Fuzzy-Wavelet Method for
Time Series Analysis

Ademola Olayemi Popoola

Submitted for the Degree of
Doctor of Philosophy
from the
University of Surrey

Department of Computing
School of Electronics and Physical Sciences
University of Surrey
Guildford, Surrey GU2 7XH, UK

January 2007

© Ademola Popoola 2007
Abstract

Fuzzy systems are amongst the class of soft computing models referred to as universal approximators. Fuzzy models are increasingly used in time series analysis, where it is important to deal with trends, variance changes, seasonality and other patterns. For such data that exhibit complex local behaviour, universal approximation may be inadequate. An investigation of the effectiveness of subtractive clustering fuzzy models in analyzing time series that are deemed to have trend and seasonal components indicates that, in general, forecast performance improves when pre-processed data is used. A general pre-processing method, based on multiscale wavelet decomposition, is used to provide a local representation of time series data prior to the application of fuzzy models.

The novelty in this work is that, unlike wavelet-based schemes reported in the literature, our method explicitly takes the statistical properties of the time series into consideration, and only recommends wavelet-based pre-processing when the properties of the data indicate that such pre-processing is appropriate. In particular, time series that exhibit changes in variance require pre-processing, and wavelet-based pre-processing provides a parameter-free method for decomposing such time series. Conversely, wavelet-based pre-processing of time series with homogeneous variance structure leads to worse results compared to an equivalent analysis carried out using raw data. The wavelet variance profile of a time series, and an automatic method for detecting variance breaks in time series, are used as indicators as to the suitability of wavelet-based pre-processing. This approach, consistent with Occam's razor, facilitates the application of our framework to the different characteristics exhibited in real-world time series.
Acknowledgements

And I thought the PhD was challenging. Now, the prospect of recalling all the people and institutions that have made this exciting journey possible, and fitting my appreciation to this single page, is even more daunting. Well, here goes...

Special thanks go to my supervisor, Professor Ahmad, for providing me the opportunity to embark on this journey. He has questioned and supported me, challenged my intellect, and routinely gone beyond the call of duty in offering assistance and advice.

Many thanks go to my friends and colleagues, in no particular order - Okey, Juhani, Elizabeth, Saif, David, Hayssam, Tugba, Mimi, Rafif- and to members of staff of the Department of Computing – Lydia, Sophie, Noelle, Kelly, Lee, Nick, Bogdan, Mathew, Gary, Michael... the list is endless! Also, I appreciate and gratefully acknowledge the financial support provided by the Department of Computing, University of Surrey throughout the course of my research.

I am grateful to my family for supporting me in this quest for knowledge, and for all the prayers, calls, emails and photos. I appreciate you all. I am especially grateful to my lovely wife, Adetola, who has endured the long hours away from home, ‘think tank’ faraway looks, short phone calls, and all the missed dates. Above all, I thank God for the gift of life, friends and family.
# Contents

Abstract ................................................................................................................. ii  

Acknowledgements ........................................................................................ iii  

Contents ........................................................................................................ iv  

List of Figures .................................................................................................. vi  

List of Tables ................................................................................................... x  

1 Introduction ........................................................................................................ 1  

1.1 Preamble ......................................................................................................... 1  

1.2 Contributions of the Thesis ........................................................................ 6  

1.3 Structure of the Thesis ................................................................................ 7  

1.4 Publications .................................................................................................. 7  

2 Motivation and Literature Review .................................................................. 9  

2.1 Time Series: Basic Notions ......................................................................... 11  

2.1.1 Components of Time Series .................................................................. 11  

2.1.2 Nonstationarity in the Mean and Variance .......................................... 15  

2.2 Time Series Models ..................................................................................... 17  

2.2.1 An Overview of Conventional Approaches ......................................... 17  

2.2.2 Soft Computing Models: Fuzzy Inference Systems ............................ 19  

2.3 Fuzzy Models for Time Series Analysis ................................................... 20  

2.3.1 Grid Partitioning .................................................................................... 22  

2.3.2 Scatter Partitioning: Subtractive Clustering ........................................ 25  

2.3.3 Criticism of Fuzzy-based Soft Computing Techniques ....................... 29  

2.4 Multiscale Wavelet Analysis of Time Series .......................................... 33  

2.4.1 Time and Frequency Domain Analysis .............................................. 33  

2.4.2 The Discrete Wavelet Transform (DWT) ........................................... 35  

2.5 Summary ..................................................................................................... 45  

3 Fuzzy-Wavelet Method for Time Series Analysis ....................................... 46  

3.1 Introduction .................................................................................................. 46  

3.2 Data Pre-processing for Fuzzy Models ..................................................... 48  

3.2.1 Informal Approaches ............................................................................ 49
List of Figures

