation, where the $\Theta(k_{min}) \rightarrow \min$, $F(k_{min})$ is the closest value to the real difference oper-

ator value D_{m+1}^n in the series. The aim is to determine the value m for which a_{m+1} is approximated based on the previous data points in time series, $a_1, a_2, a_3, \dots, a_m$. Fig 3.6 shows an example of Θ_k^n (when $n = 1$ and $k \in [1, 600]$).

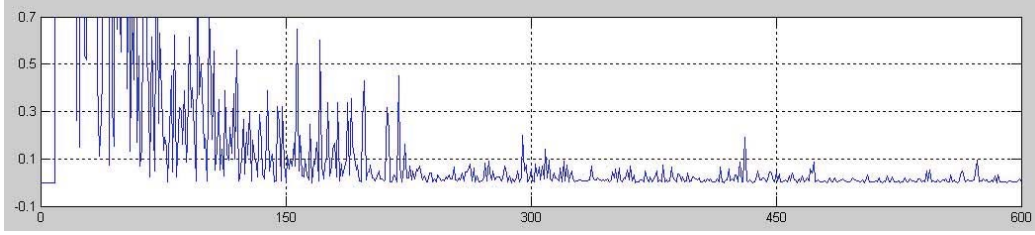


Figure 3.6: The Manhattan Distance Value Series, Θ_m^n , where $n = 1$ and X Coordinate Axis Lists the Values of $m \in [1, 600]$ and Y Coordinate Axis Illustrates the Values for Distance.

According to eq.(3.26), choosing two large values of term index L_m and order level of difference L_n for locating the suitable m and n in the formulae:

$$F_{m \times n} = \frac{m}{m-1} D_m^n + \frac{-1}{m-1} D_1^n \quad (3.27)$$

$$\Theta_{m \times n} = \|F_{m \times n} - D_n^{m+1}\| \quad (3.28)$$

(where $m \in [1, L_m]$ and $n \in [1, L_n]$ in the second formula), in order to identify the area of minimum values.

Based on eq.(3.26) and $\Theta_{m_{\min} \times n_{\min}} \rightarrow \min$, it can be inferred to propose two suitable values for index m_{\min} and n_{\min} for increasing the prediction precision of algorithm MANoD. Fig 3.7 shows a map of matrix $\Theta_{m \times n}$, where

$m \in [1, 500]$ and $n \in [1, 20]$.

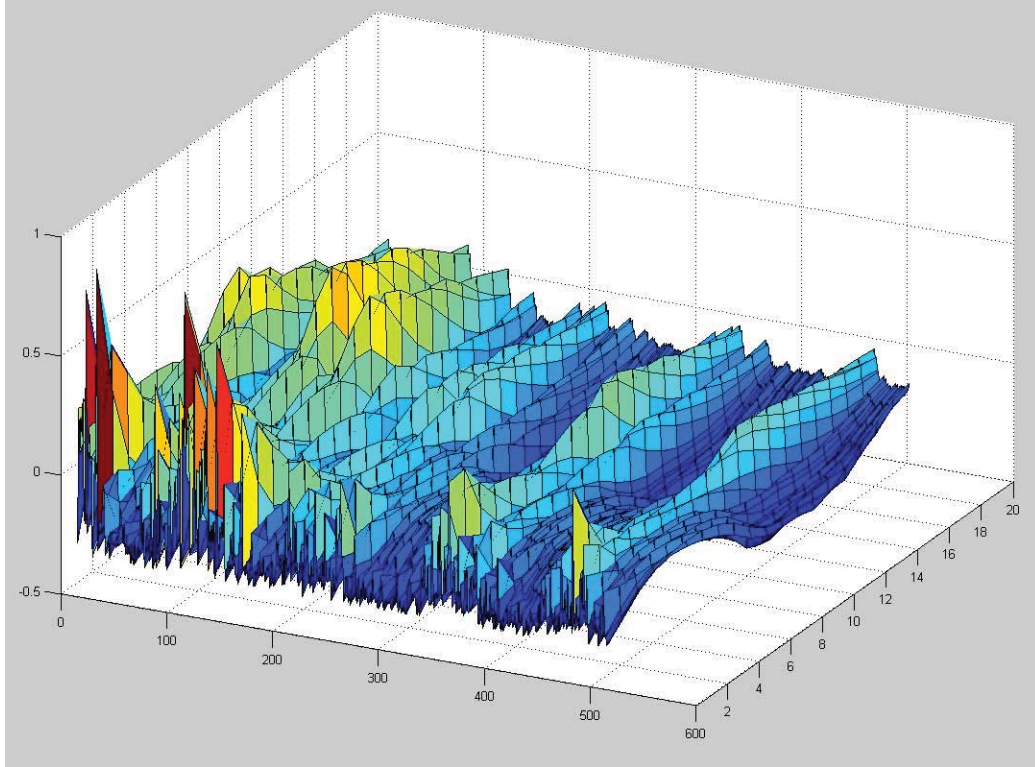


Figure 3.7: The Value Map of Matrix: Θ for Sunspot Number Time Series Data Set (where $m \in [1, 500]$ and $n \in [1, 20]$)

3.5 Case Studies

This section presents the application of the *Moving Average of n^{th} -order Difference* time series prediction algorithm for a monthly average sunspot number time series, a global earthquakes' Richter Magnitude Scale (RMS) time series and a synthetic pseudo-periodical time series (chapter 2 introduced the description of time series data set in details).

• **Sunspot Number Time Series Prediction**

The 1200 monthly average observations of sunspot number time series has been imported into the proposed algorithm.

The following Fig 3.8 shows the initial sunspot number time series values and MANoD prediction results. MANoD produced prediction results very well both on the trends and on values, there are fluctuation in prediction values, but the errors were very small.

Figure 3.9 illustrates the prediction errors ($||\text{Prediction} - \text{Original}||$) by MANoD algorithm.

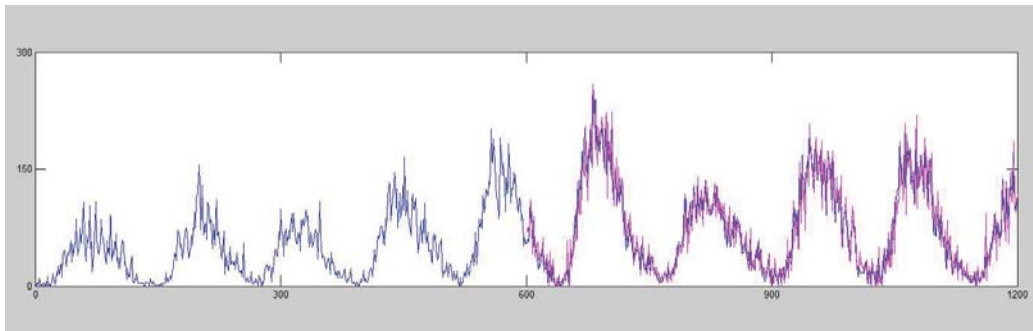


Figure 3.8: The Initial Monthly Average Sunspot Number Time Series and Prediction Results by Algorithm MANoD (where $m = 12$ and $n = 12$); X Coordinate Axis Lists the Index of Time Intervals (1200 Values) and Y Coordinate Axis Illustrates the Original (in blue with 1200 values) and Prediction Values (in purple with 600 values).

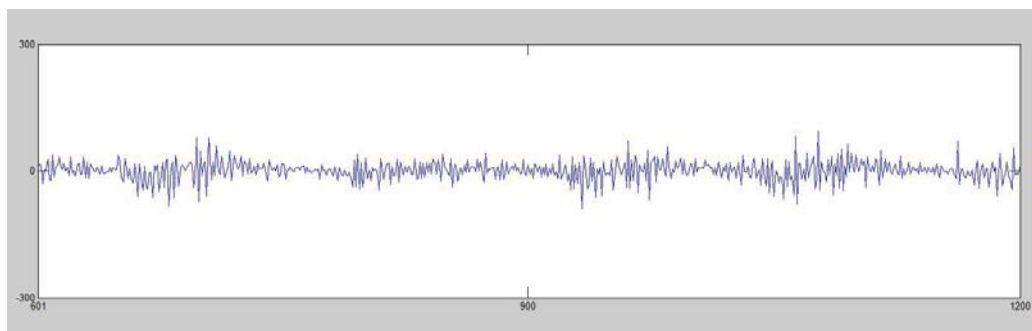


Figure 3.9: Sunspot Number Time Series Prediction Errors (600 data values) by MANoD; X Coordinate Axis Lists the Index of Time Intervals (600 values) and Y Coordinate Axis Illustrates the Prediction Error Values ($||\text{Prediction} - \text{Original}||$) for 600 Values.

• Earthquake Time Series Prediction

A 1351 measurements global earthquakes' RMS time series have been imported into the proposed MANoD algorithm. Fig 3.10 illustrates the original time series and MANoD algorithm's prediction results. MANoD produced a very good trends for predicting Earthquake time series. Figure 3.11 shows the prediction errors ($||\text{Prediction} - \text{Original}||$) by MANoD algorithm.

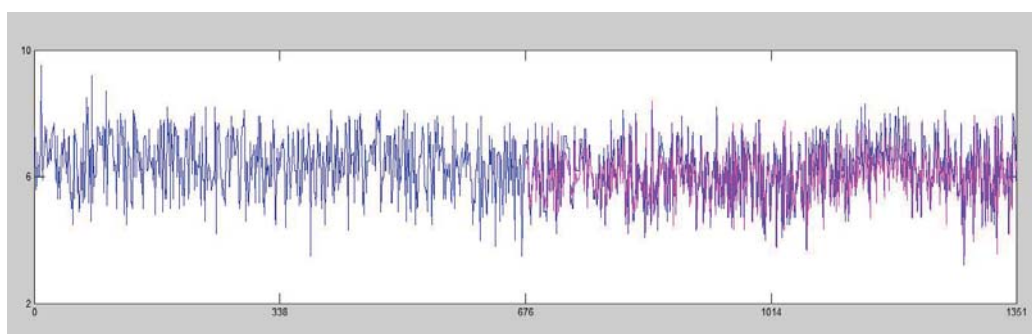


Figure 3.10: The Initial Global Earthquakes' Richter Magnitude Scale (RMS) Time Series and Prediction Results by Algorithm MANoD; X Coordinate Axis Lists the Index of Time Intervals (1351 Values) and Y Coordinate Axis Illustrates the Original (in blue with 1351 values) and Prediction Values (in purple with 676 values).

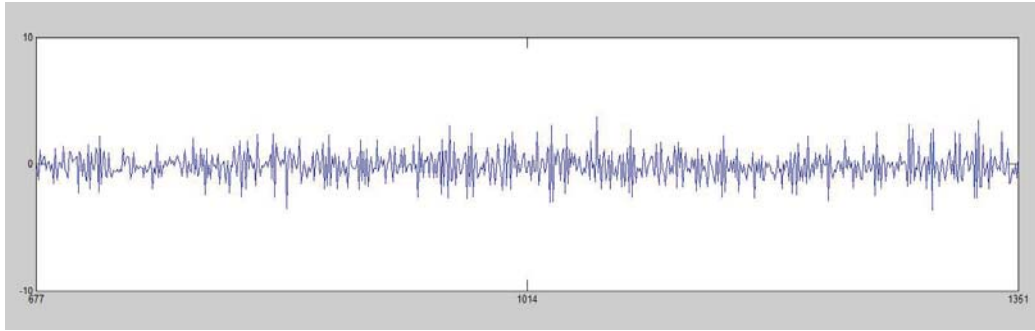


Figure 3.11: Global Earthquakes' Richter Magnitude Scale Time Series Prediction Errors by MANoD; X Coordinate Axis Lists the Index of Time Intervals (676 values) and Y Coordinate Axis Illustrates the Prediction Error Values ($||\text{Prediction} - \text{Original}||$) for 676 Values.

• Synthetic Pseudo-Periodical Time Series Prediction

A generated synthetic pseudo-periodical time series with 100000 values by the mathematical function:

$$\bar{y} = \sum_{i=3}^7 \frac{1}{2^i} \sin \left(2\pi (2^{2+i} + \text{rand}(2^i)) \bar{t} \right), \quad 0 \leq \bar{t} \leq 1$$

has been imported into MANoD.

Fig 3.12 depicts the time series source values and its prediction results by MANoD algorithm. MANoD produced prediction results very well both on the trends and on values for synthetic pseudo-periodical time series.

Fig 3.13 shows the prediction errors ($||\text{Prediction} - \text{Original}||$) by MANoD algorithm.

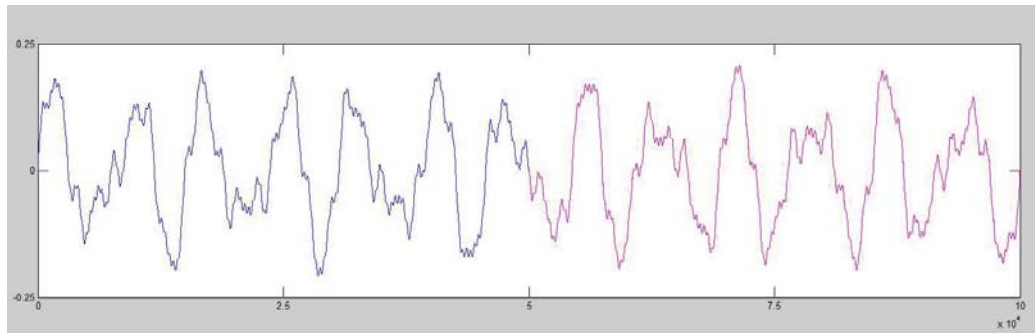


Figure 3.12: The Synthetic Pseudo-Periodical Time Series Source Values and Prediction Results by Algorithm MANoD; X Coordinate Axis Lists the Index of Time Intervals (100000 Values) and Y Coordinate Axis Illustrates the Original (in blue with 100000 values) and Prediction Values (in purple with 50000 values).

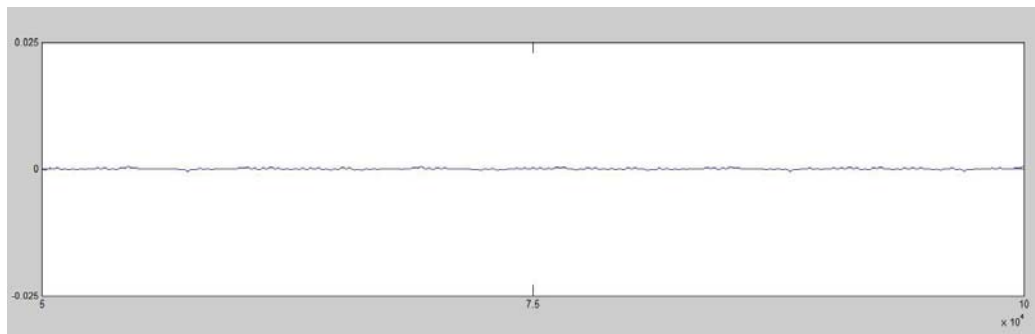


Figure 3.13: The Synthetic Pseudo-Periodical Time Series Prediction Errors by MANoD; X Coordinate Axis Lists the Index of Time Intervals (50000 values) and Y Coordinate Axis Illustrates the Prediction Error Values ($||\text{Prediction} - \text{Original}||$) for 50000 Values.

3.6 Summary

This chapter has introduced a time series prediction algorithm based on *Moving Average of n^{th} -order Difference* and applied it for predicting three different types of time series.

As two core concepts of algorithm MANoD, the definitions of (n^{th} -order) *Difference Operator* and (*Cumulative*) *Moving Average* have been introduced. Then, assembling both together we proposed to establish a computational model and approach for predicting the next value of a pseudo-periodical time series.

MANoD algorithm presents a simple way to determine the range of values necessary for a good prediction of the time series terms in cases of bounded pseudo-periodical time series. The developed algorithm to predict time series based on a number of previous known values necessarily addresses also the noise of the actual collected measurements of a time series. The errors obtained by the algorithm in this thesis are represented as difference between actual and expected value of averages sum (difference of moving average). The method also provides a logical development in a transparent way, avoiding the use of “Black Box” methods.

The limitation of the algorithm MANoD is the dependency of the (still) error between the moving average of n^{th} -order difference values at the prediction step, $n + 1$ and n . The MANoD algorithm generates therefore a good prediction for the trends of the time series (including the pseudo-periodicity), but the precision of prediction (amplitude) suffers because of dependency on how many orders (i.e. value of n) difference have been considered, which increases the complexity calculus though and introduces a tuning parameter of the order of difference. Another direction for further research is the approximation of error in using machine learning techniques, in order to reduce the differences induced by the possibility to obtain a non-zero average

of n^{th} -order difference for a period close to the prediction moment.

The research for MANoD's development and the relevant work has been published as described in section Declaration.

Next chapter will present another original time series prediction approach based on *Series Features Extraction*, and also used for time series prediction.

Chapter 4

A Time Series Prediction Algorithm based on Series Features Extraction

Contents

4.1	Introduction	68
4.2	Time Series Data Classification based on a Combination Rule of Successive Neighbouring Data Points . . .	74
4.3	The Approach of Series Features Extraction Algorithm for Time Series Analysis and Prediction	98
4.4	Case Studies	112
4.5	Summary	116

4.1 Introduction

4.1.1 Epistemology

As one important branch of philosophy study, *Epistemology* is rendered as a “Theory of Knowledge”; it focuses on certain understanding of *Knowledge*. Epistemology is also the investigation into a debate on the knowledge itself and on acquiring knowledge [Britannica, 2008] [IEP, 2008] [SEP, 2008].

Epistemology deals with such questions as how knowledge come from experience or from innate reasoning ability; it concerns with the nature, sources and scope of knowledge, then it attempts to address several basic questions related to knowledge like: what is knowledge? where the knowledge comes from? how to differentiate the *Truth* and *Believe*?

Epistemology, generally, involves a debate on the question of whether knowledge can be acquired *a priori* or *a posteriori*. The analysis of the progress of time series can be seen a regression progress, therefore, it avoids the “experience” knowledge (*a posteriori*) as referred to the effect of the analysis and prediction results. Moreover, epistemology helps to identify that the acquired knowledge in time series belongs to *a priori*.

The first step in time series analysis by epistemology is to determine the nature of knowledge for a time series, which means to obtain the description and understanding of the original data sequences. Then, the second step is to determine the scope of knowledge, which is to differentiate *truth* knowledge

(sets of *truth* information) (*a priori*) and to acquire the *believe* knowledge (*a posteriori*) for a view to control the development time series.

Consequently, the author considered that estimation approaches of time series analysis and prediction as epistemology methods require a method based on *a priori* knowledge rather than *a posteriori* knowledge to make use of the essence stored in the time series.

4.1.2 A Priori and A Posteriori Knowledge

Terms “*a priori*” and “*a posteriori*” help to distinguish between two opposite types of knowledge. Table 4.1 shows the definition of them [Gensler, 2001] [Dickie, 1996] [Scruton et al., 2001].

Table 4.1: The Definition of *a priori* and *a posteriori* Knowledge

	is knowledge independent of sensory experience.
A Priori	e.g. “All bachelors are un-married.”
Knowledge	“One time series data point value is higher than another.” “The inner construction of a time series data set.”
<hr/>	
	is knowledge dependent of sensory experience.
A Posteriori	e.g. “Some bachelors are happy.”
Knowledge	“This time series data point value is too high.” “The exhibition of a time series data set.”

Hence, *a priori* knowledge is non-empirical received beforehand; it is considered a top priority of Logic and Mathematics domain and focuses on abstract and formal objects. *A posteriori* knowledge is empirical and received afterwards.

In a time series database, *a priori* knowledge is referred to the attributes of data, i.e. value (mathematical knowledge) and value comparison (mathematic and logical knowledge) and so on. *A posteriori* knowledge is referred to the judgment of data characteristics, i.e. the fluctuation of the assemble in the first half sequence of values.

4.1.3 Features, Patterns and Model

Features are the individually measurable heuristic properties of the phenomena being observed. The concept of feature in fact specifies a structure in data sequence, which may include a simple measurement or complex structures/objects. In terms of applications, the use of features extraction are well developed in regression analysis, statistical pattern recognition, computer vision, and so on.