Figure 1.1. Framework for pre-processing method selection ................................................... 5
Figure 1.2. Wavelet-based pre-processing scheme with diagnosis phase ..................................... 6
Figure 2.1. Time series of IBM stock prices .............................................................................. 9
Figure 2.2. Closing value of the FTSE 100 index from Nov. 2005 – Oct. 2006 fitted with linear trend line ................................................................. 13
Figure 2.3. Women’s clothing sales for January 1992 – December 1996 showing unadjusted (blue) and seasonally adjusted (red) data .................................................. 14
Figure 2.4. Irregular component of women’s clothing sales obtained by assuming (i) a difference stationary trend (blue) and (ii) a trend stationary model (red) ........... 15
Figure 2.5. Time series data that is stationary in the mean and variance .................................... 15
Figure 2.6. Time series data that is nonstationary in the (a) mean and (b) variance. ................. 16
Figure 2.7. Fuzzy partition of two-dimensional input space with $K_1 = K_2 = 5$ (Ishibuchi et al, 1994) ........................................................................................................... 23
Figure 2.8: Mapping time series data points to fuzzy sets (Mendel, 2001) .......................... 24
Figure 2.9. Time series generated from SARIMA(1,0,0)(0,1,1) model ........................................ 27
Figure 2.10. Scatter plot of 3-dimensional vector ..................................................................... 28
Figure 2.11. Clusters generated by the algorithm ..................................................................... 28
Figure 2.12. Forecast accuracy (NDEI) of hybrid models plotted on a log scale ...................... 32
Figure 2.13. Time and frequency plots for random data (top) and noisy periodic data (bottom) ......................................................................................................................... 34
Figure 2.14. Time and frequency plots for data with sequential periodic components ............. 35
Figure 2.15. Time-frequency plane partition using (a) the Fourier transform (b) time domain representation; (c) the STFT (Gabor) transform and (d) the wavelet transform .......... 36
Figure 2.16. (a) Square-wave function mother wavelet (b) wavelet positively translated in time (c) wavelet positively dilated in time and (d) wavelet negatively dilated in time (Gencay, 2001) ...................................................... 37
Figure 2.17. Generating wavelet coefficients from a time series ............................................. 38
List of Figures

Figure 2.18. Flow diagram illustrating the pyramidal method for decomposing $X_t$ into wavelet coefficients $W_j$ and scaling coefficients $V_j$ .............................................. 41

Figure 2.19. Plot of original time series (top) and its wavelet decomposition structure (bottom). .......................................................... 42

Figure 2.20. Flow diagram illustrating the pyramidal method for reconstructing wavelet approximations $S_t$ and details $D_t$ from wavelet coefficients $W_t$ and scaling coefficients $V_t$ .................................................. 43

Figure 2.21. Five-level multiscale wavelet decomposition of time series $X_t$ showing the wavelet approximation, $S_t$, and wavelet details $D_t - D_{t-4}$ ................. 44

Figure 3.1. Time series generated from SARIMA(1,0,0)(0,1,1) model .................. 51

Figure 3.2. Synthetic data 'detrended' using (a) first difference (b) first-order polynomial curve fitting .................................................. 51

Figure 3.3. Synthetic data 'deseasonalised' using (a) seasonal difference (b) 12-month centred MA .................................................. 52

Figure 3.4. Trend-cycle, seasonal and irregular components of simulated data computed using the additive form of classical decomposition method .............. 54

Figure 3.5. Mallat's pyramidal algorithm for wavelet multilevel decomposition .......... 55

Figure 3.6. Simulated time series ($X_t$) and its wavelet components $D_1 - D_4$, $S_4$ .................................................. 56

Figure 3.7. (a) Time plot of AR(1) model with seasonality components (b) sample autocorrelogram for AR(1) process (solid line) and AR(1) process with seasonality components (dashed line). Adapted from Gençay et al. (2001) .................. 57

Figure 3.8. (a) Time plots of AR(1) model without seasonality components (blue) and wavelet smooth $S_4$ (red); (b) sample autocorrelograms for AR(1) process (solid line), AR(1) process with seasonality components (dashed line), and wavelet smooth $S_4$ (red dotted line) ............................................. 58

Figure 3.9. (a) Time plots of aperiodic AR(1) model with variance change between 500-750 (blue) and corresponding wavelet smooth $S_4$ (red); (b) sample autocorrelograms for AR(1) process with variance (solid blue line), and wavelet smooth $S_4$ (red dotted line) ............................................. 59

Figure 3.10. Flowchart of 'informal' and 'formal' pre-processing methods .................. 60

Figure 3.11. Time plot of random series and corresponding PACF plot. None of the coefficients has a value greater than the critical values (blue dotted line) ......... 61

Figure 3.12. Time plot of simulated series and corresponding PACF plot. Coefficients at lags 1, 3, 4, 5, 7, 9, 10, 11 and 12 have values greater than the critical values (blue dotted line) .................................................. 62

Figure 3.13. Plot showing (a) first order autoregressive (AR(1)) process with constant variance and (b) the associated wavelet variance .................................................. 65
Figure 3.14. Plot showing (a) first order autoregressive (AR(1)) process with variance change at $t_{3,500}$ and (b) the associated wavelet variance......................... 65

Figure 3.15. Time plot of random variables with homogeneous variance (top) and associated normalized cumulative sum of squares (bottom).......................... 67

Figure 3.16. Time plot of random variables with variance change at $n_{400}$ and $n_{700}$ (top) and associated normalized cumulative sum of squares (bottom)............. 67

Figure 3.17. Framework of the proposed 'intelligent' fuzzy-wavelet method............... 69

Figure 3.18. Schematic representation of the wavelet/fuzzy forecasting system. $D_1, \ldots D_3$ are wavelet coefficients, $S$ is the signal “smooth”.............................. 70

Figure 4.1. Simulated SARIMA model $(1, 0, 0)(0, 1, 1)_12$ time series.................. 75

Figure 4.2. (a) USCB clothing stores data exhibits strong seasonality and a mild trend; (b) FRB fuels data exhibits nonstationarity in the mean and discontinuity at around the 300th month.................... 77

Figure 4.3. PACF plot for simulated time series.................................................................. 79

Figure 4.4. Model error for raw and pre-processed simulated data.......................... 80

Figure 4.5. Time plot of series 3 (USCB Department) and corresponding PACF plot for raw data.......................................................... 83

Figure 4.6. Model error for raw and pre-processed USCB Department.................. 83

Figure 4.7. Time plot of series 1 (USCB Furniture) and corresponding PACF plot for raw data.......................................................... 84

Figure 4.8. Model error for raw and pre-processed series 1 (USCB Furniture) data.... 84

Figure 4.9. Time plot of series 5 (FRB Durable goods) and PACF plot for raw data (top panel); time plot of seasonally differenced series 5 and related PACF plot (bottom panel); 85

Figure 4.10. Model error for raw and pre-processed FRB Durable Goods series........ 85

Figure 4.11. (a) 3-dimensional scatter plot of series 5 using raw data (b) Four rule clusters automatically generated to model the data................................. 86

Figure 4.12. (a) 3-dimensional scatter plot of series 5 using FD data (b) One rule cluster automatically generated to model the data............................... 87

Figure 4.13. (a) 3-dimensional scatter plot of series 5 using SD data (b) Six rule clusters automatically generated to model the data............................... 87

Figure 4.14. (a) 3-dimensional scatter plot of series 5 with SD+FD data (b) One rule cluster automatically generated to model the data............................... 88
Figure 4.15. (a) Cumulative energy profile of 5-level wavelet transform for FRB Durable goods time series (b) A closer look at the energy localisation in $S_5 (t = 0, 1, \ldots, 27)$. ................................................................. 96

Figure 4.16. Multiscale wavelet variance plots for wavelet-processed data showing best (left column) and worst (right column) performing series ........................................ 98

Figure 4.17. Multiscale wavelet variance for time series plotted on a log scale. Plots 1-4 and 10 indicate inhomogeneous variance structure; 5-9 exhibit homogeneous structure, though with noticeable discontinuities between scales 1 and 2 for plots 5 and 9................................................................. 100

Figure 4.18. FRB durable goods series ($X_t$) and its wavelet components $D_1$-$D_3$, $S_3$ ..... 103