In essence, the features are series of information before the initial data measurements are observed, which are *a priori* information and knowledge. They may be categorized into natural classes based on their distinctive features; each of them describes a quality or characteristic of the natural class, for example, (mathematical) thresholds of time intervals and time series data

sets, etc.

The type of features that are related by all their membership to a common theme is a number of terms for the joint characteristic, and a successive set of features in time series may include the changing and/or trends of time series data over time intervals. That phenomenon of cyclic appearing can be considered as a simple of pattern.

The term “pattern” is usually used as an aid to design a “model”. In other words, this concept of having collection(s) of patterns in a specific order or recurrence form is useful to construct a summary of the initial data series. In time series analysis, a pattern for a variable identifies a subset of all (possible) measurements, and a successive set of patterns consist of a new data sequence; this new one can be treated as a description data set for the initial data sequence. On the other side, data classification procedure also gives patterns into groups. Those patterns (groups) have common one or more characteristics, such as: data attributes, variables, and so on.

A model can be formally defined as a set of data elements and relationships among data sets. However, according to the quantity of *a priori* knowledge (or information) included, a modelling process is classified into “White Box” and “Black Box” models [Beizer, 1995]. It is considered preferable to use a “White Box” model to make the model’s description comprehensible, because of the *a priori* knowledge (information) root in the data sets. On the other hand, “Black Box” models call *a posteriori* knowledge (information), which is dependent of experience, to estimate the relationship among the data or

between parameters of model itself. For a huge or outstretching time series database, a “Black Box” model will increase the computational difficulty and complexity, while the most important problem is that results are normally in-comprehensible.

4.1.4 The Methodology of Series Features Extraction Approaches

In a small data set, the features would be easily identifiable, for example, size, structure and everything else of original data sequence after understanding and describing data series. However, for a huge data set with a series data attributes, there may simply be un-known factors that affect the data. This requires to identify features existing as hidden pieces of entire database. As a result, the extraction of a series features provides additional information for searching patterns and constructing the predictive model.

Relative to the concept of model, pattern is a “local” summary for one or more pieces of data sequence. Based on *a priori* knowledge, pattern recognition aims to classify data, either to classify measurements into groups or extract patterns from classified groups. A feature extraction system computes the numeric or symbolic information from the observations; features that contain a common significance consists of a pattern (this pattern relies on the extracted features but may show different expressions over different time intervals in difference time series).

In the sense of *un-supervised* (machine) learning, a pattern recognition scheme is not always given *a priori* labeled features and/or patterns, however, it is able to establish classes with *a priori* features if the features extraction system exports *a priori* knowledge only based on the mathematical and logical methods. This pattern recognition mechanism also deals with *a priori* knowledge independent of the human sensory experience; the model can, for the entire contents of the initial time series data sets, explicitly determine the nature and meaning of data.

In fact, a model is also a pattern. It is designed to show the main significance of data as a “global” summary of the entire data sequence. As opposing to an un-structured data sets, a modelling process describes the representation and access of data. As a common problem, modelling also faces the challenge of choosing the source of knowledge between *a priori* and *a posteriori* knowledge. Classical methods mainly limited to in using mathematical model(s) for analysis and prediction refer to the length of data series, meanwhile, the performance of the methods relate to the complexity of data structure (or data dimensionality). In addition, the modern learning methods try to take into account the difficulties of requiring *a priori* knowledge.

Therefore, to establish such a model which only deals with *a priori* knowledge extracted from the initial time series, will be a big step forward in time series analysis and prediction. The prerequisites, such an approach, requires purely to process *a priori* knowledge.

4.2 Time Series Data Classification based on a Combination Rule of Successive Neighbouring Data Points

4.2.1 Data Classification for a Generic Data Sequence Set

Data Sequence Classification is the procedure in which sub-sets, several data points as a group, or even each individual data point, are placed into “Classes” (also named “Groups”) based on quantitative information or foundation knowledge on one or more characteristics inherent in the initial data sequence set (referred to as attributes, variables, characters, etc).

There are two steps in classification [Kotsiantis, 2007] [Kotsiantis et al., 2006]. The first is to establish/choose a classifier, which will be trained to describe existing data (or training sets); then, the next step is to use the generated model to classify previously unseen data/samples.

A classifier is a model that describes structural features and behavior of a given data collection. A good classifier should produce a category, with minimum quantity classes, of elements that have common features from the initial data sets; and this category should presents all significant samples in the original data sequence.

Given an un-marked original data sequence, where the measurements of data

represent the known information over time (*a priori* knowledge), the aim is the construction of successive data points. In other words, the use of neighbouring measurements is to compose a primary pattern.

Since time series prediction is the use of a model to predict future event(s) based on the past known event(s), a combination point of three successive neighbouring data points contains the most straightforward knowledge (also *a priori*), which are: data from previous, data at present and at the next moment.

We propose to use the combination point of three successive neighbouring data point as the primary element of our new approach, as described below.

4.2.2 Combination of Time Series Data Points

If using one single data item to represent the three neighbouring data, this procedure can be treated as the first step for modelling of data series; then, this type of combined data is able to reveal a “trend” of the original data series.

In a more basic sense, this kind of combination rule can be applied to any types of time series and it could be mostly commonly used within time series to smooth out shorter term fluctuations and highlight longer term’s trend and/or cycle period. This combination of short term and long term aims depends on the application and parameters (of classifier) which will be set up accordingly.

For example, for business and economic time series data set with seasonality, the combination series of three month measurements should include more meaningful significance or values than one month data series. Another example is the stock market time series data set, which always exhibits an up/down trend in short term period and contains cycle period(s) in long term, thus, the combination of neighbouring simplifies the data series construction and smoothes out their fluctuations.

Given an original and un-marked time series with known values:

$$A = \{a_1, a_2, \dots, a_{t-1}, a_t, a_{t+1}, \dots\} \quad \text{where } t \in \mathbf{N}$$

there are three successive neighbouring data points at moments L, C, N

$$\begin{aligned} a_L &\in A : \text{the Lastest data;} \\ a_C &\in A : \text{the Current data;} \\ a_N &\in A : \text{the Next data;} \end{aligned} \tag{4.1}$$

where $L + 1 = C = N - 1$.

In view of all possibilities based on the differences between data values, there are totally 13 correct group cases for all reliable classes, and each group (combining data points) represents a different 2-Dimensional shape as defined by the sequence of 3 values in the original time series (where the horizontal axis means the time dimension t , and the vertical axis represents the data values a_t , see Table 4.2 for more details).

Therefore, we propose a projection of all time series values in a 3-Dimensional space of consecutive differences in order to study the possible clusters of these data according to their shown direction over a time interval. The other missing 14 distinct cases (out of all 27 possible combinations of the three coordinates in the consecutive differences space) define all impossible cases from the point of view of time series values. For example the case $(-, 0, -)$ gives $a_C = a_N, a_C < a_L$ and $a_L < a_N$ which gives simultaneously $a_C = a_N$ and $a_C < a_N$ which is an impossible case. All other cases follow a similar treatment.

Ideally, there is a 3-dimensional space (Fig 4.1) to express the difference between three successive data point in eq.(4.2); and let X -dimension denotes the value of $a_C - a_L$; Y -dimension presents the value of $a_C - a_N$; Z -dimension shows the value of $a_L - a_N$. Therefore, any point in that space with coordinate $P(x, y, z)$ is defined by the differences of the combination data set generated from original time series. For example, $P(0, 1, 2)$ means that: $a_C - a_L = 0$, $a_C - a_N = 1$, $a_L - a_N = 2$.

The Table 4.2 shows the categorization of combination rule (note: in the column “condition”, “+” means the first value bigger than the second one; “-” means the first value smaller then the second one; and “0” means the first value equals the second one. The last column “Shape” denotes the geometric sketch).

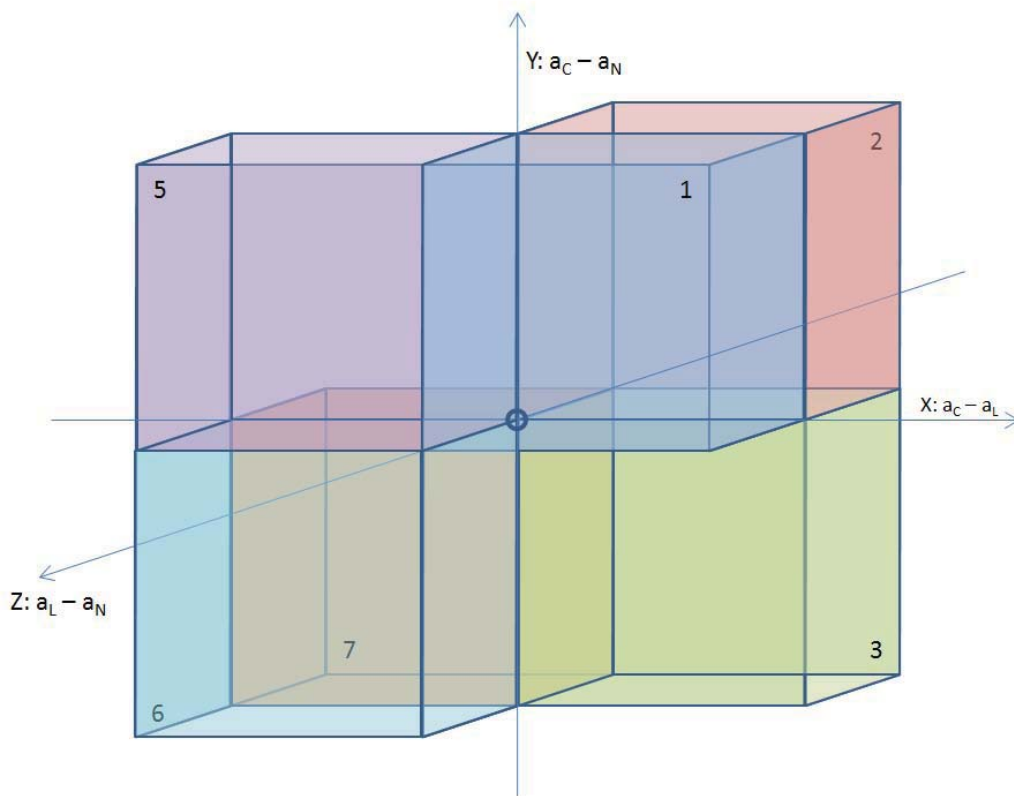















Figure 4.1: Difference of 3 Successive Neighbouring Data in a 3-Dimensional Space

Table 4.2: Categorization of Definition for Combination (3 Successive Data)
Rule of Grouping 13

Grouping 13	Condition			Skeleton Shape
	$a_C - a_L$	$a_C - a_N$	$a_L - a_N$	
Group 1	+	+	-	
Group 2	+	0	-	
Group 3	+	-	-	
Group 4	+	+	0	
Group 5	+	+	+	
Group 6	0	-	-	
Group 7	0	0	0	
Group 8	0	+	+	
Group 9	-	-	-	
Group A	-	-	0	
Group B	-	+	+	
Group C	-	0	+	
Group D	-	-	+	

In the Fig 4.1, based on the combination rule showed in Table 4.2, each group locates in a specific space, such as: a cube interspace (3-Dimension) or a plane (2-Dimension). The definition of each group's position is provided in Table 4.3

Table 4.3: The 3D Sub-domain Correspondence of Grouping 13 Cases

	Sub-domain Correspondence	Exclude
Group 1	rear top-right cube	red plane
Group 2	blue plane	the origin
Group 3	rear bottom-right cube	green plane
Group 4	red plane	the origin
Group 5	front top-right cube	red, blue & purple plane
Group 6	green plane	the origin
Group 7	the origin (in black)	–
Group 8	purple plane	the origin
Group 9	rear bottom-left cube	aqua, green & orange plane
Group A	aqua plane	the origin
Group B	front top-left cube	purple plane
Group C	orange plane	the origin
Group D	front bottom-left cube	aqua plane

4.2.3 Optimizing the Categorization

We proposed in the previous section a scheme of 13 classes (groups), in which the data sequence points of the initial time series can be classified into, based on the classification of combinations for three successive neighbouring data points.

We call this new data series the **Combination Data Series (CDS)**. Each point in the newly generated combination data series can be treated as the outcome value from a *Discrete Random Variable*, and there is no relationship between any two consecutive combination data points.

As a result, the probability distribution of that combination data series is a discrete distribution (see definitions below); any new forthcoming data to join the combination data series also obeys that rule.

Definition 4.1 - Discrete Probability Distribution:

Discrete probability distributions have the values to be observed restricted within a pre-defined list of possible values. This list has either a finite number of members, or at most is countable.

The distribution of a given random variable $X = \{x_1, x_2, \dots, x_n\}$ is discrete (Discrete Probability Distribution) if the probability function $P(x_i)$ defined over $i = 1, 2, \dots, n$, has a distribution function:

$$\mathcal{F}(X) = \sum_{i=1}^n P(x_i) = 1 \quad (4.2)$$

Definition 4.2 - Discrete Random Variable:

A *Discrete Random Variable* is a random variable which can only take a finite number of distinct values and is characterized by a *Discrete Probability Distribution*: given $t \in \mathbf{Z}$, the countable values:

$$X = \{x_1, x_2, \dots, x_t\}$$

is a *Discrete Random Variable*.

The combination rule for “Grouping 13” is defined in Table 4.2, the set of output values is a *Discrete Random Variable* set, with a finite number of elements (13). Then its probability function is:

$$\mathcal{F}(G) = \sum_{i=1}^D P(G_i) = 1 \quad (4.3)$$

where $0 \leq P(G_i) \leq 1$ and $i \in \{1 \sim 9, A, B, CD\}$.

We tested the suitability for further implementation of *Grouping 13* distribution on five different time series data sets (see Table 4.4 for details).

Table 4.4: Five Samples of Time Series for Testing

Index	Time Series Sample	Measurements	Figure
1.	Earthquake RMS (global)	1351	Fig 4.2
2.	Foreign Exchange Rates (GBP to USD)	2295	Fig 4.3
3.	Nile River Low Flows	570	Fig 4.4
4.	Sunspot Number (Monthly Average)	1200	Fig 4.5
5.	Synthetic Pseudo-periodical Data Series	100001	Fig 4.6

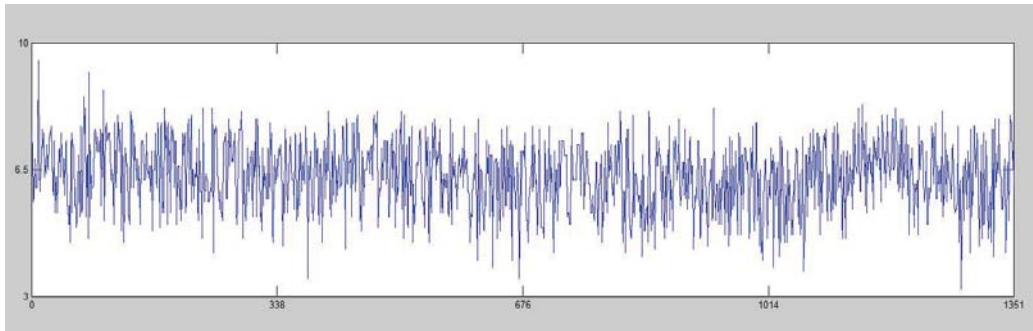


Figure 4.2: Earthquake RMS Testing Time Series, X Coordinate Axis Lists the Index of Time Intervals and Y Coordinate Axis Illustrates Earthquakes RMS Values.

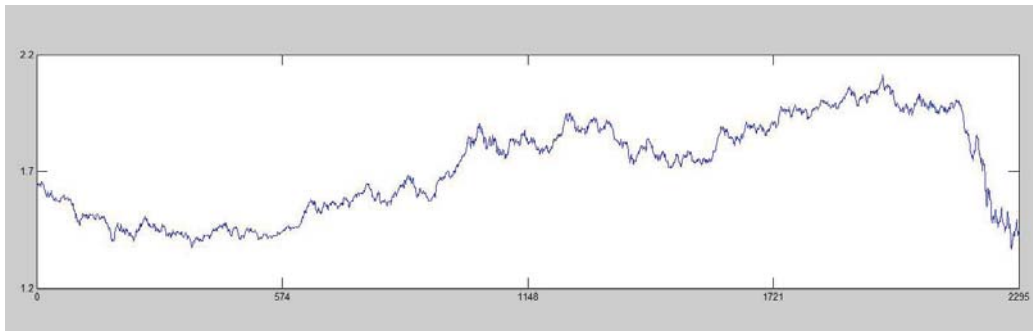


Figure 4.3: Foreign Exchange Rates Testing Time Series; X Coordinate Axis Lists the Index of Time Intervals and Y Coordinate Axis Illustrates Foreign Exchange Rates Values.

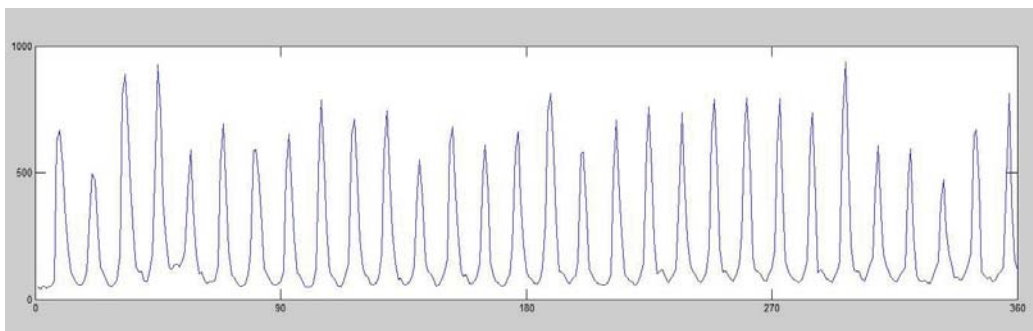


Figure 4.4: Nile River Low Flows Testing Time Series; X Coordinate Axis Lists the Index of Time Intervals and Y Coordinate Axis Illustrates Nile River Low Flows Values.

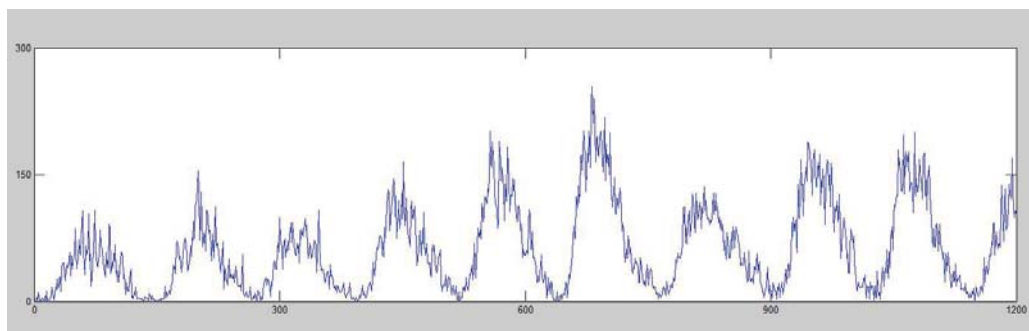


Figure 4.5: Sunspot Number Testing Time Series; X Coordinate Axis Lists the Index of Time Intervals and Y Coordinate Axis Illustrates Sunspot Number Values.

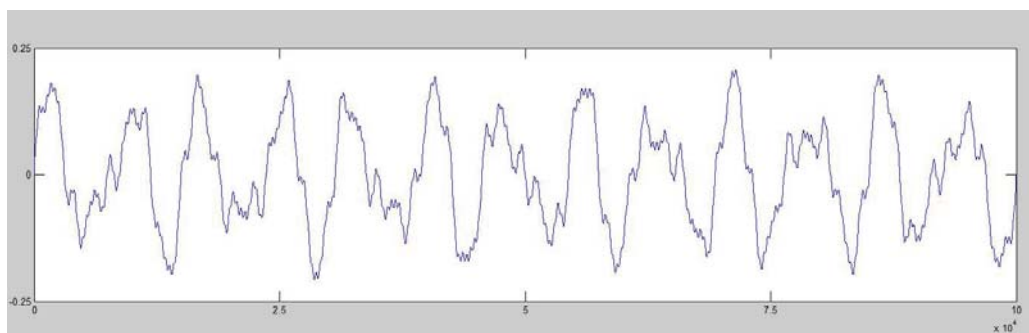


Figure 4.6: Synthetic Pseudo-Periodical Testing Time Series; X Coordinate Axis Lists the Index of Time Intervals and Y Coordinate Axis Illustrates Synthetic Pseudo-Periodical Time Series Data Set Values.