Figure 4.19. Scatter plot of series FRB durable goods series and associated rule clusters for $D_1$ (a); $D_2$, (b); $D_3$ (c); and $S_3$ (d). ................................................................. 104

Figure 4.20. Wavelet variance profiles of time series where hypothesis test: (a) correctly detects homogeneity of variance; (b) correctly detects variance inhomogeneity and (c) fails to detect variance inhomogeneity ................. 105
List of Tables

Table 2.1  Episodes in IBM time series and the corresponding linguistic description ........... 10
Table 2.2  Execution stages of fuzzy inference systems ....................................................... 20
Table 2.3  Exemplar application areas of fuzzy models and hybrids for time series analysis .......... 21
Table 2.4  Forecast results of different hybrid models on Mackey-Glass data set ................ 31
Table 3.1  PACF values at different lags 1-20, and critical values + 0.0885 ..................... 62
Table 3.2  PACF values at different lags 1-20 showing positive significant values (boldface) at critical values + 0.0885 .......................................................... 63
Table 4.1  Real-world economic time series used for experiments ................................. 76
Table 4.2  Average MAPE performance on simulated data using different pre-processing methods ................................................................. 79
Table 4.3  Minimum and maximum MAPE for each of the ten series and the pre-processing technique resulting in minimum error ............................... 81
Table 4.4  PACF-based recommendations and actual results ............................................. 82
Table 4.5  Ratio of the number of rule clusters using specific pre-processing method to the maximum number of clusters generated using any data, and corresponding MAPE forecast performance ........................................... 89
Table 4.6  Comparison of RMSE on naive random walk model and subtractive clustering fuzzy models on raw data ..................................................... 91
Table 4.7  Comparison of RMSE on AR-TDNN^a, TDNN^b (Taskaya-Temizel and Ahmad, 2005) ARIMA^a, ARIMA-NN^d (Zhang and Qi, 2005), and fuzzy models^c ........................................ 92
Table 4.8  Comparison of MAPE on raw, informal (ad hoc) and formal (wavelet processed) data using fuzzy clustering model ........................................ 94
Table 4.9  Aggregate forecast performance (MAPE) when the contribution of fuzzy models generated from each wavelet component is excluded ............................. 96
Table 4.10 Pre-processing Method Selection: Comparison of Algorithm Recommendations and Actual (Best) Methods ................................................. 99
Table 4.11 Comparison of Forecast Performance (MAPE) of Fuzzy Models Derived from Wavelet and Box-Cox Transformed Data (Worse Results Relative to Raw Data shown in boldface) .......... 102
Chapter 1

Introduction

1.1 Preamble

The Oxford English Dictionary (OED) defines a time series as 'the sequence of events which constitutes or is measured by time.' Time series are used to characterize the time course of the behaviour of a wide variety of biological, physical and economic systems. Brain waves are represented as time ordered events, and electrocardiograms produce time-based traces of heart waves. In meteorology, wind speed, temperature, pressure, humidity, and rainfall measurements over time are associated with weather conditions. Geophysical records include time-indexed measurements of movements of the earth, and the presence of radioactivity in the atmosphere. Industrial production data, interest rates, inflation, stock prices, and unemployment rates, amongst other time serial data, provide a measure of the health of an economy. In general, phenomena of interest are observed by looking at key variables over time – either continuously or discretely.

The ubiquity of time series makes the study of such data important and, for centuries, people have been fascinated by, and attempted to understand events that vary with time. Records of the annual flow of the River Nile have existed as early as the year 622 A.D., and astrophysical phenomena like sunspot numbers have been recorded since the 1600s. The interest in time varying events ranges from gaining better understanding of the underlying system producing the time series, to being able to foretell the future evolution of the data generating process. Researchers have generally adopted time series analysis methods in an attempt to comprehend time series data. Such methods are based on the assumptions that one might discern regularity in the values of measured variables in an approximate sense, and that there are, many at times, patterns that persist over time.

Time series analysis is of great importance to the understanding of a range of economic, demographic, and astrophysical phenomena, and industrial processes. Traditionally, statistical methods have been used to analyse time series - economists model the state of an economy, social scientists analyse demographic data, and business managers model demand for
products, using such parametric methods. Models derived from the analysis of time series serve as crucial inputs to decision makers and are routinely used by private enterprises, government institutions and the academia.