Table 4.5 shows some statistical details about the 13 codes (groups) occurrences into the transformed combination data series, where “Q.#” shows quantities of each corresponding group from the combination series; “P.#” expresses the proportions of each group (“Q.#”) into the total combination data set samples. For example, in the first row (Q.1), the value “172” presents that there are “172” samples belonging to “Group 1” whereas $P.1 = 0.13 = 172/1349$.

Table 4.5: Statistical Results of Combination Rule for Grouping 13

	Group Index												Total	
	1	2	3	4	5	6	7	8	9	A	B	C		D
Q.1	172	65	178	18	156	49	31	69	176	19	196	53	167	1349
P.1	0.13	0.48	0.13	0.13	0.12	0.36	0.23	0.51	0.13	0.14	0.15	0.39	0.12	1
Q.2	301	9	570	1	278	11	0	8	301	5	528	10	271	2293
P.2	0.13	4e-3	0.25	4e-4	0.12	4e-3	0.00	4e-3	0.13	2e-3	0.23	4e-3	1e-3	1
Q.3	70	19	103	6	71	18	3	15	68	9	102	14	70	568
P.3	0.12	0.03	0.18	0.01	0.13	0.03	5e-2	0.03	0.12	0.02	0.18	0.02	0.12	1
Q.4	174	7	233	4	166	6	0	4	182	1	257	3	161	1198
P.4	0.15	6e-3	0.02	3e-3	0.14	5e-3	0.0	3e-3	0.15	8e-4	0.21	3e-3	0.13	1
Q.5	63	0	50064	0	70	0	0	0	63	0	49669	0	70	99999
P.5	6e-4	0.0	0.50	0.0	7e-4	0.0	0.0	0.0	6e-4	0.0	0.50	0.0	7e-4	1

As can be seen in Table 4.5 on Grouping 13 codes occurrences, it is possible that there are no values in some classes for specific time series, such as: Group 7 in row Q.2, Group 2, 4, 6, 7, 8, A, C in row Q.5.

Of course, the groups with few or zero elements are different for various time series: for example, Groups 2, 4, 6, 7, 8, A, C contain 0 elements for synthetic pseudo-periodical time series (see Q.5 and P.5) but the same codes contain different (positive) quantities of elements for other time series (see Q.1 and

P.1)

Therefore, it is very necessary at this stage to propose as a second step the classifier code optimization - starting from the initial 13 Grouping codes, we need a new classification combination data sequence, which obeys a fixed distribution, and provides more effective information and makes use of (*a priori*) knowledge from the original time series.

For an ideal discrete probability distribution, the values of each output elements' probability should be greater than 0. Meanwhile, the number of output classification codes should be kept to a minimum whereas the number of occurrences in each class should be positive; that is to say: (μ is the value of probability)

$$\Omega = \{\omega : X(\omega) = \mu_i\} \quad i = 1, 2, 3, \dots \quad (4.4)$$

$$\mathcal{F}(X) = \sum_{i=1}^{\min} P(X = \mu_i) I_{\Omega}(\Omega_i) = 1$$

where I_{Ω} is the *Indicator Function* of Ω .

That means a better classification set can be designed to produce a minimum number of classes with a positive (preferably maximum) number of elements from the original data sequence, ideally. In other words, if any group's proportion tends to zero, that means there are no output values belonging to this pre-defined class (group) from the input discrete random variable. Thus, for this case, it has been obtained a meaningless class (group), which should be not contained in the classification codes.

Therefore, the initially proposed Grouping 13 combination rule set can be used to project numerical data sets into the transformed data series based on translation of groups of three sequential values according to Table 4.2. However, it should be optimized in terms of number of codes/classes as discussed above if the distribution of codes in the generated data series requires re-consideration.














We propose a new grouping combination rule, named “Grouping 07”, constructed as follows:

The new class “Group 4” contains groups 2, 4, 6, 7, 8, A, C from Grouping 13 (because these were groups having their codes based on just one or two distinctive consecutive values i.e. a horizontal line is part of or a good approximation of the graphical representation of the group values).

Other new classes are labeled “Group 1, 2, 3” and “Group 5, 6, 7” as related to the remaining groups from Grouping 13.

In geometric representation, Group 1, 2, 3 and Group 5, 6, 7 represent the fluctuations of consecutive values of initial time series, whereas Group 4 depicts a flat behavior. As a result, the optimized combination “Grouping 07” is proposed, and its classes are presented below in Table 4.6:

Table 4.6: Definition of Classes for Combinations based on 3 Successive Values in Grouping 07

Grouping 07	Conditions			Shape
	$a_C - a_L$	$a_C - a_N$	$a_L - a_N$	
Group 1	+	+	+	
Group 2	+	+	-	
Group 3	+	-	-	
Group 4	+	0	-	
	+	+	0	
	0	-	-	
	0	0	0	
	0	+	+	
	-	-	0	
	-	0	+	
Group 5	-	+	+	
Group 6	-	-	+	
Group 7	-	-	-	

Based on the combinations presented in Table 4.6, the Grouping 07 can also be depicted in 3-dimensional space (like we represented Grouping 13 previously): Fig 4.7 shows the classes for Grouping 07 using the 3D coordinates described at the start of section 4.2.2.

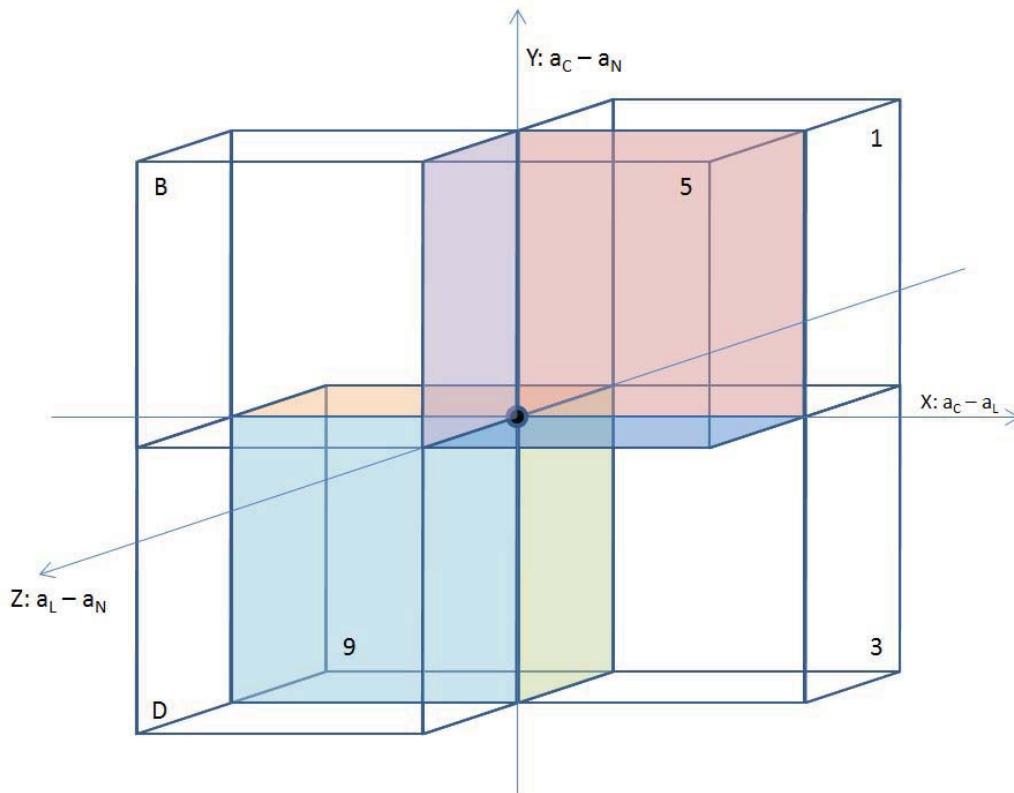


Figure 4.7: Difference of 3 Successive neighbouring Data in a 3-Dimensional Space

In this 3-dimensional space (Fig 4.7) the coordinates allow us to represent the different relative positions between groups from “Grouping 07” combinations, where the X-dimension represents the difference of “ $a_C - a_L$ ”; the

Y-Dimension represents the difference of “ $a_C - a_N$ ”; and the Z-Dimension shows the difference of “ $a_L - a_N$ ”. A point “ $P(x, y, z)$ ” expresses the class with a particular sequence of time series values belongs to the Grouping 07’s (see details in Table 4.7):

Table 4.7: The 3D Sub-domain Correspondence for Grouping 07 classes

	Sub-domain Correspondence	Exclude
Group 1	front top-right blue cube	the origin, X, Y, Z axis
Group 2	rear top-right red cube	the origin, X, Y, Z axis
Group 3	rear bottom-right green cube	the origin, X, Y, Z axis
Group 4	the origin (in black), X, Y, Z axis	–
Group 5	front top-left purple cube	the origin, X, Y, Z axis
Group 6	front bottom-left aqua cube	the origin, X, Y, Z axis
Group 7	rear bottom-left orange cube	the origin, X, Y, Z axis

Consequently, the differences between the initial Grouping 13 definition and the new Grouping 07 are given by the coding of those groups of two or three data points with same value. Grouping 13 takes into account all possible cases about three successive neighbouring data points, and therefore the intention of this classifier’s definition is to cover all possible 3-point sequences. It defines a class for each case of time series consequent values comparisons, and associates with every class (group) an unique geometrical shape.

However, Grouping 13 also produces some empty/void classes (groups) for a generic time series considered. The number of classes (groups) with no elements could affect the probability distribution, and then they could increase the processing complexity of data sequence analysis and the prediction algorithm later.

The Grouping 07 method is, from the same perspective, defined to review the movement and trend of initial data sequence. It does not take anymore into consideration the value comparison of time series points but focuses onto local data movement, for example, up/down trend of the time series.

In other words, The Grouping 07 indicates if there is a continuous although local fluctuation (or not) of the initial time series, such as: Group 1, 2, 6, 7 relate to signal fluctuation, Group 4 relates to steady signals or changing to steady values, and Group 3, 5 tell that initial time series' trend is up or down, respectively. Meanwhile, succession of new data groups may define local *peak* situations, e.g. Group 3 followed by 5. Similarly, a sequence of Group 5 followed by Group 3 may describe a local *valley* case.

Consequently, Group 2, 4, 6, 7, 8, A, C of the Grouping 13 approach can be amalgamated together into a single Group 4 in the new combination approach Grouping 07. Table 4.8 presents the corresponding relationships between Grouping 13 and Grouping 07.

Also, in geometrical interpretation, Grouping 07 merges groups from the previous Grouping 13 approach with same geometric shape interpretation into one class. For example, given three consecutive points from the initial time

Table 4.8: Correspondence between groups of Grouping 13 and Grouping 07 approaches

Categorization of Grouping 07	Categorization of Grouping 13
Group 1	Group 5
Group 2	Group 1
Group 3	Group 3
Group 4	Group 2, 4, 6, 7, 8, A, C
Group 5	Group B
Group 6	Group D
Group 7	Group 9

series: A_L , A_C , A_N and t representing the interval over the time dimension of the time series, let's suppose A_C is located in the origin of a Cartesian Coordinate System. Thus, there are two vectors, $\vec{\alpha}$ and $\vec{\beta}$, related to the movement and trend of the original time series:

$$\vec{\alpha} = \overrightarrow{A_L A_C} = \langle \Delta t, V_{LC} \rangle \quad (4.5)$$

$$\vec{\beta} = \overrightarrow{A_C A_N} = \langle \Delta t, V_{CN} \rangle \quad (4.6)$$

where the Δt is defined as a scalar value of difference the time intervals and V denotes the vertical projection of each vector.

Then, the 2-dimensional inner product of our two vectors is:

$$\vec{\alpha} \cdot \vec{\beta} = \Delta t^2 + V_{LC} V_{CN}$$

So far, if there are any two or three equal values of data points from original time series, then either or both values of V_{LC} or V_{CN} are 0, so that the scalar of inner product becomes a fixed value: Δt^2 .

The two types of combination rules, the performance of “Grouping 13” and “Grouping 07”, are compared below by using the concepts of *Discrete Random Variable (Weighted) Mean* and *Discrete Random Variable (Weighted) Standard Deviation* [Kallenberg, 2002] [Papoulis and Pillai, 2002].

Definition 4.3 - Discrete Random Variable (Weighted) Mean (DRV Mean):

For a given discrete random variable $X = \{x_1, x_2, \dots, x_n\}$ and its corresponding probabilities $P = \{P(x_1), P(x_2), \dots, P(x_n)\}$, the Discrete Random Variable Mean μ is calculated as:

$$\mu = \frac{\sum_{i=1}^n x_i P(x_i)}{\sum_{i=1}^n P(x_i)}$$

where the series of probabilities P are taken as weights.

Based on the eq.(4.2), the DRV Mean becomes:

$$\mu = \sum_{i=1}^n x_i P(x_i)$$

Definition 4.4 - Discrete Random Variable (Weighted) Standard Deviation (DRV S.D.):

For a given discrete random variable $X = \{x_1, x_2, \dots, x_n\}$ and its corresponding probabilities $P = \{P(x_1), P(x_2), \dots, P(x_n)\}$, the Discrete Random Variable Standard Deviation is calculated as:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \mu)^2 P(x_i)}{\sum_{i=1}^n P(x_i)}}$$

where μ is the Discrete Random Variable Mean; and the series of probabilities P are taken as weights. Based on the eq.(4.2), the DRV S.D. becomes:

$$\sigma = \sqrt{\sum_{i=1}^n (x_i - \mu)^2 P(x_i)}$$

The performances of our “Grouping 07” and “Grouping 13” are assessed on five different time series: Earthquake time series, Foreign Exchange Rates (GBP to USD) time series, Nile River Low Flows time series, Sunspot Number (Monthly Average) time series and Synthetic Pseudo-periodical time series (see Table 4.4 for details). Due to some processing requirements, the outcome values of Groups index 1 ~ 9, A, B, C, D are named 1 ~ 9, 10, 11, 12, 13 corresponding in the results depicted below.

Table 4.9 shows the results of experiments to compare both the “Grouping 13” and “Grouping 07”, where the notations follow the description provided in Table 4.5:

Table 4.9: Experiments Results Comparison Between Combination Rules Grouping 13 and Grouping 07

G.13	Q.1	172	65	178	18	156	49	31	69	176	19	196	53	167
	P.1	0.13	0.05	0.13	0.01	0.12	0.04	0.02	0.05	0.13	0.01	0.15	0.04	0.12
	DRV Mean: 7.0334							DRV S.D.: 4.12						
G.07	Q.1	156	172	178	304	196	167	176						
	P.1	0.12	0.13	0.13	0.23	0.15	0.12	0.13						
	DRV Mean: 4.0504							DRV S.D.: 1.87						
G.13	Q.2	301	9	570	1	278	11	0	8	301	5	528	10	271
	P.2	0.13	0.0	0.25	0.0	0.12	0.0	0.0	0.0	0.13	0.0	0.23	0.0	0.12
	DRV Mean: 6.8744							DRV S.D.: 4.23						
G.07	Q.2	278	301	570	44	528	271	301						
	P.2	0.12	0.13	0.25	0.02	0.23	0.12	0.13						
	DRV Mean: 3.9856							DRV S.D.: 1.94						
G.13	Q.3	70	19	103	6	71	18	3	15	68	9	102	14	70
	P.3	0.12	0.03	0.18	0.01	0.13	0.03	0.01	0.03	0.12	0.02	0.18	0.02	0.12
	DRV Mean: 6.9489							DRV S.D.: 4.17						
G.07	Q.3	71	70	103	84	102	70	68						
	P.3	0.13	0.12	0.18	0.15	0.18	0.12	0.12						
	DRV Mean: 3.9824							DRV S.D.: 1.88						
G.13	Q.4	174	7	233	4	166	6	0	4	182	1	257	3	161
	P.4	0.15	0.01	0.19	0.0	0.14	0.01	0.0	0.0	0.15	0.0	0.21	0.0	0.13
	DRV Mean: 7.0159							DRV S.D.: 4.25						
G.07	Q.4	166	174	233	25	257	161	182						
	P.4	0.14	0.15	0.19	0.02	0.21	0.13	0.15						
	DRV Mean: 4.0384							DRV S.D.: 2.04						
G.13	Q.5	63	0	50064	0	70	0	0	0	63	0	49669	0	70
	P.5	0.0	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.0
	DRV Mean: 6.9845							DRV S.D.: 4.00						
G.07	Q.5	70	63	50064	0	49669	70	63						
	P.5	0.0	0.0	0.5	0.0	0.5	0.0	0.0						
	DRV Mean: 3.9960							DRV S.D.: 1.00						

In an ideal situation of (*Uniform*) Discrete Probability Distribution, all values of discrete variables are equal:

$$P_{G13} = \{P : P(X_t) = 1/13\} \quad t = 1, 2, \dots, 13$$

$$P_{G07} = \{P : P(X_t) = 1/7\} \quad t = 1, 2, \dots, 7$$

Since the outcome values of Grouping 13 and Grouping 07 are named respectively: $X_{G13} = \{1, 2, \dots, 13\}$ and $X_{G07} = \{1, 2, \dots, 7\}$, the DRV Mean of the two combination rules are:

$$\mu_{G13} = \sum_{i=1}^{13} X_{G13}(i)P_{G13}(X_{G13}(i)) = 7$$

$$\mu_{G07} = \sum_{i=1}^7 X_{G07}(i)P_{G07}(X_{G07}(i)) = 4$$

All five time series used in our case studies also show corresponding results close to the results described above in Table 4.9.

As seen in the Table 4.9, the DRV Mean DRV S.D. values for both Grouping 13 and Grouping 07 rules are approximately equal to the ideal value; this also proves that both proposed combination rules (Grouping 13 and Grouping 07) correspond with natural distribution of generic time series values.

The normal (un-weighted) *Standard Deviation* values for Grouping 13 and Grouping 07 are:

$$\sigma_{G13} = \sqrt{\frac{1}{13} \sum_{i=1}^{13} (X_{G13}(i) - \mu_{G13})^2} = 3.74165738677 \quad (4.7)$$

$$\sigma_{G07} = \sqrt{\frac{1}{7} \sum_{i=1}^7 (X_{G07}(i) - \mu_{G07})^2} = 2.0 \quad (4.8)$$

Consequently, based on comparisons between the combination rules and also between DRV Standard Deviation and normal Standard Deviation values, classifier Grouping 07 is able to generate a more stable distribution combination time series. In contrast to it, the dispersion of Grouping 13 is even bigger than the ideal value of Uniform Discrete Distribution (compare values between Table 4.9 and eq.4.8). At the same time, the absence of values corresponding to some groups in the approach Grouping 13 may affect the probability distribution and this justifies even more our optimization approach Grouping 07.

In conclusion, the combination rule Grouping 07 produces a better distribution of samples between groups than Grouping 13.

4.3 The Approach of Series Features Extraction Algorithm for Time Series Analysis and Prediction

4.3.1 Eigenvector

Eigenvectors and *Eigenvalues* are important interrelated concepts in linear algebra. The determination of the eigenvector and eigenvalue of a system is extremely important in engineering, (quantum) mechanics and many other domains. It arises in common applications as stability analysis, such as the oscillations of vibrating systems [Korn and Korn, 2000] [Strang, 2003].

Many kinds of mathematical objects can be treated as vectors, such as: functions, ordered pairs, etc. If a transformation on a (non-zero) vector only changes its magnitudes but not its direction, then this vector is called an eigenvector of that transformation. In these cases, the concept of vector's direction loses the ordinary significance as an abstract meanings, the transformation only effects scalars of vector, for example, stretching, compression, rotation or any combination of these.