In particular, financial and economic time series present an intellectual challenge coupled with monetary rewards and penalties for understanding (and predicting) future values of key variables from the past ones. Proponents of the Efficient Market Hypothesis (EMH) assert that price changes in financial markets are random and it is impossible to consistently outperform the market using publicly available information (Fama, 1965; Malkiel, 2003). However, Lo & MacKinlay (1999) argue that the EMH is an economically unrealizable idealization that 'is not... well-defined and empirically refutable', and that the Random Walk Hypothesis is not equivalent to the EMH. It has also been argued that an informationally efficient market is impossible (Grossman & Stiglitz, 1980), individuals exhibit bounded rationality (Simon, 1997), and market expectations may be irrational (Huberman & Regev, 2001). According to Lo (2005), the existence of active markets implies that profit opportunities must be present, and 'complex market dynamics, with cycles and trends and other phenomena' routinely occur in natural market ecologies. Alan Greenspan's famous 'irrational exuberance' speech (Federal Reserve Board, 1996) is another indicator that financial markets are not always efficient.

Increasingly, soft computing techniques (Zadeh, 1994) such as fuzzy systems, neural networks, genetic algorithms and hybrids, have been used to successfully model complex underlying relationships in nonlinear time series. Such models, referred to as universal approximators, are theoretically capable of uniformly approximating any real continuous function on a compact set to any degree of accuracy. Consider the case of fuzzy systems, defined in Wikipedia as 'techniques for reasoning under uncertainty', and based on fuzzy set theory developed by Zadeh (1973). Such systems have the advantage that models developed are characterised by linguistic interpretability, and rules generated can be understood, verified and extended (Chiu, 1997). Methods derived from fuzzy systems, such as the Takagi-Sugeno-Kang (TSK) fuzzy model (Takagi & Sugeno, 1985; Sugeno & Kang, 1988) have been used for analyzing time series, and over the past 20 years, sophisticated TSK hybrid methods, where fuzzy systems are combined with neural networks, genetic algorithms, both neural networks and genetic algorithms, or probabilistic fuzzy systems, have been employed to analyse time series data. In particular, many hybrid fuzzy models are designed to improve the forecast accuracy of fuzzy models by enhancing the system identification and optimisation techniques employed. It turns out that, if a sophisticated method is used without
understanding the underlying properties of the time series, then, ironically, for certain classes of time series, the forecasts are worse than for simpler methods.

Simple fuzzy systems, as well as complicated hybrids, have been used to analyse real-world time series, which are usually characterized by mean and variance changes, seasonality and other local behaviour. Such real-world time series are not only invariably nonlinear and nonstationary, but also incorporate significant distortions due to both 'knowing and unknowing misreporting' and 'dramatic changes in variance' (Granger, 1994). The presence of these characteristics in time series has led to considerable research, and debate, on the presumed ability of universal approximators to model nonstationary time series and the desirability of data pre-processing (Nelson et al, 1999; Zhang et al, 2001; Zhang & Qi, 2005). These studies have focused on investigating the ability of neural networks, another class of universal approximators, to model nonstationary time series, and the effect of data pre-processing on the forecast performance of neural networks. Similar studies on fuzzy systems have, to our knowledge, not been reported.

It has also been argued that most real-world processes, especially in financial markets, are made up of complex combinations of sub-processes or components, which operate at different frequencies or timescales (Gençay et al, 2002) and that observed patterns may not be present in fixed intervals over a (long) period of observation. Methods that involve decomposing a time series into 'its time-scale components and devising appropriate forecasting strategies for each' (Ramsey, 1999: 2604) have been developed for analysing real-world data. Typically, a well-informed modeller specifies the behaviour of each of the components: seasonal and business cycle components are specified together with trend components. Each component is then forecasted based on historical knowledge and experience. Much literature in financial and economic time series analysis requires the modeller to make a decision about the components, and various strategies have been used for modelling or filtering so-called components of time series. For example, variants of the classical decomposition model use different moving average filters to estimate the trend-cycle component (Makridakis et al, 1998). However, such decomposition methods are ad hoc, and are designed primarily for ease of computation rather than the statistical properties of the data (Mills, 2003). It has been argued that this rather informal approach has been formalised through the use of wavelet analysis (Ramsey, 1999) in the sense that the wavelet formalism decomposes the time series into component parts through a succession of approximations at different levels, such that trends, seasonalities, cycles and shocks can be discerned.
Methods based on the multiscale wavelet transform provide powerful analysis tools that decompose time series data into coefficients associated with time and a specific frequency band, whilst being unrestrained by the assumption of stationarity (Gencay et al., 2002: 1). Wavelets are deemed capable of isolating underlying low-frequency dynamics of nonstationary time series, and are robust to the presence of noise, seasonal patterns and variance change. Motivated by the capability of wavelets, hybrid models that use wavelets as a pre-processing tool for time series analysis have been developed. Wavelet analysis has been used for data filtering when employed in combination with neural networks and autoregressive (AR) models. In these studies, models built from wavelet-processed data consistently resulted in superior model performance (Aussem & Murtagh, 1997; Zhang et al., 2001; Soltani, 2002; Renaud et al., 2003; Murtagh et al., 2004; Renaud et al., 2005).

In this thesis, we extend the scope of studies on data pre-processing for soft computing methods to fuzzy systems (recall that studies on neural networks have been reported). Motivated by Ramsey’s (1999) assertion that traditional time series decomposition is formalized by using wavelets, we classify pre-processing methods into two categories: (i) conventional, ad hoc or ‘informal’ techniques, and (ii) ‘formal’, wavelet-based techniques. We investigate the single-step forecast performance of subtractive clustering TSK fuzzy models on nonstationary time series, and examine the effect of different informal ad hoc pre-processing strategies on the performance of the model. We then propose a formal wavelet-based approach for automatic pre-processing of time series prior to the application of fuzzy models. We argue that, whilst wavelet-based processing is generally beneficial, fuzzy models built from wavelet-processed data may underperform compared to models trained on raw data i.e. the performance of the wavelet-based method depends on the properties of the time series under analysis. In particular, our study of subtractive clustering TSK fuzzy models of nonstationary time series indicates that time series that exhibit change in variance require pre-processing, and wavelet-based pre-processing is a ‘natural’, parameter-free method for decomposing such time series. However, where the variance structure of a time series is homogeneous, wavelet-based pre-processing leads to worse results compared to an equivalent analysis carried out using raw data. This is indicative of the bias/variance dilemma (Geman et al., 1992) where the use of a complex, wavelet-based method to analyse data with a simple structure results in models that exhibit poor out-of-sample generalisation.

We present a framework where time series data could be pre-processed, given that a decision to use informal or formal methods has been made (Figure 1.1). While using an informal method, a well-informed modeller decides either to preserve or eliminate time series
components. Following the decision of not preserving the components, our framework provides well-known tests for investigating the properties of the time series and recommending appropriate methods for pre-processing. For the case where the formal pre-processing method is preferred, our motivation is to create multiresolution-based techniques, because only with such techniques can we deal with local as well as global phenomena, particularly variance breaks and related inhomogeneity in the series.

It is important to establish the suitability of wavelet pre-processing and to look beyond conventional wavelet analysis, particularly for the process of prediction. An automatic method for detecting variance breaks in time series is used as an indicator as to whether or not wavelet-based pre-processing is required. Once the time series is diagnosed, we are ready to extract the patterns using wavelets, if the framework suggests this (Figure 1.2). We have used the maximal overlap discrete wavelet transform (MODWT) for decomposing a time series, and employed some of the most commonly used and freely available packages for the analysis.
In order to evaluate our framework, we have utilised well-known economic time series data, comprising monthly data, which have been used in the evaluation of forecasting methods in the soft computing literature. Monthly series are used since they exhibit stronger seasonal patterns than quarterly series, and are characterized, in varying degrees, by trend and seasonal patterns, as well as discontinuities. We have also examined the behaviour of fuzzy models generated from synthetic time series in order to investigate the effects of pre-processing on the forecast performance of such models. The complexity of fuzzy models, in terms of the number of rule clusters automatically generated using differently processed time series, has also been investigated.