A vector is stretched un-equally in different directions along the coordinate axes, then there two eigenvalues as the scaling factors in different directions (*General Eigenvector*). After repeatedly applying this action of stretching/shrinking, almost any vector in vector space could be oriented close enough. However, in this case, if there is still a large distance between them,

then those vectors are not belong to one cluster, in other words, those vector contain different significance.

As discussed, in the newly transformed time series generated by our combination rule, every data point is based on *a priori* information (values and their sequence) from initial time series. Therefore, the series of successive transformed data points helps identifying repetitive schemes showing qualitative movements over time intervals or the trend of time series. Meanwhile, any single or several data points in the combination series can be treated as a vector (eigenvector of the combination transformation), because it contains two factors, time intervals and trend of the original time series (eq.(4.5), eq.(4.6)).

Since the combination rule does not take into account the corresponding magnitudes of any three successive data points because it only evaluates the difference of successive values, all eigenvectors extracted from combination time series can be treated as a stretched primitive eigenvector.

Definition 4.5 - Eigenvector:

Linear transformations of a vector space, such as rotation, reflection, stretching, compression, shear or any combination of these, may be visualized by the effect they produce on vectors. That means the vector has its property (direction) staying same by the transformation, but scaled by a factor.

Given a linear transformation A , a non-zero vector ξ is defined to be an

eigenvector of the transformation if it satisfies the eigenvalue equation:

$$A\xi = \lambda\xi \quad \text{or} \quad A\xi = \lambda I\xi$$

for some scalar λ . In this situation, the scalar λ is called an eigenvalue of A corresponding to the eigenvector ξ . [Korn and Korn, 2000] [Strang, 2003]

For instance, given A transformation, 2×2 vector ξ and scalar λ :

$$A\xi = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \lambda \times x \\ \lambda \times y \end{bmatrix} = \lambda \begin{bmatrix} x \\ y \end{bmatrix} = \lambda\xi$$

Definition 4.6 - General Eigenvector:

Given a linear transformation A and B , a non-zero vector v is defined to be an (general) eigenvector of linear transformation if it satisfies the eigenvalue equation:

$$Av = \lambda Bv$$

for some scalar λ . In this situation, the scalar λ :

$$\begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} \tag{4.9}$$

is called an (general) eigenvalue of A and B corresponding to the eigenvector v [Korn and Korn, 2000] [Strang, 2003]. In relation to concept of normal eigenvector, this is to explain there are two vectors with a set of un-equal scaling.

On the other hand, if any existing eigenvector X of the combination time series, and I_X is the primitive impartible eigenvector of X , λ is an eigenvalue:

$$X = \lambda I_X$$

In view of that component on horizontal dimension is the irreversible time dimension, any eigenvector can be regarded as a stretched component on vertical dimension (the perpendicular directions along the coordinate axes).

Therefore, two series of successive data points in the combination time series:

$$A = \{A_1(t_{p+1}, V_{p+1}), A_2(t_{p+2}, V_{p+2}), \dots, A_n(t_{p+n}, V_{p+n})\} \in \mathbf{R}^n$$

$$B = \{B_1(t_{p+1}, V_{p+1}), B_2(t_{p+2}, V_{p+2}), \dots, B_n(t_{q+n}, V_{q+n})\} \in \mathbf{R}^n$$

where $A_i \in \mathbf{R}^2$, $B_i \in \mathbf{R}^2$, $i \in [1, n]$, both consist of two eigenvector respectively, α and β denoted as: (ignoring component scalar magnitudes because there is an unique time interval)

$$\alpha = \langle V_p, V_{p+1}, \dots, V_{p+n} \rangle \in \mathbf{R}^n$$

$$\beta = \langle V_q, V_{q+1}, \dots, V_{q+n} \rangle \in \mathbf{R}^n$$

4.3.2 Transformation of Time Series

We address the time series data classification task by grouping the trends as described above in order to determine class intervals for input data in order to provide analysis and forecast tools. Grouping *07* is used to label three successive data from the original time series based on their values and it produces a new time series with newly processed value set. Consequently, this kind of time series transformation helps to project the existing data set into a new format, which helps analyzing and detecting patterns in time series and locating local peak/valley data points, and also consequently determining (pseudo-)periods where applicable.

We demonstrate the applicability and performances of our proposed approach on case studies related to the following publicly available data set: Flu Trends in United States (data source: [GoogleTrends, 2009], further reading on [USCDC, 2009]).

Fig 4.8 shows a sample of pseudo-periodical time series with 299 measurements representing the percentage $((\text{influenza}/\text{entire population}) \times 100\%)$ of Flu Trends in United States over five years (2003 ~ 2009) from U.S. Center for Disease Control and Prevention. Table 4.10 illustrates the organization of original and transformed Flu Trends In United States time series in detail (segments for demonstration) by the combination rule “Grouping *07*”.

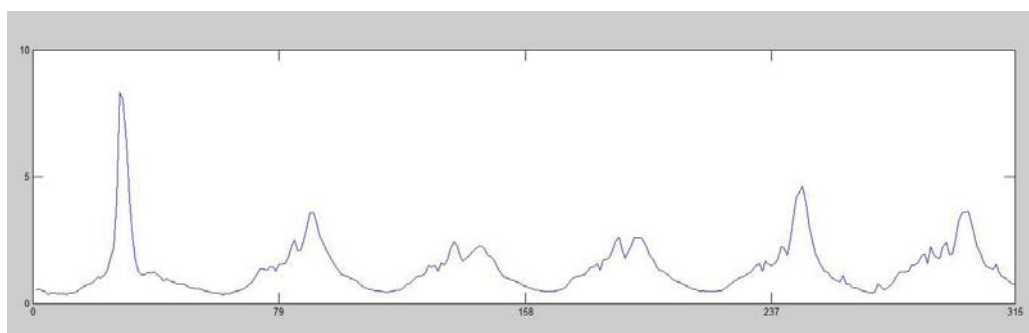


Figure 4.8: A Sample Time Series of Flu Trends in United States; X Coordinate Axis List the Index of Time Intervals and Y Coordinate Axis Illustrates the Flu Trends Time Series Data Set Values.

Table 4.10: Original and Transformed for Flu Trends Time Series

Index	1	2	3	4	5	6	7	8
Original	0.509	0.546	0.501	0.457	0.357	0.408	0.397	0.372
Combination		1	5	5	6	2	5	5
...
Index	25	26	27	28	29	30	31	32
Original	1.799	2.187	4.039	8.3	8.056	6.352	4.116	2.602
Combination	3	3	3	2	5	5	5	5
...
Index	86	87	88	89	90	91	92	93
Original	2.111	2.490	2.971	3.574	3.572	3.176	2.654	2.439
Combination	3	3	3	2	5	5	5	5
...
Index	298	299	310	311	312	313	314	315
Original	1.360	1.525	1.194	1.035	0.991	0.873	0.780	0.739
Combination	7	1	5	5	5	5	5	

4.3.3 Feature Extraction and Pattern Recognition from Historical Values

As discussed previously, one single data point from the generated combination data sequence represents a basic feature of the initial time series, of which the features have been categorized by the natural classifier (data measurements' comparison). Therefore, the generated combination data sequence is an *a priori* knowledge sequence.

Due to the fact that the combination sequence contains information on data changing, a succession of data points provides knowledge of the initial data series. For example, in Table 4.10 and combination rule showed in Table 4.6, points “3 3 3 2 5 5 5 5” (with the index from 25 to 32) indicate there is a “peak” in the initial time series. This acquired knowledge is computed only by measurements' comparison, and not obtained by experience, so a piece of combination sequence is also an *a priori* knowledge sequence.

Because a series of cyclic periods are emerging over time intervals in a pseudo-periodical time series, and the time series prediction progress is a regression progress, the historical data give an opportunity to finding a pattern(s) with contain a common significance. For example, the points with the index from 86 to 93 also give a same string of characters “3 3 3 2 5 5 5 5”; this is to say in the history there are other “peaks” identified, at the same time, one cycle period of this pseudo-periodical time series has been identified (i.e. $P_t = 89 - 28 = 61$). Extending the same principle, two characters strings (patterns) for expressing the “peak” and “valley” shapes based on the

Grouping 07 combination rule are showed below by the *Regular Expression*:

Peak: $[3]^{\{+\}} [1 | 2 | 4]^{\{1\}} [5]^{\{+\}}$

Valley: $[5]^{\{+\}} [4 | 6 | 7]^{\{1\}} [3]^{\{+\}}$

The peak/valley's reference helps to understand the nature of time series, however, for a pseudo-periodical time series it will return a set of unequal values for cyclic pseudo-periods. It is difficult to predict the next value based on that. In particular, by controlling the development of time series in many industry and science domains, the cyclic period could be shifted earlier or postponed later.

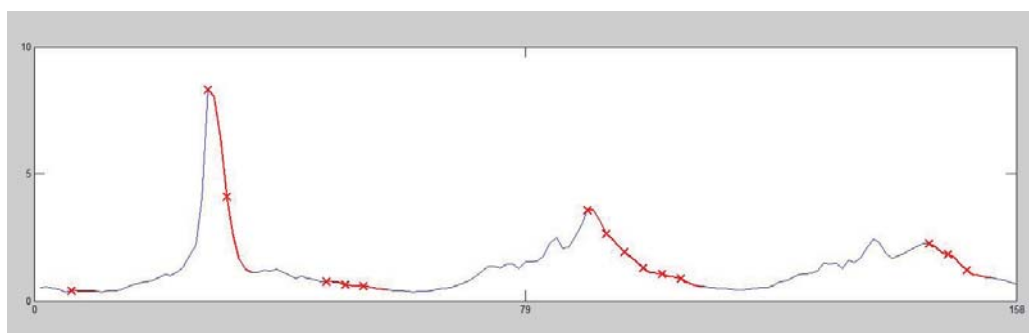
Consequently, the proposed approach is to search for a pre-specified pattern, which normally is the latest string in combination data sequence (due to the motive of prediction), from the entire combination data series for all matched records. That will bring all possible patterns back into comparison, plus the pre-specified pattern is acquired as an *a priori* knowledge and the searching progress is also dependent on experience. Thus, from the point of view of machine learning, this type of un-supervised learning approach is a kind of "white box" method, able to produce *a priori* knowledge.

For example, let's take the first half of Flu Trends in United States time series as the available time series source, and the second half as the target for comparison with the prediction. We obtain the following combinations as in Table 4.11.

Table 4.11: The First Half of Initial and Combination Data Series of Flu Trends in U.S.

Index	1	2	3	4	...	155	156	157	158
Initial	0.509	0.546	0.501	0.457	...	0.857	0.828	0.724	0.665
Combination		1	5	5	...	5	5	5	

To select the latest three (a sample for representation and testing) characters string from the combination data series, which is “5 5 5”, then search the entire combination data sequence, it will return a sets of records as in Fig 4.9.

**Figure 4.9:** Patterns Matched (Each matched string starts with “×”); X Coordinate Axis Lists the Index of Time Intervals and Y Coordinate Axis the Flu Trends Time Series Data Set Values.

This patterns recognition process classifies the pre-specific patterns from the un-labeled raw data sets. Additionally, this description scheme (classification) relied on neither the availability of the given pattern or the initial time series data sets.

4.3.4 Filtering the Returned Matches

The combination rule Grouping \mathcal{O} focused on the comparison between the successive neighbouring data points, and ignored the concrete quantities in time series. In the views of general eigenvector's concepts, two sets of successive data points can be considered (as unequally scaled on each other); for example, two combination data in the group "4" can represent two different shapes during the data changing. Therefore, a filtering mechanism is required to be installed for eliminating all uncertain patterns from the matches returned. In order to precisely measure and filter the unequal patterns, a normalizing pre-process, "Z Score (so-called: Standard Score)", is proposed as the unification step.

Definition 4.7 - Z Score:

A Z Score associated with a variable X is defined as below, where μ is the mean and the σ is the standard deviation of X [Larsen and Marx, 2005].

$$Z = \frac{X - \mu}{\sigma} \quad (4.10)$$

Each recognized pattern contains same structure of data changing. We assume that there is a primitive eigenvector (pattern) I_p to present the primary structure, and let M_0 expresses the latest pattern (pre-specific string) and M_1, M_2, \dots, M_k , ($k \geq 0$) express the returned patterns*:

$$M_0 \xi = \lambda_0 I_p \quad (4.11)$$

*All return patterns, M_1, M_2, \dots, M_k , are normalized by Z Score.

and relationships' equations by the concept of general eigenvector are:

$$M_1\xi = \lambda_1 M_0 = \lambda_1 \lambda_0 I_p$$

$$M_2\xi = \lambda_2 M_0 = \lambda_2 \lambda_0 I_p$$

$$\vdots \quad \quad \quad \vdots \quad \quad \quad \vdots$$

$$M_k\xi = \lambda_k M_0 = \lambda_k \lambda_0 I_p$$

Therefore, the comparison among the scaling eigenvalues, $\lambda_1, \lambda_2, \dots, \lambda_k$ are able to reflect the status of matched patterns from historical time series.

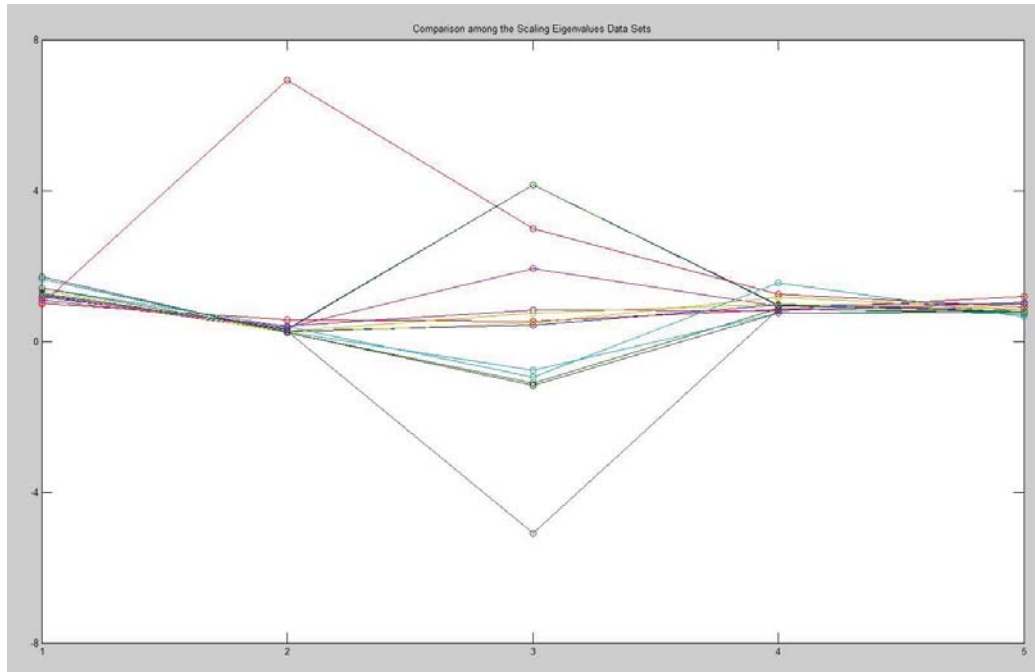


Figure 4.10: Unequal Scaling (Eigenvalues: $\lambda_1, \lambda_2, \dots, \lambda_k$) Sets Comparison for Matched Patterns from Historical Time Series (each “o” on each color line represents one of a series values of eigenvalues (λ)); X coordinate axis presents the index of eigenvalue, and Y coordinate axis indicate their values).

As Fig 4.10 illustrates, most of the eigenvalues show a similar figure with approximate scaler, but there is one (several in others cases) clearly different from anyone else. In other words, each corresponding parallel values in one eigenvalue is approximately equivalent to the others.

$$\frac{\lambda_i(x)}{\lambda_p(x)} \simeq \frac{\lambda_j(x)}{\lambda_q(x)}$$

where $i, j, p, q = 1, 2, \dots, k$, x is the data index of λ .

Meanwhile, the comparison of (algebraic) *Determinants* for the eigenvalues are able to identify and classify those (please see Definition 4.8 and Figure 4.11).

Definition 4.8 - Determinants:

The fundamental geometric meaning of a determinant is a scale factor for measure when the matrix is regarded as a linear transformation. The determinant of a matrix of arbitrary size can be defined by the Leibniz formula:

$$\det(A) = \sum_{\sigma \in S_n} \text{sgn}(\sigma) \prod_{i=1}^n A_{i,\sigma(i)}$$

where the set of all permutations is denoted S_n , σ is any permutation in the S_n , and A is represented as the target data set (matrix).

Hence, after searching the pre-specific pattern from the historical data in combination data sequence, then comparing the eigenvalues, the patterns are in accordance with the pre-specific one but have a different structure and the composition can be filtered from the returned matches.

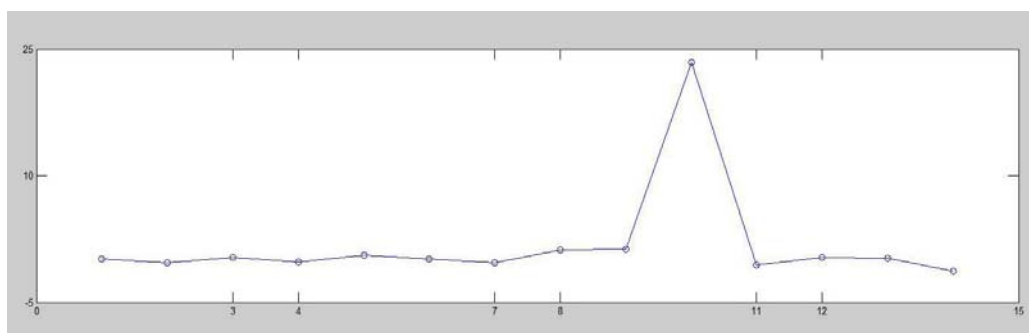


Figure 4.11: The Comparison of Eigenvalues' Determinants; X Coordinate Axis Lists the Index of Eigenvalues Series and Y Coordinate Axis Illustrates the Determinant Values for each Eigenvalues Series.

In details, at the first, the algorithm SFE takes the initial time series data set, where indexed by time intervals t and $t \in \mathbf{N}$. Secondly, SFE converts/transforms the initial time series into combination data sequence by Grouping *07* combination rule. Then, SFE records the latest three characters from the transformed data sequence as a target pattern. Next, searching same patterns from historical data set, which have same structure with the target pattern (if no any pattern matched after searching, SFE consider the future value stay same with latest value because history did not own same data values what have same trend (moving behaviour) or the historical data set do not have enough data samples, otherwise, the historical data set probably contains error(s) data). And then, SFE compares and filters the matched patterns with the target pattern by the general eigenvalues and the determinant values. After filtering (classifying) out the patterns contain different significance, SFE compute the regression coefficients by using the original data sequences values (backtrack the index of time intervals by filtered patterns location in the initial time series) and the next one data point. Finally, SFE exports the prediction value based on the target pattern and the regression coefficients.

Table 4.12: Pseudo-codes for Time Series Prediction Algorithm based on Series Features Extraction

```

INPUT: An Initial General Time Series Data Set;
METHOD: Series Features Extraction;
OUTPUT: Predicted Time Series Data;

```

```

01. // Import the initial time series data set;
02. SET A[ ] to READ(initial time series);
03. // Convert time series into combination data sequence;
04. GET G[ ] from Grouping07(A[ ]);
05. // Get latest several characters from combination ...
06. // sequence as a sample pattern;
07. GET P from G[end-L to end];
08. // Searching same patterns with structure ...
09. // from histrocal data set;
10. SET counter to 0;
11. WHILE counter < LENGTH(G[ ])
12.     IF G[counter to counter+L] match P
13.         ADD G[counter to counter+L] to MG[ ];
14.         ADD A[counter to counter+L+2] to MA[ ];
15.     ENDIF
16. ENDWHILE
17. // Calculate the general eigenvalues of MA[ ];
18. GET Xi[ ] from Eigenvalue(MA[ ]);
19. FOR each of Xi[ ]
20.     FOR each of Xi[ ]
21.         GET R[ ] from Xi[start to end-1]/Xi[start+1 to end];
22.     ENDFOR
23. ENDFOR
24. // Set a threshold based on normal distribution;
25. GET NorDis to a normal discription from Determinant(R[ ]);
26. SET Threshold from NorDis;
27. // Filter the eigenvalues;
28. FOR each of R[ ]
29.     IF R[runner] NOT IN NorDis
30.         DELETE R[runner];
31.     ENDIF
32. ENDFOR
33. // Compute the prediction by regression analysis;
34. GET P RasAna(R[ ])
35. // Export the prediction;
36. OUTPUT(P);

```

4.3.5 Computing the Prediction

So far, the learning and filtering processes from the features extraction (Grouping *07* combination rule), pattern recognition (a successive data points in the combination sequence) to filtering the matches returned, handle the *a priori* information and knowledge of the initial time series data set. Therefore, it distinguishes a set of filtered cyclic periods from *a priori* knowledge without using *a posteriori* knowledge. According to these information on periods, prediction can be computed by regression analysis. Table 4.12 lists the pseudo-code for the time series prediction algorithm based on Series Features Extraction.

4.4 Case Studies

This section presents the application of *Series Features Extraction* (SFE) time series prediction algorithm for flu trends time series, foreign exchange rates (GBP to USD) time series and U.S. interests rates time series (see chapter 2 for the data' description).

• Flu Trends Time Series Prediction

An influenza rates (weekly reported 315 measurements) of flu trends in United States time series has been imported into the SFE prediction algorithm. Fig 4.12 depicts the initial time series measurements and prediction of values based on our method. SFE produced very good prediction results

on the trends of time series, however, there are still errors on the *peak*. Fig 4.13 illustrates the prediction errors by algorithm SFE.

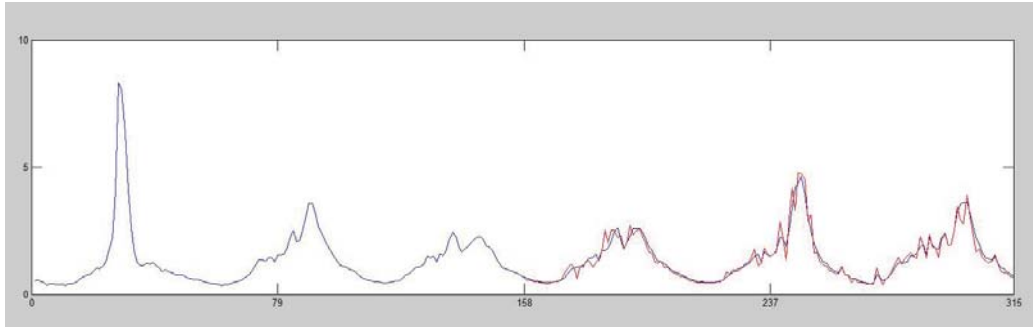


Figure 4.12: The Initial Flu Trends Time Series Values and Prediction Results by SFE Algorithm; X Coordinate Axis Lists the Index of Time Intervals (315 Values) and Y Coordinate Axis Illustrates the Original (in blue with 315 values) and Prediction Values (in red with 157 values).

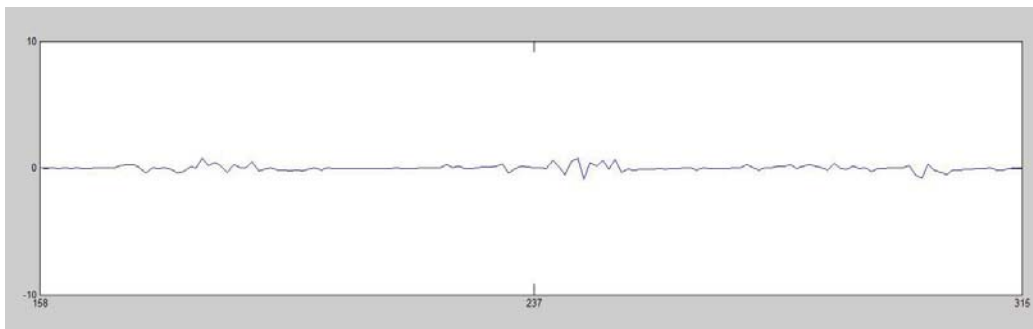


Figure 4.13: The Flu Trends Time Series Prediction Errors by SFE Algorithm; X Coordinate Axis Lists the Index of Time Intervals (315 values) and Y Coordinate Axis Illustrates the Prediction Error ($||\text{Prediction} - \text{Original}||$) Values with 157 Values.

- **Foreign Exchange Rates (GBP to USD) Time Series Prediction**

A time series of foreign exchange rates with 2295 days' observations has been tested by SFE algorithm. The Fig 4.14 illustrates the original observations and SFE's prediction results. SFE produce excellent results both on the

trends and on the values, meanwhile, SFE produced good peaks and valleys prediction values. Fig 4.13 shows the prediction errors by algorithm SFE.

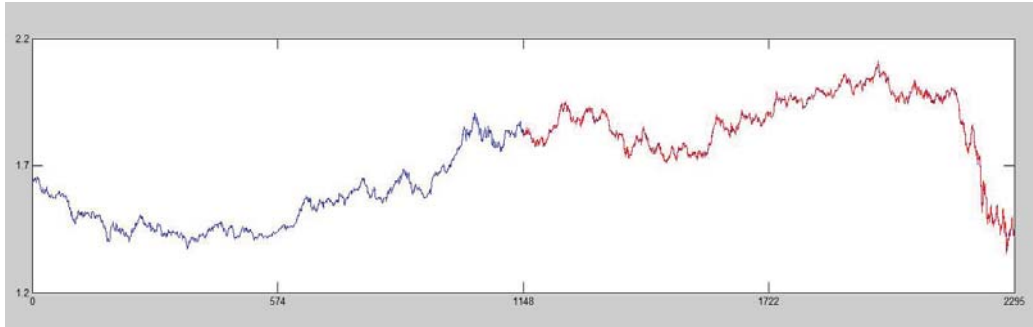


Figure 4.14: The Initial Foreign Exchange Rates (GBP to USD) Time Series Values and Prediction Results by SFE Algorithm; X Coordinate Axis Lists the Index of Time Intervals (2295 Values) and Y Coordinate Axis Illustrates the Original (in blue with 2295 values) and Prediction Values (in red with 1148 values).

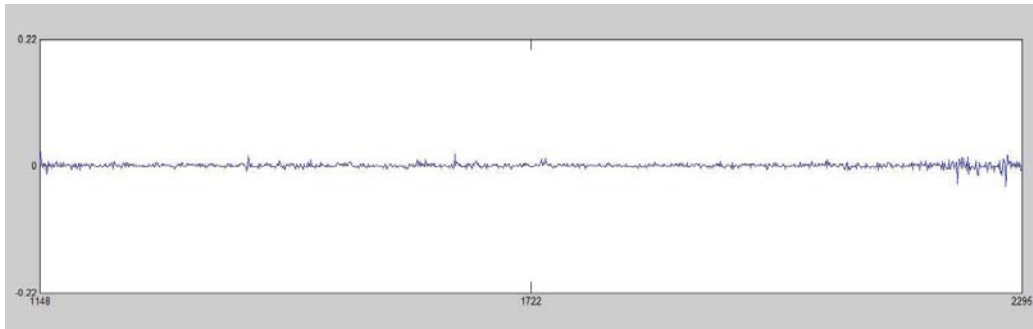


Figure 4.15: The Foreign Exchange Rates (GBP to USD) Time Series Prediction Errors by SFE Algorithm; X Coordinate Axis Lists the Index of Time Intervals (2295 values) and Y Coordinate Axis Illustrates the Prediction Error ($||\text{Prediction} - \text{Original}||$) Values with 1148 Values.

- **United States Interests Rates**

A time series with 582 measurements for United States interesting rates has been input to the algorithm. Fig 4.16 shows the testing time series values

and its prediction results. SFE produced prediction results very well both on the trends and on values, there are fluctuation in prediction values, but the errors were very small. Fig 4.13 shows the prediction errors by algorithm SFE.

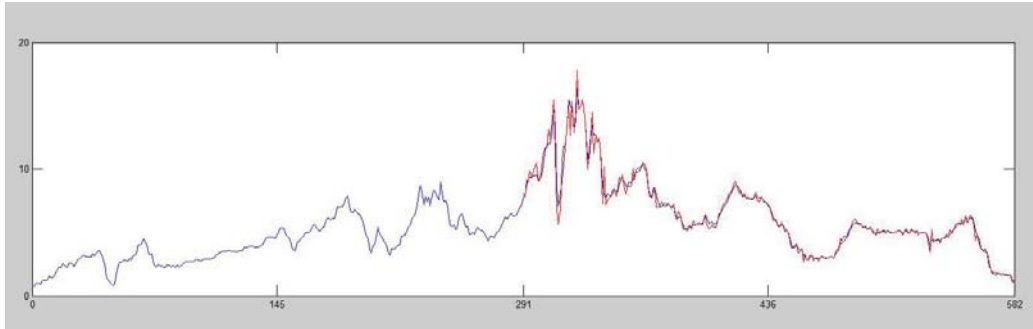


Figure 4.16: The Initial U.S. Interests Rates Time Series Values and Prediction Results by SFE Algorithm; X Coordinate Axis Lists the Index of Time Intervals (582 Values) and Y Coordinate Axis Illustrates the Original (in blue with 582 values) and Prediction Values (in red with 291 values).

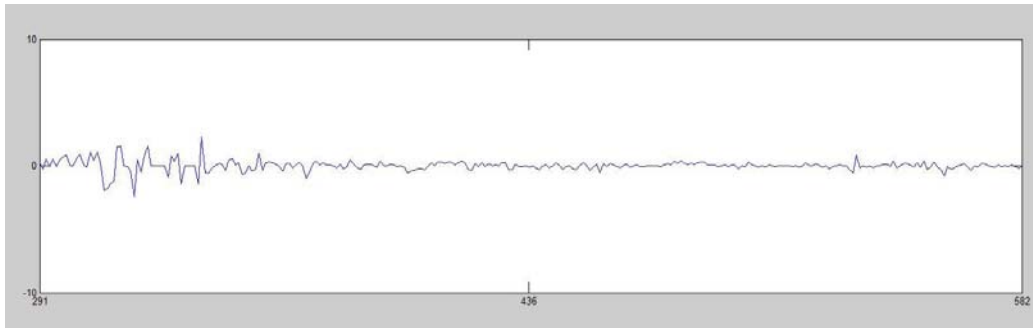


Figure 4.17: The U.S. Interests Rates Time Series Prediction Errors by SFE Algorithm; X Coordinate Axis Lists the Index of Time Intervals (582 values) and Y Coordinate Axis Illustrates the Prediction Error ($||\text{Prediction} - \text{Original}||$) Values with 291 Values.

4.5 Summary

This chapter has introduced a time series prediction algorithm based on *Series Features Extraction* and its predicting performance by testing three types of time series.

At the beginning of this chapter, the introduction section has presented the background of an “un-supervised” learning’s requirements and methodologies for the development of our algorithm: *Epistemology, A Priori and A Posteriori Knowledge, Features, Patterns and Models*.

The concepts of combination rules *Grouping 13* and *Grouping 07* have been presented for a transformation of the initial time series. Due to the result of comparison between the two combination rules, *Grouping 07* showed a better performance than *Grouping 13*.

Following the introduction of *General Eigenvector*, a filtering mechanism have been proposed for removing inappropriate patterns from the matches. Therefore, the prediction results can be computed based on that. The case studies section presents the prediction results by our proposed algorithm.

Chapter 5

Evaluation

Contents

5.1	Implementation of Classical Methods and Proposed Prediction Algorithms	118
5.2	Evaluation of Prediction Results	120
5.3	Summary	138

5.1 Implementation of Classical Methods and Proposed Prediction Algorithms

This chapter evaluates the performance and compares the prediction results of proposed algorithms, *Moving Average of n^{th} -order Difference (MANoD)* and *Series Features Extraction (SFE)*, with classical methods, *Linear Regression (LR)* and *Auto-Regression Moving Average (ARMA)*.

As described in previous chapters, MANoD and SFE algorithms have their own prediction parameters and coefficients; for a unified approach of implementation and evaluation, a single length of data points as regression coefficients is used for all prediction algorithms, e.g. the length of regression should be same for predicting the next value:

$$a_{t+1} = F\left(\underbrace{a_t, a_{t-1}, a_{t-2}, \dots, a_{t-l+1}}_{l: \text{length of data for regression}}\right)$$

where the data sequence $a_t, a_{t-1}, a_{t-2}, \dots, a_{t-l+1}$ is also called *Regressor*; F is the time series prediction methods, in this thesis, it represents the classical methods: LR and ARMA (introduced in chapter 2); our proposed algorithms: MANoD and SFE.

In 1982, E. J. Hannan and J Rissanen presented a equation for estimation of auto-regression moving average order [Hannan and Rissanen, 1982]. They considered the problem for estimating of degrees of the ARMA lag operators, and give a equation about how to define the orders (p, q) , of an ARMA

sequence $y(T)$ by minimizing a criterion:

$$\log \sigma^2 + (p + q) \log T/T \quad (5.1)$$

where σ^2 is the maximum likelihood estimate of the variance of the innovations. The equation shows how the sequence of regression may, for $p = q$, be economically recursive, and calculates by embedding then in a sequence of bivariate auto-regressions [Hannan and Rissanen, 1982] [Hannan and Kavalieris, 1984].

However, in some special types of statistical analysis, the definition of regressors may depends on the specific context. We apply the estimation equation on the *synthetic time series* data set, introduced in chapter 2, for estimating a regressor, because it is generated by a mathematical function and has a high pseudo-periodicity. As a result, the regressor, the data length for regression, was found as “10”.

Consequently, to define a fair same measure for evaluating the performance of algorithms, the found regressor, 10, is also used into the classical method LR and proposed algorithms MANoD and SFE. The key equation for prediction defined as below:

$$LR(10) : \hat{X} = \alpha_0 + \sum_{i=1}^{10} \alpha_i X_i + \varepsilon \quad (5.2)$$

$$ARMA(10, 10) : X_t = \sum_{i=1}^{10} \alpha_i X_{t-i} + \sum_{i=1}^{10} \beta_i \varepsilon_{t-i} + \varepsilon_t \quad (5.3)$$

$$MANoD(10) : D_m^{10} = \sum_{i=0}^{10} (-1)^{n-i} C_n^i X_{10+i} \quad (5.4)$$

Because the algorithm SFE analyzes the combination data sequences, and each data points in combination sequences represents three successive data point of the initial time series. As a results, SFE(8) is able to cover 10 data points, with the same length as the other three models (LR(10), ARMA(10, 10) and MANoD(10)).

5.2 Evaluation of Prediction Results

5.2.1 Testing Time Series Case Studies for Evaluation

To demonstrate the efficiency of the proposed algorithms, we use five Testing Time Series (TTS), as introduced in chapter 2, characterized by different data length, attributes, structure and source (see Table 5.1).

All five time series are pseudo-periodical time series, where there is a continuous variable τ and $\min(\tau) = \min(t)$ and $\max(\tau) = \max(t)$ (t is the time interval of testing time series); a continuous function $\mathcal{F}(\tau)$ existing and the testing time series data set $X \subseteq \mathcal{F}(\tau)$. Also, at least one or a series of time

interval ξ existing, then let:

$$\mathcal{F}(\tau_0) = \mathcal{F}(\tau_0 + \xi), \quad \tau_0 \in \tau, \tau_0 + \xi \in \tau, \xi > 0 \quad (5.5)$$

Table 5.1: Testing Time Series Details

Index	Name	Contents	Source
TTS 1	Earthquakes	Richter Magnitude Scale	NGDC
TTS 2	Flu Trends	Influenza Rates	Google Trends
TTS 3	Nile Flooding	Monthly Average Flow	Time Series Library
TTS 4	Sunspot Number	Monthly Average	NDGC
TTS 5	Synthetic	Generate by Function	KDD Archive

5.2.2 Prediction Results Comparison

5.2.2.1 Measures for Results Evaluation

Two error measures are used for testing the performance of the classical and proposed algorithms: Mean Absolute Error (MAE) and Correlation Coefficient percentage (CCp).

Definition 5.1 - Mean Absolute Error (MAE):

In statistics, the Mean Absolute Error (MAE) is a quantity used to measure how close forecasts' and/or predictions' values are to eventual outcomes. The

MAE is defined by:

$$E_{MAE} = \frac{1}{N} \sum_{i=1}^N \|\hat{X}_i - X_i\| \quad (5.6)$$

where X_i is an original and true value; \hat{X}_i is a prediction value.

Definition 5.2 - Correlation Coefficient percentage (CCp):

The percentage of Correlation Coefficient can be interpreted as a correlation index between the respective variables; it is able to indicate the degree of correlation between the original and prediction time series.

$$\begin{aligned} E_{CCp} &= \frac{cov(X, Y)}{\sigma_X \sigma_Y} \times 100\% \\ &= \frac{E((X - \mu_X)(Y - \mu_Y))}{\sigma_X \sigma_Y} \times 100\% \end{aligned} \quad (5.7)$$

where X, Y are the initial and prediction values, $cov()$ expresses the covariance, μ depicts standard deviations and $E()$ is the expected value.

5.2.2.2 Linear Regression: Prediction Results

In Fig 5.1, 5.2, 5.3, 5.4 and 5.5, we show the initial and prediction result's values of Earthquakes (Richter Magnitude Scale), Flu Trends in United States (influenza rates), Nile River Flow (monthly average), Sunspot Number (monthly average) and Synthetic Pseudo-Periodical time series.

From Fig 5.1 below showed, LR produced a good “shape” of prediction com-

pared with the original time series, however, a number of prediction results are smaller than actual time series values.

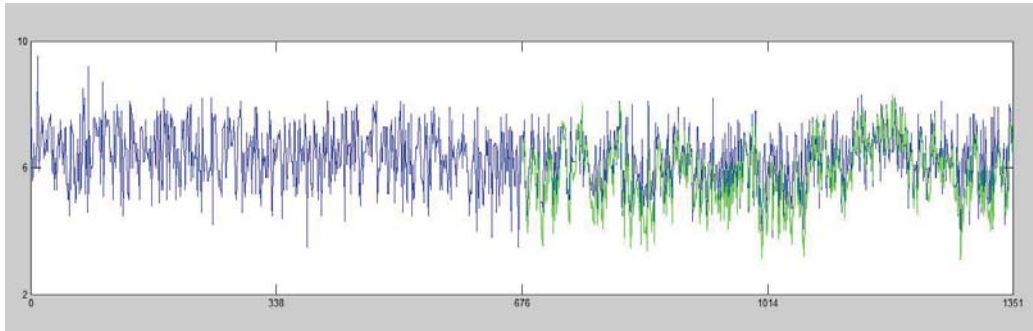


Figure 5.1: Prediction Results for Earthquakes (Richter Magnitude Scale) Time Series by LR Method; X Coordinate Axis Lists the Index of Time Intervals (1351 Values) and Y Coordinate Axis Illustrates the Original (in blue with 1351 values) and Prediction Values (in green with 676 values).

Like the prediction results on earthquakes time series (Fig 5.2), LR also produced good trends on influenza rates time series, however, there are quite lot of values bigger than actual values.

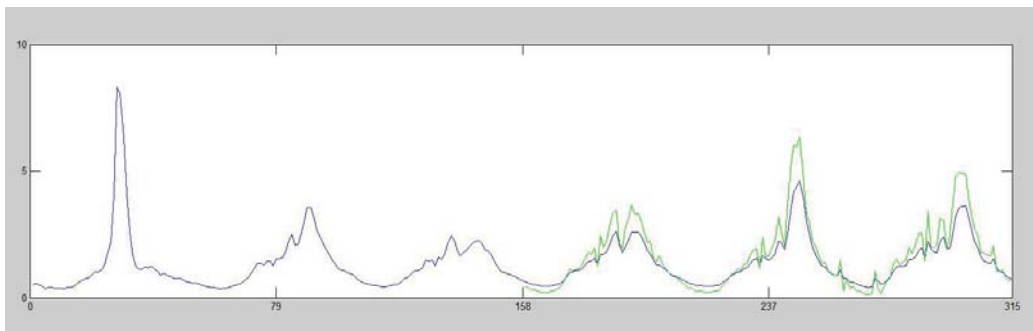


Figure 5.2: Prediction Results for Flu Trends in United States (Influenza Rates) Time Series by LR Method; X Coordinate Axis Lists the Index of Time Intervals (315 Values) and Y Coordinate Axis Illustrates the Original (in blue with 315 values) and Prediction Values (in green with 158 values).