![Flowchart Diagram](image)

**Figure 1.2.** Wavelet-based pre-processing scheme with diagnosis phase.

### 1.2 Contributions of the Thesis

Two areas of investigation are addressed in this thesis: first, a study of the effects of pre-processing is carried out on subtractive clustering fuzzy models; second, the use of a wavelet-based framework is proposed for data pre-processing prior to the application of a fuzzy model. Specifically, the contributions can be summarised as follows:

1. **We extend previous work on the effects of data pre-processing on the forecast performance of neural networks, another class of soft computing models, to subtractive clustering fuzzy systems.**
2. **We propose a systematic method for selecting traditional informal methods for data pre-processing for fuzzy models.**
3. **We present a fuzzy-wavelet framework for automatic time series analysis, using formalised wavelet-based data pre-processing methods.**
iv) We present an intelligent approach for testing the suitability of wavelets for fuzzy TSK models.

Recall that the EMH mainly applies to (noisy, high-frequency) financial data. We note that, although some of the concepts described in our research are applicable to financial time series, this thesis mainly deals with economic time series, which are aggregated and relatively noise-free, there is no direct contribution to the debate on the EMH.

1.3 Structure of the Thesis

This thesis is organized into five chapters. Following the general introduction in this chapter, Chapter 2 presents a comprehensive review of the literature relevant to the research subject of this thesis, beginning with a description of time series components and characteristics, and conventional and soft computing approaches to time series analysis. This is followed by a detailed discussion of fuzzy models used for time series analysis, including a critique of fuzzy models, and a description of the multiscale wavelet transform as a pre-processing tool.

In Chapter 3, details of the methods undertaken to address the research questions are presented. The chapter starts with a discussion of informal data pre-processing techniques, and the limitations inherent in such methods. Subsequently, the chapter discusses tests for determining the suitability of data pre-processing in both formal and informal frameworks, and describes the proposed method, which features wavelet-based times series pre-processing.

Chapter 4 describes the criteria used for evaluating the proposed method, and then presents a discussion of the experimentation results achieved using the proposed framework for informal and formal pre-processing of real-world time series with characteristics of interest – trend, seasonality, and discontinuities.

In chapter 5, a general assessment of the outcome of the research vis-à-vis the research objectives set forth in Chapter 1, is presented, followed by conclusions and suggested directions for future work.

1.4 Publications

The author initiated and made significant contributions to the papers listed below, under the supervision of, and in close collaboration with, the supervisor. The papers are as follows:


Motivation and Literature Review

Real world statistical material often takes the form of a sequence of data, indexed by time. Such data are referred to as time series and occur in several areas of human endeavour: share prices in financial markets, astrophysical phenomena like sunspots, sales figures for a business, demographic information of a geographical entity, amongst others. Time series measurements may be continuous - made continuously in time - or discrete i.e. at specific, usually equally spaced intervals. The essence of time series analysis is that there are patterns of repeated behaviour that can be identified and modelled. The repetition, either of smooth or turbulent behaviour, is essential for generalization. Conventional statistical methods, soft computing methods, and hybrids have been used to characterise repeating patterns in time series data. In particular, fuzzy models represent time series in terms of fuzzy rules. Such rule-based methods are considered to be advantageous because they provide not only an insight into the reasoning process used to generate results but also the interpretation of results obtained from such methods. Fuzzy rules provide a potent framework for mining and explaining input/output data behaviour, and fuzzy systems enable qualitative modelling with the use of approximate information and uncertainty.

A time series can, in principle, be used to generate a rule set of fuzzy rules, each rule reflecting the behaviour in a given proximity. Consider the well-analyzed time series of daily IBM stock prices from May 17, 1961 to November 2, 1962 (Box & Jenkins, 1970) shown below (Figure 2.1).
The value of the stock price may be described using so-called linguistic variables:

- **low** if \( p_i \leq 350 \)
- **medium** if \( 300 \leq p_i \leq 450 \)
- **high** if \( 400 \leq p_i \leq 550 \)
- **very high** if \( p_i \geq 500 \)

where \( p_i \) is the price at time \( t \). Consider three ‘episodes’ in the IBM series over a five day period and the corresponding linguistic descriptions (Table 2.1).