LR method produces very good results on Nile river flow time series (Fig 5.3), both on the trends and on values, however, it gives negative values and they are impossible for a river flow observation.

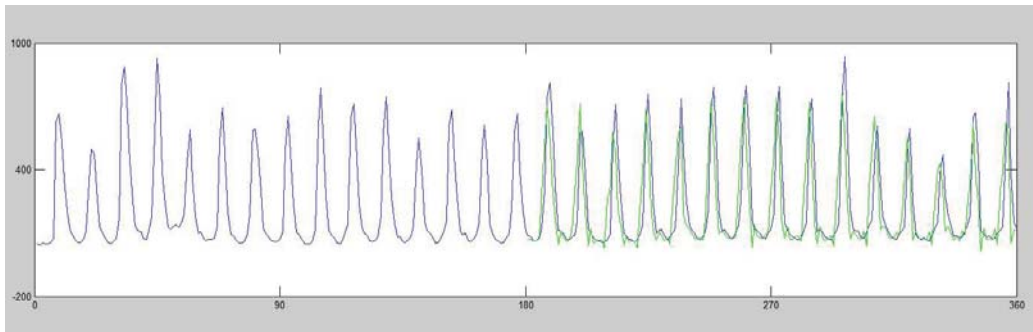


Figure 5.3: Prediction Results for Nile River Flow Time Series by LR Method; X Coordinate Axis Lists the Index of Time Intervals (360 Values) and Y Coordinate Axis Illustrates the Original (in blue with 360 values) and Prediction Values (in green with 180 values).

Exactly similar to the prediction results for earthquake time series, LR methods also provided a number of small values than actual values (Fig 5.4), this caused the prediction errors.

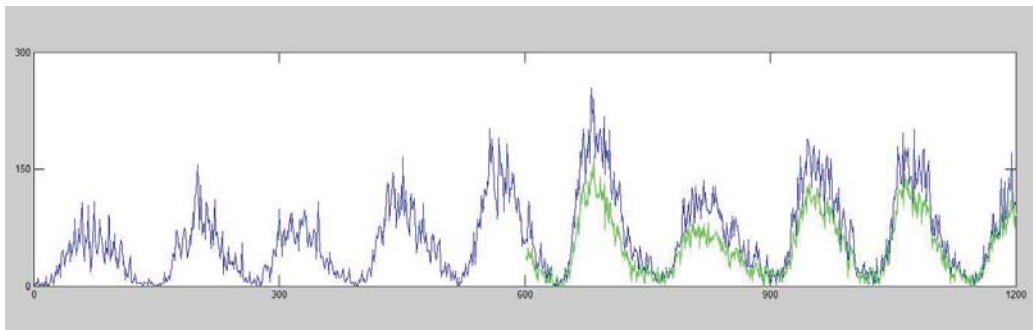


Figure 5.4: Prediction Results for Sunspot Number Time Series by LR Method; X Coordinate Axis Lists the Index of Time Intervals (1200 Values) and Y Coordinate Axis Illustrates the Original (in blue with 1200 values) and Prediction Values (in green with 600 values).

Because the synthetic pseudo-periodical time series is generated by a mathematical function, the data sequence appears highly periodical, but never exactly repeats itself. LR produced very good prediction results.

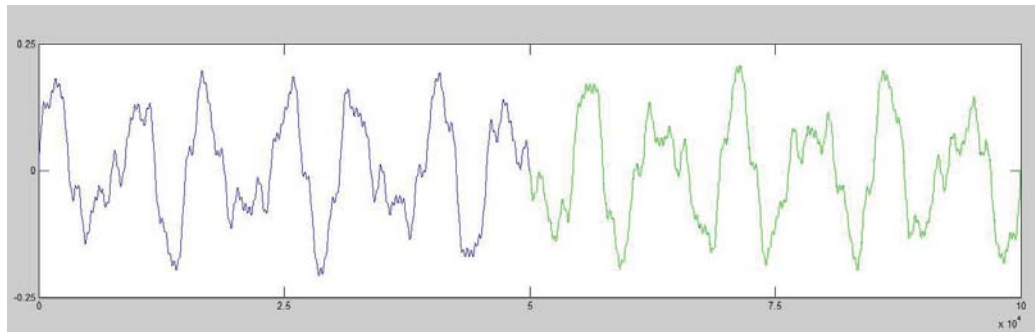


Figure 5.5: Prediction Results for Synthetic Pseudo-Periodical Time Series by LR Method; X Coordinate Axis Lists the Index of Time Intervals (100000 Values) and Y Coordinate Axis Illustrates the Original (in blue with 100000 values) and Prediction Values (in green with 50000 values).

5.2.2.3 Auto-Regression Moving Average: Prediction Results

Fig 5.6, Fig 5.7, Fig 5.8, Fig 5.9 and Fig 5.10 show the initial and prediction result's values of Earthquakes (Richter Magnitude Scale), Flu Trends in United States (influenza rates), Nile River Flow (monthly average), Sunspot Number (monthly average) and Synthetic Pseudo-Periodical time series.

ARMA method produced good results for a short term (see the prediction results at the first), but it did not predict good enough after. (Fig 5.7)

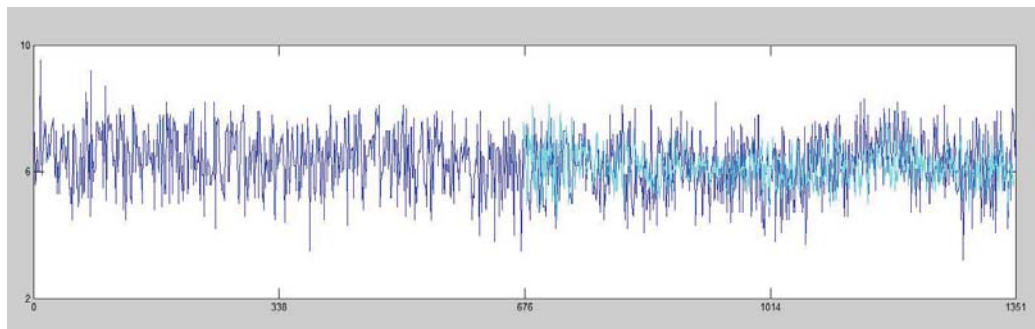


Figure 5.6: Prediction Results for Earthquakes (Richter Magnitude Scale) Time Series by ARMA Method; X Coordinate Axis Lists the Index of Time Intervals (1351 Values) and Y Coordinate Axis Illustrates the Original (in blue with 1351 values) and Prediction Values (in cyan with 676 values).

Although, ARMA methods predicted good results both on the data trends and on the values for the flu trends time series, ARMA predictions have a delay on time dimension (see Fig 5.7).

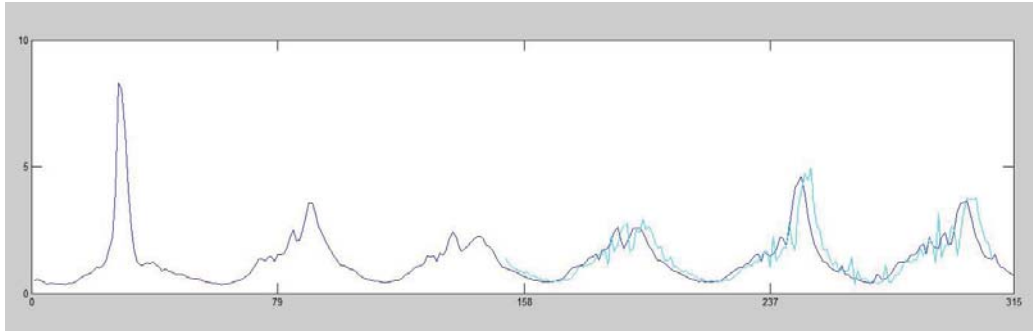


Figure 5.7: Prediction Results for Flu Trends in United States (Influenza Rates) Time Series by ARMA Method; X Coordinate Axis Lists the Index of Time Intervals (315 Values) and Y Coordinate Axis Illustrates the Original (in blue with 315 values) and Prediction Values (in cyan with 158 values).

ARMA methods produced good prediction results for the Nile river time series, however, there are large errors on several (pseudo-)periods (Fig 5.8), and with prediction of amplitude of some *peak* values.

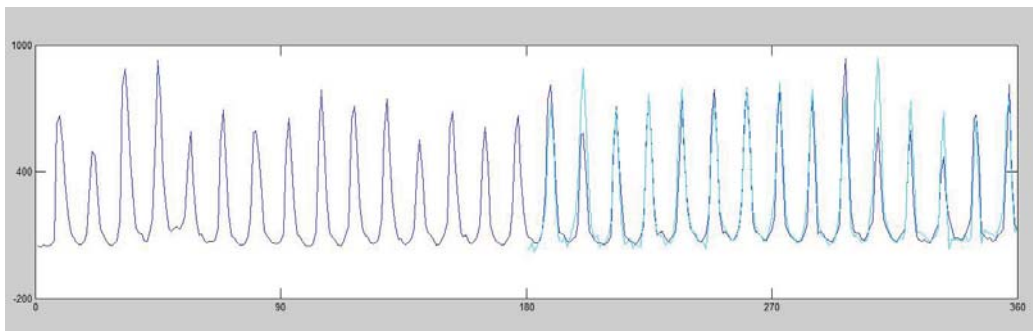


Figure 5.8: Prediction Results for Nile River Flow Time Series by ARMA Method; X Coordinate Axis Lists the Index of Time Intervals (360 Values) and Y Coordinate Axis Illustrates the Original (in blue with 360 values) and Prediction Values (in cyan with 180 values).

There is also a delay on prediction results for sunspot number time series by ARMA methods, and ARMA produced several large errors on (pseudo-)periods (Fig 5.9)

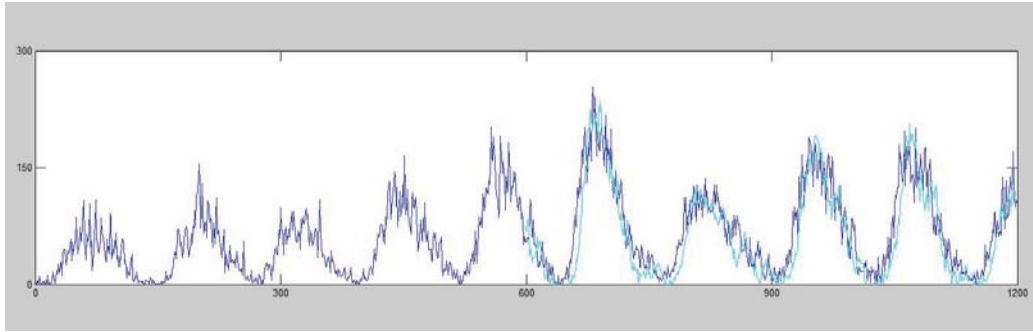


Figure 5.9: Prediction Results for Sunspot Number Time Series by ARMA Method; X Coordinate Axis Lists the Index of Time Intervals (1200 Values) and Y Coordinate Axis Illustrates the Original (in blue with 1200 values) and Prediction Values (in cyan with 600 values).

Similar to the prediction results by LR, ARMA method produced a good results on the synthetic pseudo-periodical time series, although, there is a delay on time dimension. (Figure 5.10)

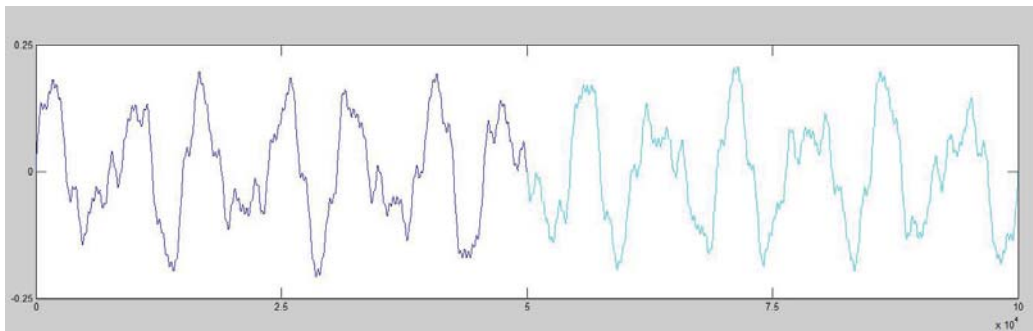


Figure 5.10: Prediction Results for Synthetic Pseudo-Periodical Time Series by ARMA Method; X Coordinate Axis Lists the Index of Time Intervals (100000 Values) and Y Coordinate Axis Illustrates the Original (in blue with 100000 values) and Prediction Values (in cyan with 50000 values).

5.2.2.4 Moving Average of n^{th} -order Difference: Prediction Results

Fig 5.11, Fig 5.12, Fig 5.13, Fig 5.14 and Fig 5.15 show the initial and prediction result's values of Earthquakes (Richter Magnitude Scale), Flu Trends in United States (influenza rates), Nile River Flow (monthly average), Sunspot Number (monthly average) and Synthetic Pseudo-Periodical time series.

For the earthquakes time series, MANoD method produced a very good prediction results (Fig 5.11), prediction errors exist but smaller than for classical algorithms.

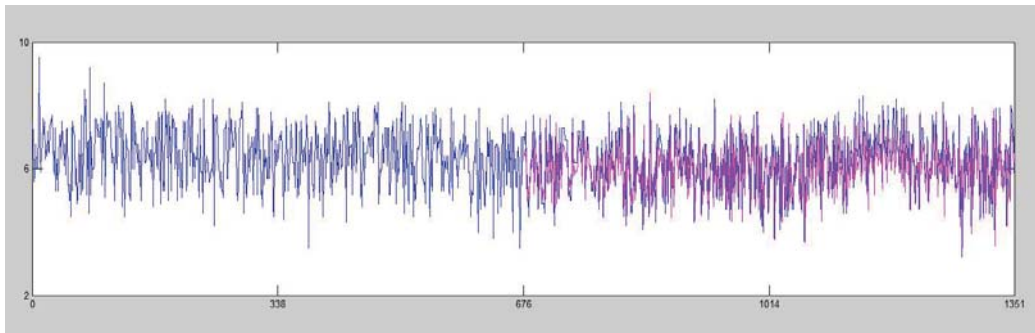


Figure 5.11: Prediction Results for Earthquakes (Richter Magnitude Scale) Time Series by MANoD Method; X Coordinate Axis Lists the Index of Time Intervals (1351 Values) and Y Coordinate Axis Illustrates the Original (in blue with 1351 values) and Prediction Values (in purple with 676 values).

MANoD gave good trends for flu trends time series prediction (Fig 5.12), it, unlike the ARMA, reduced the delay on time dimension.

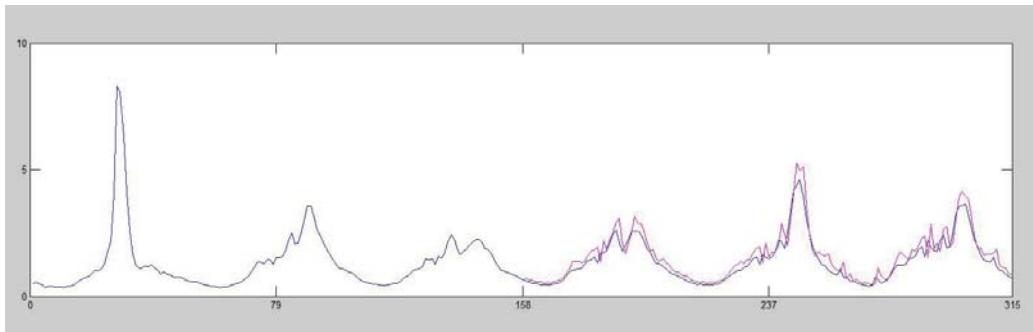


Figure 5.12: Prediction Results for Flu Trends in United States (Influenza Rates) Time Series by MANoD Method; X Coordinate Axis Lists the Index of Time Intervals (315 Values) and Y Coordinate Axis Illustrates the Original (in blue with 315 values) and Prediction Values (in purple with 158 values).

Although MANoD produced good results on every (pseudo-)periods's *peak*, but not on the *valley*, MANoD gave good prediction results on the Nile river flow time series (Fig 5.13).

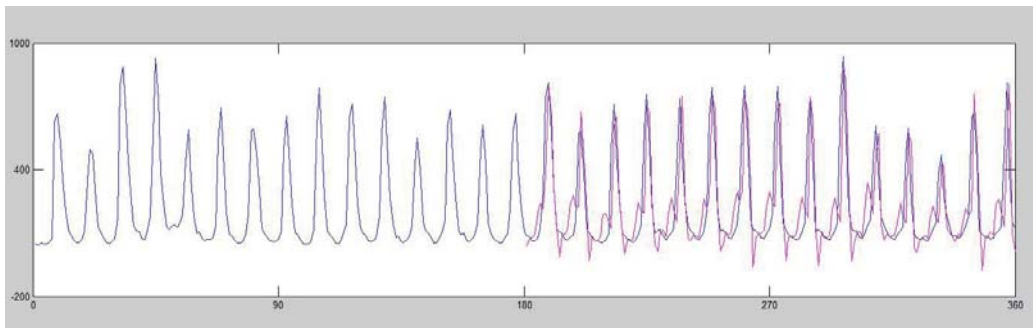


Figure 5.13: Prediction Results for Nile River Flow Time Series by MANoD Method; X Coordinate Axis Lists the Index of Time Intervals (360 Values) and Y Coordinate Axis Illustrates the Original (in blue with 360 values) and Prediction Values (in purple with 180 values).

MANoD produced a number of large errors, however, MANoD produced a very good trends for sunspot number time series (Fig 5.14).

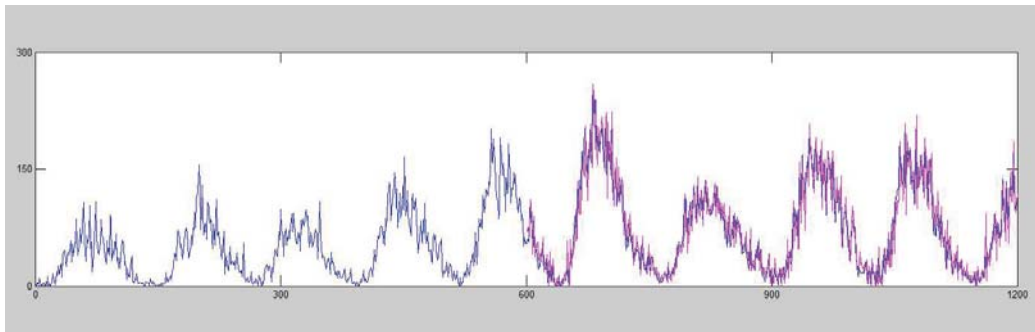


Figure 5.14: Prediction Results for Sunspot Number Time Series by MANoD Method; X Coordinate Axis Lists the Index of Time Intervals (1200 Values) and Y Coordinate Axis Illustrates the Original (in blue with 1200 values) and Prediction Values (in purple with 600 values).

MANoD produced prediction results very well both on the trends and on values for synthetic pseudo-periodical time series (Fig 5.15), errors were very small.

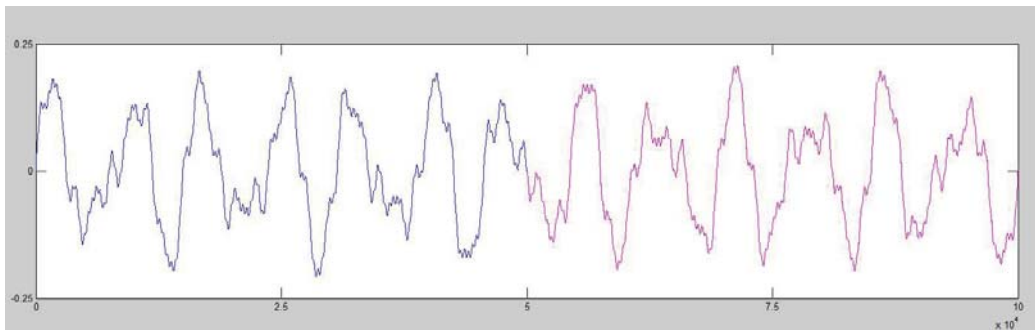


Figure 5.15: Prediction Results for Synthetic Pseudo-Periodical Time Series by MANoD Method; X Coordinate Axis Lists the Index of Time Intervals (100000 Values) and Y Coordinate Axis Illustrates the Original (in blue with 100000 values) and Prediction Values (in purple with 50000 values).

5.2.2.5 Series Features Extraction: Prediction Results

Fig 5.16, Fig 5.17, Fig 5.18, Fig 5.19 and Fig 5.20 show the initial measurements and predicted results of Earthquakes (Richter Magnitude Scale), Flu Trends in U.S. (influenza rates), Nile River Flow (monthly average), Sunspot Number (monthly average) and Synthetic Pseudo-Periodical time series.

SFE methods produced good prediction results on the short term for earthquakes time series (the first part of results in Fig 5.16), there are errors for long term prediction, but SFE still produced a perfect trends.

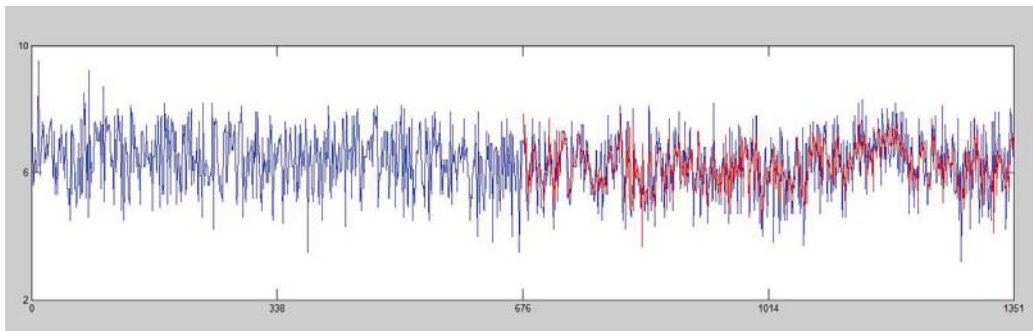


Figure 5.16: Prediction Results for Earthquakes (Richter Magnitude Scale) Time Series by SFE Method; X Coordinate Axis Lists the Index of Time Intervals (1351 Values) and Y Coordinate Axis Illustrates the Original (in blue with 1351 values) and Prediction Values (in red with 676 values).

SFE produced an excellent prediction results for the flu trends time series (Fig 5.17), compared to other three methods, SFE gave the best prediction results.

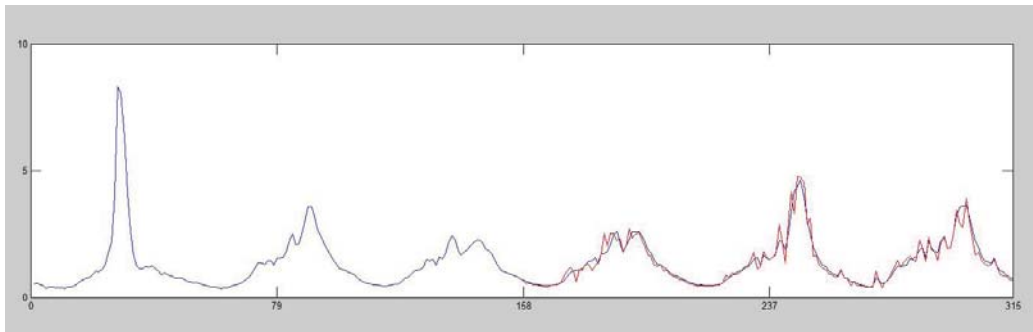


Figure 5.17: Prediction Results for Flu Trends in United States (Influenza Rates) Time Series by SFE Method; X Coordinate Axis Lists the Index of Time Intervals (315 Values) and Y Coordinate Axis Illustrates the Original (in blue with 315 values) and Prediction Values (in red with 158 values).

In contrast with others three methods' prediction results, SFE provided good trends for Nile river flow time series, SFE reduced the errors of prediction on the *valley* (Fig 5.18).

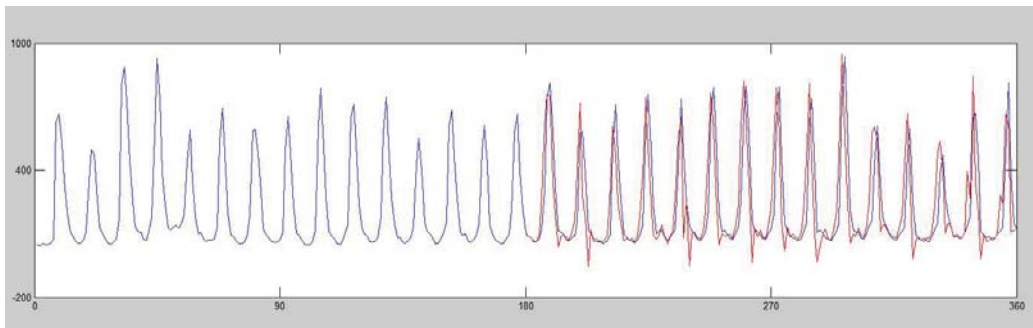


Figure 5.18: Prediction Results for Nile River Flow Time Series by SFE Method; X Coordinate Axis Lists the Index of Time Intervals (360 Values) and Y Coordinate Axis Illustrates the Original (in blue with 360 values) and Prediction Values (in red with 180 values).

For the sunspot number time series, SFE produced an excellent prediction results both on the trends and values, even better than proposed algorithm MANoD (Fig 5.19).

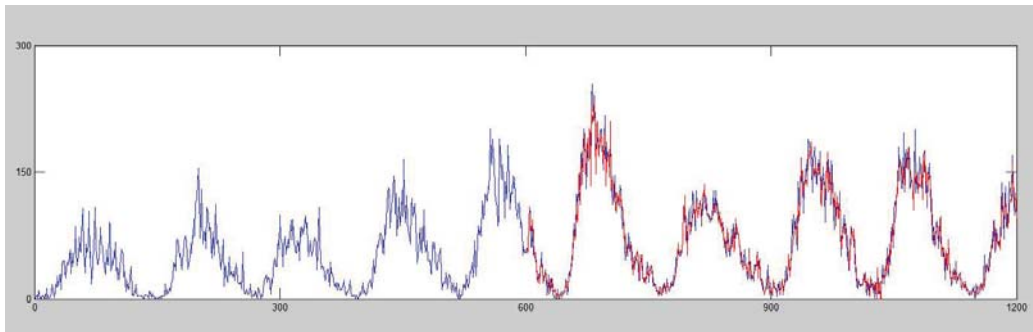


Figure 5.19: Prediction Results for Sunspot Number Time Series by SFE Method; X Coordinate Axis Lists the Index of Time Intervals (1200 Values) and Y Coordinate Axis Illustrates the Original (in blue with 1200 values) and Prediction Values (in red with 600 values).

Like the other three methods, SFE also produced a wonderful prediction results for synthetic time series (Fig 5.20).

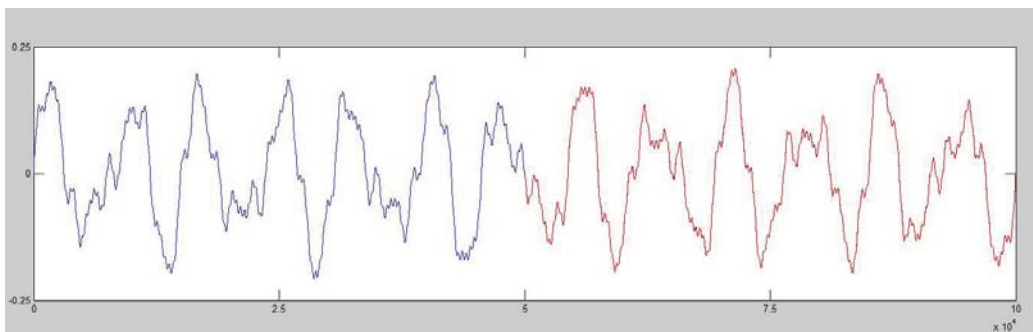


Figure 5.20: Prediction Results for Synthetic Pseudo-Periodical Time Series by SFE Method; X Coordinate Axis Lists the Index of Time Intervals (100000 Values) and Y Coordinate Axis Illustrates the Original (in blue with 100000 values) and Prediction Values (in red with 50000 values).

5.2.2.6 Results Comparison

We conclude with the prediction results of both the classical methods and our original algorithm for five testing pseudo-periodical time series, and some significant samples from prediction results are depicted below:

Table 5.2: Prediction Results Comparison

Prediction Results of Earthquakes Time Series					
Original Values		Predicted Values			
Index	Values	LR	ARMA	MANoD	SFE
677	7.00000	6.17000	7.57580	6.41590	6.84360
678	6.80000	6.49130	6.75620	6.59960	6.61390
...
1014	6.20000	4.70400	6.14770	5.51670	6.03830
1015	5.90000	4.94970	4.88490	6.06410	5.74600
...
1350	7.00000	7.32170	5.54980	5.87620	7.09960
1351	6.00000	6.47540	5.71710	6.53510	6.73860
E_{MAE}	—	0.58112	0.82465	0.88783	0.50493
E_{CCP}	—	79.718%	59.025%	82.292%	87.084%
Prediction Results of Flu Trends Time Series					
Original Values		Predicted Values			
Index	Values	LR	ARMA	MANoD	SFE
158	0.61100	0.43150	1.39350	0.72190	0.63670
159	0.56900	0.37320	1.18290	0.65340	0.52800
...
238	1.46800	1.62090	1.15970	1.55570	1.46820
239	1.56600	1.97570	1.05280	1.73290	1.65600
...
314	0.78000	0.65770	2.02740	0.91340	0.70200
315	0.73900	0.72000	1.42310	0.85320	0.69700
E_{MAE}	—	0.35962	0.86375	0.26042	0.1469
E_{CCP}	—	87.631%	84.976%	91.849%	96.78%

Table 5.3: Prediction Results Comparison (cont'd)

Prediction Results of Nile River Flow Time Series					
Original Values		Predicted Values			
Index	Values	LR	ARMA	MANoD	SFE
181	89.3548	73.6985	28.5130	36.4666	94.0426
182	79.6429	67.9064	63.6096	72.4597	86.9684
...
269	100.000	98.7068	114.099	128.360	132.839
270	121.667	311.937	184.469	195.111	219.467
...
359	153.333	113.432	113.331	90.8656	112.622
360	119.355	114.394	136.895	117.357	123.970
E_{MAE}	—	99.1768	100.449	100.365	99.3569
E_{CCP}	—	48.055%	44.234%	51.422%	87.46%
Prediction Results of Sunspot Number Time Series					
Original Values		Predicted Values			
Index	Values	LR	ARMA	MANoD	SFE
601	59.9000	35.9193	73.6607	69.3256	61.0900
602	59.9000	36.4420	72.6889	75.1463	55.9000
...
900	7.80000	20.2226	16.6749	22.5713	8.61520
901	8.10000	20.5592	10.0164	1.65010	10.0458
...
1199	106.800	74.662	106.830	97.7593	125.665
1200	104.400	94.788	113.276	119.184	94.9363
E_{MAE}	—	25.2253	23.4995	17.5425	14.8258
E_{CCP}	—	67.03%	74.90%	82.18%	97.83%

Table 5.4: Prediction Results Comparison (cont'd)

Prediction Results of Synthetic Pseudo-Periodical Time Series					
Original Values		Predicted Values			
Index	Values	LR	ARMA	MANoD	SFE
50001	-8.93e-5	-1.78e-4	-8.96e-5	-8.96e-5	-8.94e-4
50002	-1.78e-4	-2.67e-4	-1.77e-4	-3.43e-4	-1.77e-4
...
75000	-0.1096	-0.1095	-0.1096	-0.1094	-0.1096
75001	-0.1094	-0.1093	-0.1094	-0.1097	-0.1093
...
100000	-9.79e-4	-4.91e-4	-1.50e-4	-3.39e-4	-8.14e-4
100001	-4.89e-4	-1.15e-4	-1.27e-4	-3.44e-4	-4.39e-4
E_{MAE}	—	0.13964	0.11480	0.10487	0.06687
E_{CCp}	—	94.98%	96.61%	96.87%	99.16%

As showed from Fig 5.1 to Fig 5.15 and in Table 5.2, Table 5.3 and Table 5.4, the proposed methods provides a better performance and accuracy for prediction than classical methods *Linear Regression* (LR) and *Auto-Regression Moving Average* (ARMA), where used a same length of regression (10 backward data points); Meanwhile, they also have lower *Mean Absolute Error* (MAE) values and higher *Correlation Coefficient percentage* (CCp) values for total.

As seen in Table 5.4, Series Features Extraction (SFE) algorithm works better than other three algorithms, it produced extremely correct prediction results on synthetic pseudo-periodical time series.

For the most noisy data sequences, earthquakes time series (some researcher

consider it as a random time series due to the frangible (pseudo-)periodicity), ARMA gained the worst prediction results, because ARMA's prediction results have a delay (lag) on time domain and as the Fig 5.6 shown, most of prediction values less than actual values. In contrast, the proposed algorithms, MANoD and SFE produced better results and estimated a good trends of time series.

Also, that MANoD and SFE predicted for flu trends in U.S. time series accurate results both on the trends and on the data values. Due to that flu trends time series is a relatively smooth time series curve, SFE could find more matched patterns (up/down, peak/valley) from historical data values, thereby, it gave even better predicting results.

Nile River Flow time series has a high pseudo-periodical time series, so that regression prediction methods could provide accurate time series' trends. However, the overlap calculation of regression could accumulate the prediction errors, as Fig 5.3 to Fig 5.18 show, and all four algorithms can compute the trends and direction of time series but not for the peak/valley values, classical algorithms produced a number of impossible (negative) values. But for the overall prediction, SFE still has the best prediction results.

Sunspot Number time series is commonly considered as the most interrelated time series, SFE is designed for recognizing the patterns from historical data sequences, as a result, SFE is the best in the four algorithms for prediction of sunspot number time series. Besides that, MANoD also produced very good prediction results.

The synthetic pseudo-periodical time series is generated by a mathematical function, and it is designed for testing indexing schemes in time series data sets [KDDArchive, 2007a]. The data appears highly periodical but never exactly repeats itself. As a results, all four algorithms produced excellent results, such 94% plus on order of accuracy. Whereas, SFE had the best accuracy on prediction.

5.3 Summary

This chapter has introduced the concepts of classical time series prediction methods, “Linear Regression (LR)” and “Auto-Regression Moving Average (ARMA)”.

For the evaluation of prediction algorithms, this chapter presented prediction results on five different testing data sets: Earthquakes, Flu Trends, Nile River Flow, Sunspot Number and Synthetic pseudo-periodical time series. *Mean Absolute Error* (MAE) and *Correlation Coefficient percentage* (CCp) have been used to measure the performances.

In the last section of this chapter, the predicted results have been illustrated for the four different methods; and computed by the measures of MAE and CCp.

Moving Average of n^{th} -order Difference (MANoD) and *Series Features Extraction* (SFE) algorithms do have good abilities to predict the forthcoming

values for pseudo-periodical time series.

MANoD proposes a simple approach to determine the range of values necessary for prediction. With increasing the order of difference, MANoD would increase the prediction accuracy. However, it could also increase the computational complexity.

SFE proposes an approach to collect and recognize as many patterns with same structure as possible. The algorithm keeps manipulating the *a priori* information (features) and knowledge (patterns) to predicting precisely. However, SFE could be not suitable for an insufficient data series with incorrect data values, because SFE approaching system thinks all imported time series data are properly correct.

Because the disadvantage of SFE approach is what if there is no pre-specified pattern matched in the historical data sequence at all. Currently, if no matches returned, we think the latest data value may stay the same; however, it could be caused by the shortage of data samples, or that existing time series data set is insufficient. In others words, SFE approaching system thinks all imported time series data are properly correct, it is not able to distinguish the wrong/error data. Consequently, we like to extend SFE algorithm to fit both on complete and on in-complete time series, i.e. develop a detecting method as the action of data cleaning.

Chapter 6

Conclusions

Contents

6.1	Summary of Research	141
6.2	Original Contributions	143
6.3	Future Work	144
6.4	Final Remarks	146

6.1 Summary of Research

The main topic of this thesis is the development of novel time series analysis and prediction approaches. The principle of current data mining approaches on time series is to analyze a finite period of existing data observations, then search back into historical data over a time interval. In the contrast, the proposed algorithms provide novel approaches: they do not depend absolutely on the length of existing time series data and also produce an accurate prediction by studying trends and features of the pseudo-periodical time series.

The *Moving Average of n^{th} -order Difference* algorithm presents a simple approach to determine the range of values necessary for a good prediction of the time series terms in cases of bounded pseudo-periodical time series. The developed algorithm to predict time series based on a number of previous known values necessarily addresses also the noise of the actual collected measurements of a time series. The errors obtained by the algorithm in this thesis are represented as differences between actual and expected value of average sum (differences of moving averages). The method also provides a logical development in a transparent way, avoiding the use of “Black Box” methods.

Therefore, the *Moving Average of n^{th} -order Difference* algorithm would generate an accurate prediction results for time series with unequal cyclic periods. With the order of difference increasing, the prediction methods will acquire more precise results. Although the error detecting and avoiding would increase the system complexity, by using the feedback to revise the model itself as a new series, the moving average will also decrease the computational

complexity after all.

The *Series Features Extraction* algorithm proposes an approach to collect and recognize as many patterns with same structure as possible. It is the most advanced part of the algorithm that keeps manipulating the *a priori* information (features) and knowledge (patterns). Meanwhile, the SFE algorithm temporarily ignores the measures of data values, which are the most important observations from the view points of the classical methods. This is a key way of our original approach to decompose the entire time series database; if the data from the value/domain would no longer interfere with the data from the time-domain, the prediction will be localized by a lower complex calculation process. Because classical methods/approaches take both value and trends (pattern) into account.

Furthermore, beyond the time series analysis domain, the *Series Features Extraction* algorithm can be applied on a general data sets, no matter how the data sequence fluctuation is, because SFE algorithm focuses on the nature of data changing, and projects all data change into finite classes; therefore, the prediction progress is simplified. Case studies using real world and synthetic time series have been used to demonstrate the efficiency and performance of the proposed algorithm. Our results are very good, comparable and better than those obtained with classical methods.

6.2 Original Contributions

I can state the following original contributions of this thesis:

- Extension of regression analysis approaches for time series analysis and prediction;
- New computational approaches based on data-driven methods for generic pseudo-periodical time series: *Moving Average of n^{th} -order Difference (MANoD)* [Lan and Neagu, 2007b] described in chapter 3 and *Series Features Extraction (SFE)* proposed in chapter 4;
- Two original time series analysis models and prediction algorithms: the *Moving Average of n^{th} -order Difference (MANoD)* algorithm [Lan and Neagu, 2007a] and *Series Features Extraction (SFE)* algorithm proposed in chapter 4;
- Original *unsupervised learning* methods for handling the *a priori* knowledge for analysis and prediction as described in chapter 4;
- New automated feature detection, extraction, classification and clustering techniques proposed in chapter 4;
- Test and implementation of the proposed approaches on various pseudo-periodical time series data sets, published in [Lan and Neagu, 2007b] [Lan and Neagu, 2007a] [Lan and Neagu, 2006];
- Study of the classical methods and comparison of the prediction results in terms of flexibility and performance as described in chapters 2 and 4.

6.3 Future Work

Once we developed and studied the performance of the *Moving Average of n^{th} -order Difference* and *Series Features Extraction* algorithms, we have found there are some future directions for research work as described below.

The algorithm MANoD has the disadvantage of dependency of the (still) error between the moving average of n^{th} -order difference values at the prediction step, $n + 1$ and n . The MANoD algorithm generates therefore a good prediction for the trends of the time series (including the pseudo-periodicity), but the precision of prediction (amplitude) suffers because of dependency on how many orders (i.e. value of n) difference have been considered, which increases the complexity calculus though and introduces a tuning parameter of the order of difference. A small order of difference reduces the computational complexity but also the prediction precision, whereas a large order of difference increases the computing effort. This can be described as an optimization problem and further work may focus on choosing the right value for n .

Another direction for further research is the approximation of error in using machine learning techniques, in order to reduce the differences induced by the possibility to obtain a non-zero average of n^{th} -order difference for a period close to the prediction moment. Provisional encouraging results to approximate the error using a neural network model have been already obtained and presented in chapter 3: the connectionist model is trained with the error values for the first 600 cases of Sunspot Number time series. We

propose to study the development of a hybrid model based on the algorithm proposed above by using the moving average of n^{th} -order difference time series prediction and also another synchronous prediction of current error given by the trained neural network in an optimized context of a tuned order of the difference.

SFE time series prediction algorithm has the advantage of a patterns recognition method, it can compare, filter and cluster the matched patterns to predict better. However, there is an estimation of regression coefficients at the last step in SFE approach, and the regression prediction itself brings errors. Regression prediction estimates the *conditional expectation* of a dependent variable given the independent variables, but this could accumulate prediction errors. As a result, a direction for further research on SFE approach is the approximation of regression coefficients by using advanced machine learning method, in order to produce an accurate prediction.

Another disadvantage of SFE approach is what if there is no pre-specified pattern matched in the historical data sequence at all. Currently, if no matches returned, we think the latest data value may stay the same; however, it could be caused by the shortage of data samples, or that existing time series data set is insufficient. In others words, SFE approaching system thinks all imported time series data are properly correct, it is not able to distinguish the wrong/error data. Consequently, we like to extend SFE algorithm to fit both on complete and on in-complete time series, i.e. develop a detecting method as the action of *data cleaning*.

6.4 Final Remarks

It is our hope that this thesis and the relevant research work have contributed to the field of *Time Series Analysis and Prediction*. By introducing two original approaches focused on predicting the future values and applied on pseudo-periodical time series, we have also demonstrated their applicability and performance. These two approaches are not only to extend classical technologies (*Regression*) but also provide new ways to obtain the *a priori* knowledge from the original time series data sets.

Finally, I conclude hoping that this research work will give others a good inspiration for a better tomorrow!

Appendix A

The mathematics proof (based on Peano's Induction Axiom) for "a n^{th} -order difference equals the difference of two lower $((n - 1)^{\text{th}}$ -order) differences" (used in eq.(3.4)):

To Be Proved:
$$D_m^n = D_{m+1}^{n-1} - D_m^{n-1}$$

Proof:

$$\begin{aligned}
D_{m+1}^{n-1} &= \sum_{i=0}^{n-1} \frac{(-1)^{(n-1)-i} \cdot (n-1)! \cdot a_{((m+1)+i)}}{i!((n-1)-i)!} \\
&= \frac{(-1)^{(n-1)-0} \cdot (n-1)! \cdot a_{((m+1)+0)}}{0!((n-1)-0)!} \\
&\quad + \frac{(-1)^{(n-1)-1} \cdot (n-1)! \cdot a_{((m+1)+1)}}{1!((n-1)-1)!} \\
&\quad + \dots \\
&\quad + \frac{(-1)^{(n-1)-(n-2)} \cdot (n-1)! \cdot a_{((m+1)+(n-2))}}{(n-2)!((n-1)-(n-2))!} \\
&\quad + \frac{(-1)^{(n-1)-(n-1)} \cdot (n-1)! \cdot a_{((m+1)+(n-1))}}{(n-1)!((n-1)-(n-1))!} \\
D_m^{n-1} &= \sum_{i=0}^{n-1} \frac{(-1)^{(n-1)-i} \cdot (n-1)! \cdot a_{(m+i)}}{i!((n-1)-i)!} \\
&= \frac{(-1)^{(n-1)-0} \cdot (n-1)! \cdot a_{(m+0)}}{0!((n-1)-0)!} \\
&\quad + \frac{(-1)^{(n-1)-1} \cdot (n-1)! \cdot a_{(m+1)}}{1!((n-1)-1)!} \\
&\quad + \dots
\end{aligned}$$

$$\begin{aligned}
 & + \frac{(-1)^{(n-1)-(n-2)} \cdot (n-1)! \cdot a_{(m+(n-2))}}{(n-2)!((n-1)-(n-2))!} \\
 & + \frac{(-1)^{(n-1)-(n-1)} \cdot (n-1)! \cdot a_{(m+(n-1))}}{(n-1)!((n-1)-(n-1))!} \\
 D_{m+1}^{n-1} - D_m^{n-1} & = \sum_{i=0}^{n-1} \frac{(-1)^{(n-1)-i} \cdot (n-1)! \cdot a_{((m+1)+i)}}{i!((n-1)-i)!} \\
 & - \sum_{i=0}^{n-1} \frac{(-1)^{(n-1)-i} \cdot (n-1)! \cdot a_{(m+i)}}{i!((n-1)-i)!} \\
 & = \frac{(-1)^{(n-1)-(n-1)} \cdot (n-1)! \cdot a_{(m+1)+(n-1)}}{(n-1)!((n-1)-(n-1))!} \\
 & + \left(\frac{(-1)^{(n-1)-(n-2)} \cdot (n-1)! \cdot a_{(m+1)+(n-2)}}{(n-2)!((n-1)-(n-2))!} \right. \\
 & \quad \left. - \frac{(-1)^{(n-1)-(n-1)} \cdot (n-1)! \cdot a_{m+(n-1)}}{(n-1)!((n-1)-(n-1))!} \right) \\
 & + \dots \\
 & + \left(\frac{(-1)^{(n-1)-0} \cdot (n-1)! \cdot a_{(m+1)+0}}{0!((n-1)-0)!} \right. \\
 & \quad \left. - \frac{(-1)^{(n-1)-1} \cdot (n-1)! \cdot a_{m+1}}{1!((n-1)-1)!} \right) \\
 & - \frac{(-1)^{(n-1)-0} \cdot (n-1)! \cdot a_{m+0}}{0!((n-1)-0)!} \\
 & = \frac{n}{n} \cdot \frac{(-1)^{n-n} \cdot (n-1)! \cdot a_{m+n}}{0!(n-1)!}
 \end{aligned}$$

$$\begin{aligned}
& + \left(\frac{n-1}{n} \cdot \frac{(-1)^{(n-(n-1))} \cdot n! \cdot a_{m+n-1}}{(n-1)!(n-(n-1))!} \right. \\
& \quad \left. - (-1)^{-1} \cdot \frac{1}{n} \cdot \frac{(-1)^{(n-(n-1))} \cdot n! \cdot a_{m+n-1}}{(n-1)!(n-(n-1))!} \right) \\
& + \dots \\
& + \left(\frac{1}{n} \cdot \frac{(-1)^{(n-1)} \cdot n! \cdot a_{m+1}}{1!(n-1)!} \right. \\
& \quad \left. - (-1)^{-1} \cdot \frac{n-1}{n} \cdot \frac{(-1)^{(n-1)} \cdot n! \cdot a_{m+1}}{1!(n-1)!} \right) \\
& + \frac{n}{n} \cdot \frac{(-1)^{(n-0)} \cdot (n-1)! \cdot a_{m+0}}{0!((n-1)-0)!} \\
& = \frac{(-1)^{n-n} \cdot n! \cdot a_{m+n}}{n!(n-n)!} \\
& \quad + \frac{(-1)^{n-(n-1)} \cdot n! \cdot a_{m+(n-1)}}{(n-1)!(n-(n-1))!} \\
& + \dots \\
& \quad + \frac{(-1)^{n-1} \cdot n! \cdot a_{m+1}}{1!(n-1)!} \\
& \quad + \frac{(-1)^{n-0} \cdot n! \cdot a_{m+0}}{0!(n-0)!} \\
& = \sum_{i=0}^n \frac{(-1)^{n-i} \cdot n! \cdot a_{m+i}}{i!(n-i)!} \\
& = D_m^n
\end{aligned}$$

Q.E.D.

Appendix B

This is the complete proof (based on Peano's Induction Axiom) of the following equation (used in eq.(3.17)):

To Be Proved:
$$E_m^n = \frac{1}{m}(D_{m+1}^{n-1} - D_1^{n-1})$$

Proof:

when $n = 1$ and $n = 2$:

$$\begin{aligned} E_m^1 &= \frac{1}{m} \sum_{k=1}^m D_k^1 \\ &= \frac{1}{m} ((a_{m+1} - a_m) + (a_m - a_{m-1}) + (a_{m-1} - a_{m-2}) \\ &\quad + \cdots + (a_4 - a_3) + (a_3 - a_2) + (a_2 - a_1)) \\ &= \frac{1}{m} (a_{m+1} - a_1) \\ &= \frac{1}{m} (D_{m+1}^0 - D_1^0) \\ E_m^2 &= \frac{1}{m} \sum_{k=1}^m D_k^2 \\ &= \frac{1}{m} ((a_{m+2} - 2a_{m+1} + a_m) + (a_{m+1} - 2a_m + a_{m-1}) \\ &\quad + \cdots + (a_4 - 2a_3 + a_2) + (a_3 - 2a_2 + a_1)) \\ &= \frac{1}{m} ((a_{m+2} - a_{m+1}) - (a_2 - a_1)) \\ &= \frac{1}{m} (D_{m+1}^1 - D_1^1) \end{aligned}$$

The statement to be proved above was verified about for $n = 1$ and $n = 2$.

we assumed the statement is true for n :

$$E_m^n = \frac{1}{m} \sum_{k=1}^m D_k^n = \frac{1}{m} (D_{m+1}^{n-1} - D_1^{n-1})$$

And demonstrate the relationship is true $n + 1$:

$$\begin{aligned} E_m^{n+1} &= \frac{1}{m} \sum_{k=1}^m D_k^{n+1} \\ &= \frac{1}{m} (D_m^{n+1} + D_{m-1}^{n+1} + D_{m-2}^{n+1} \\ &\quad + \cdots + D_3^{n+1} + D_2^{n+1} + D_1^{n+1}) \\ &= \frac{1}{m} ((D_{m+1}^n - D_m^n) + (D_m^n - D_{m-1}^n) + (D_{m-1}^n - D_{m-2}^n) \\ &\quad + \cdots + (D_4^n - D_3^n) + (D_3^n - D_2^n) + (D_2^n - D_1^n)) \\ &= \frac{1}{m} (D_{m+1}^n - D_1^n) \end{aligned}$$

Q.E.D.

References

- [Aho et al., 1983] Aho, A. V., Ullman, J. D., and Hopcroft, J. E. (1983). *Data Structures and Algorithms*. Addison Wesley. ISBN: 0201000237.
- [Beizer, 1995] Beizer, B. (1995). *Black Box Testing*. John Wiley & Sons, Inc. ISBN: 0471120944.
- [Bethea, 1995] Bethea, R. M. (1995). *Statistical Methods for Engineers and Scientists*. CRC. ISBN: 0824793358.
- [Beveridge, 1921] Beveridge, W. H. (1921). Weather and harvest cycles. *Economic Journal*, 31(124):429–452.
- [Björck, 1996] Björck, Å. (1996). *Numerical Methods for Least Squares Problems*. Society for Industrial Mathematics. ISBN: 0898713609.
- [Bollerslev, 1986] Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327. <http://www.sciencedirect.com/science/article/B6VC0-46VV78N-4/2/ce371daca28a0d38824736988b1d0ef1>.
- [Box and Jenkins, 1976] Box, G. and Jenkins, G. M. (1976). *Time Series Analysis: Forecasting & Control*. Holden-Day, second edition. ISBN: 0816211043.
- [Box et al., 1994] Box, G., Jenkins, G. M., and Reinsel, G. (1994). *Time Series Analysis: Forecasting & Control*. Prentice Hall, third edition. ISBN: 0130607746.

- [Britannica, 2008] Britannica (2008). Epistemology. <http://www.britannica.com/EBchecked/topic/190219/epistemology>. Britannica Encyclopædia.
- [Chatfield, 2003] Chatfield, C. (2003). *The Analysis of Time Series: An Introduction*. Chapman & Hall. ISBN: 1584883170.
- [Chatterjee and Hadi, 2006] Chatterjee, S. and Hadi, A. S. (2006). *Regression Analysis by Example*. Wiley-Blackwell, forth edition. ISBN: 0471746967.
- [Chou, 1975] Chou, Y.-L. (1975). *Statistical Analysis*. Holt, R & W. ISBN: 0030894220.
- [Cohen et al., 2002] Cohen, J., Cohen, P., West, S. G., and Aiken, L. S. (2002). *Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences*. Lawrence Erlbaum, third edition. ISBN: 0805822232.
- [Dickie, 1996] Dickie, G. (1996). *The Century of Taste: The Philosophical Odyssey of Taste in the Eighteenth Century*. OUP USA. ISBN: 0195096800.
- [Easton and McColl, 2008] Easton, V. J. and McColl, J. H. (accessed 2008). Statistics glossary. <http://www.stats.gla.ac.uk/steps/glossary/index.html>. Statistics Glossary from SETPS v1.1.
- [Edwards, 1976] Edwards, A. L. (1976). *Introduction to Linear Regression and Correlation*. W.H.Freeman & Co Ltd. ISBN: 0716705613.

- [Engle, 1982] Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, 50(4):987–1007. <http://www.jstor.org/stable/1912773>.
- [Flajolet and Sedgewick, 1995] Flajolet, P. and Sedgewick, R. (1995). Mellin transforms and asymptotics: Finite differences and rice’s integrals. *Theoretical Computer Science*, 144(1–2):101–124. [http://dx.doi.org/10.1016/0304-3975\(94\)00281-M](http://dx.doi.org/10.1016/0304-3975(94)00281-M).
- [Gensler, 2001] Gensler, H. (2001). *Introduction to Logic*. Routledge. ISBN: 0415226740.
- [Goodman and Hedetniemi, 1977] Goodman, S. E. and Hedetniemi, S. T. (1977). *Introduction to the Design and Analysis of Algorithms*. Mcgraw-Hill College. ISBN: 0070237530.
- [GoogleTrends, 2009] GoogleTrends (2009). Flu trends in united states time series. http://www.google.org/flutrends/intl/en_us/.
- [Hand et al., 2001] Hand, D. J., Mannila, H., and Smyth, P. (2001). *Principles of Data Mining*. The MIT Press. ISBN: 026208290X.
- [Hannan and Kavalieris, 1984] Hannan, E. J. and Kavalieris, L. (1984). A method for autoregressive-moving average estimation. *Biometrika*, 71(2):273–280.
- [Hannan and Rissanen, 1982] Hannan, E. J. and Rissanen, J. (1982). Recursive estimation of mixed autoregressive-moving average order. *Biometrika*, 69(1):81–94.

- [Hinde, 1998] Hinde, A. (1998). *Demographic Methods*. Hodder Education. ISBN: 0340718927.
- [Hyndman, 2009] Hyndman, R. J. (2009). Time series library. <http://www.robjhyndman.com/TSDL/>.
- [IEP, 2008] IEP (2008). Epistemology. <http://www.iep.utm.edu/e/epistemo.htm>. The Internet Encyclopædia of Philosophy (IEP).
- [Ishikawa, 1986] Ishikawa, K. (1986). *Guide to Quality Control*. Asian Productivity Organization, second edition. ISBN: 9283310357.
- [Kallenberg, 2002] Kallenberg, O. (2002). *Foundations of Modern Probability*. Springer, second edition. ISBN: 0387953132.
- [KDDArchive, 2007a] KDDArchive (2007a). Pseudo-periodic synthetic time series. <http://kdd.ics.uci.edu/databases/synthetic/synthetic.data.html>. University of California, Irvine.
- [KDDArchive, 2007b] KDDArchive (2007b). Uci knowledge discovery in database archive. <http://kdd.ics.uci.edu/>. University of California, Irvine.
- [Korn and Korn, 2000] Korn, G. A. and Korn, T. M. (2000). *Mathematical Handbook for Scientists and Engineers: Definitions, Theorems, and Formulas for Reference and Review*. Dover Publications, second edition. ISBN: 0486411478.
- [Kotsiantis et al., 2006] Kotsiantis, S., Zaharakis, I., and Pintelas, P. (2006). Machine learning: A review of classification and combining techniques.

- Artificial Intelligence Review*, 26(3):159–190. <http://www.springerlink.com/content/p602226847634v14/>.
- [Kotsiantis, 2007] Kotsiantis, S. B. (2007). Supervised machine learning: A review of classification techniques. *Informatica*, 31:249–268. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.95.9683>.
- [Lan and Neagu, 2006] Lan, Y. and Neagu, D. (2006). A new algorithm based on the average sum of n^{th} -order difference for time series prediction. *The Sixth annual UK Workshop on Computational Intelligence (UKCI)*, 1:183–189.
- [Lan and Neagu, 2007a] Lan, Y. and Neagu, D. (2007a). Applications of the moving average of n^{th} -order difference algorithm for time series prediction. *The Third International Conference on Advanced Data Mining and Applications (ADMA)*, 4632/2007:264–275.
- [Lan and Neagu, 2007b] Lan, Y. and Neagu, D. (2007b). A new time series prediction algorithm based on moving average of n^{th} -order difference. *The Sixth International Conference on Machine Learning and Applications (ICMLA)*, 1:248–253.
- [Larsen and Marx, 2005] Larsen, R. J. and Marx, M. L. (2005). *An Introduction to Mathematical Statistics and Its Applications*. Prentice Education, fourth edition. ISBN: 0132018136.
- [Mills, 1990] Mills, T. C. (1990). *Time Series Techniques for Economists*. Cambridge University Press. ISBN: 0521405742.

- [Minsky and Papert, 1969] Minsky, M. and Papert, S. (1969). *Perceptrons: An Introduction to Computational Geometry*. The MIT Press. ISBN: 0262130432.
- [NGDC, 2006] NGDC (2006). Monthly average sunspot number time series. <http://www.ngdc.noaa.gov/stp/SOLAR/ftpsunspotregions.html>. National Geophysical Data Center (NGDC).
- [Papoulis and Pillai, 2002] Papoulis, A. and Pillai, S. U. (2002). *Probability, Random Variables and Stochastic Processes with Errata Sheet*. McGraw-Hill Higher Education, forth edition. ISBN: 0071226613.
- [Pearl, 1984] Pearl, J. (1984). *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley Pub. ISBN: 0201055945.
- [Percival and Walden, 1993] Percival, D. B. and Walden, A. T. (1993). *Spectral Analysis for Physical Applications*. Cambridge University Press. ISBN: 0521435412.
- [Scruton et al., 2001] Scruton, R., Singer, P., Janaway, C., Tanner, M., and Thomas, K. (2001). *German Philosophers: Kant, Hegel, Schopenhauer, Nietzsche*. Oxford Press. ISBN: 0192854240.
- [Seber and Wild, 2003] Seber, G. A. F. and Wild, C. J. (2003). *Nonlinear Regression*. Wiley-Interscience. ISBN: 0471471356.
- [SEP, 2008] SEP (2008). Epistemology. <http://plato.stanford.edu/entries/epistemology/>. Stanford Encyclopædia of Philosophy (SEP).

- [Strang, 2003] Strang, G. (2003). *Introduction to Linear Algebra*. Wellesley-Cambridge Press, U.S., third edition. ISBN: 0961408898.
- [USCDC, 2009] USCDC (2009). Data & statistics. <http://www.cdc.gov/DataStatistics/>. United States Center for Disease Control and Prevention (USCDC).
- [Werbos, 1994] Werbos, P. J. (1994). *Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting*. John Wiley & Sons, Inc. ISBN: 0471598976.
- [Wikipedia., 2004] Wikipedia. (2004). Sunspot number. http://en.wikipedia.org/wiki/Sunspot_number. wikipedia.com.
- [Yuille, 2009] Yuille, B. (2009). Short selling: What is short selling? <http://www.investopedia.com/university/shortselling/shortselling1.asp>. Investopedia.com.
- [Yule, 1926] Yule, G. U. (1926). Why do we sometimes get nonsense-correlations between time-series? — a study in sampling and the nature of time-series. *Journal of the Royal Statistical Society*, 89(1):1–63.