<table>
<thead>
<tr>
<th>Episode</th>
<th>Period</th>
<th>( p_{i-5} )</th>
<th>( p_{i-4} )</th>
<th>( p_{i-3} )</th>
<th>( p_{i-2} )</th>
<th>( p_i )</th>
<th>Period</th>
<th>( p_{i+1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0-5</td>
<td>460 high</td>
<td>457 high</td>
<td>452 high</td>
<td>459 high</td>
<td>462 high</td>
<td>459 high</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>95-100</td>
<td>541 high*</td>
<td>547 high*</td>
<td>553 very high</td>
<td>559 very high</td>
<td>557 very</td>
<td>557 very</td>
<td>101</td>
</tr>
<tr>
<td>3</td>
<td>265-270</td>
<td>374 med</td>
<td>359 med</td>
<td>335 low*</td>
<td>323 low*</td>
<td>306 low*</td>
<td>333 low*</td>
<td>271</td>
</tr>
</tbody>
</table>

From the tabulation above, the following rules can be inferred:

1. if \( p_{i-5} \) is high and \( p_{i-4} \) is high and ... and \( p_i \) is high
   \[ \text{then } p_{i+1} \text{ is high} \]

2. if \( p_{i-5} \) is high and \( p_{i-4} \) is high and ... and \( p_i \) is very high
   \[ \text{then } p_{i+1} \text{ is very high} \]

3. if \( p_{i-5} \) is medium and \( p_{i-4} \) is medium and ... and \( p_i \) is low
   \[ \text{then } p_{i+1} \text{ is low} \]

The asterisked values (Table 2.1) capture the fuzziness of the description in the overlapping high-to-very-high and medium-to-low regions. This example illustrates the usefulness of fuzzy systems in providing qualitative, human-like characterization of numerical data.

Time series analysis has been carried out using fuzzy systems, due to the approximation capability and linguistic interpretability of such methods. Typically, fuzzy systems are trained on raw or return data, and the approximation accuracy is improved only by developing sophisticated structure identification and parameter optimisation methods. These methods combine fuzzy systems with neural networks, genetic algorithms, or both. However, a review of fuzzy models used for time series analysis indicates that the use of sophisticated methods does not necessarily result in significant accuracy improvements. We argue that an alternative approach to improving forecast accuracy, using data pre-processing, is beneficial.
The remainder of this chapter is structured as follows. In the next section, basic notions of time series, including time series components and nonstationarity in time series, are discussed. This is followed by an overview of conventional parametric methods used for time series analysis, and a description of soft computing models, in particular, fuzzy models for time series analysis. A critique of state-of-the-art fuzzy models is then provided, and an alternative approach for improving forecast performance, based on reducing data complexity via wavelet-based pre-processing, is discussed. Finally, a summary of the chapter is provided.

2.1 Time Series: Basic Notions

2.1.1. Components of Time Series

A time series can be represented as \( \{X_t: t = 1, \ldots, N\} \) where \( t \) is the time index and \( N \) is the total number of observations. In general, observations of time series are related i.e. autocorrelated. This dependence results in patterns that are useful in the analysis of such data. Time series are deemed to comprise different patterns, or components, and are functions of these components:

\[ X_t = f(T_t, S_t, C_t, I_t) \]

where \( T_t, S_t, C_t \) and \( I_t \) respectively represent the trend, seasonal, cyclical and irregular components. There are two general types of models of time series based on the decomposition approach: the additive and multiplicative models (Makridakis et al, 1998). Mathematically, the additive model is represented as:

\[ X_t = T_t + C_t + S_t + I_t \]

and the multiplicative model is defined by:

\[ X_t = T_t \times C_t \times S_t \times I_t \]

Box 2.1 illustrates how different components contribute to a time series in an additive model.

(i) The trend component

The trend component represents the long-term evolution – perhaps underlying growth or decline – in a time series. In the simulated series (Box 2.1), the trend component is simply a straight line or a linear trend. In real-world data, trends may be caused by various factors,
including economic, weather or demographic changes, and many economic and financial variables exhibit trend-like behaviour.

A time series with additive components can be made up of trend, seasonal/cyclical and irregular components, where $t$ is the time index, $T_t = 0.3t$ is the trend component, the irregular component, $I_t$ is a Gaussian random variable, and the seasonal/cyclical component $S_t$ is defined as:

$$S_t = \sum_{i=1}^{4} 3 \sin\left(\frac{2\pi}{P_i} t\right) ;$$

The four periodic components in $S_t$ are $P_1 = 2$, $P_2 = 4$, $P_3 = 8$, $P_4 = 16$.

Box 2.1. A simulated time series and its additive components

For example, consider the movements in the closing value of the UK stock market index, FTSE 100 (Figure 2.2), which shows a growth trend in the value of the index. The trend can be fitted with a linear function as an indication of the long-term movement of the index. However, the definition of what constitutes a trend is not exact, except in the context of a model (Franses, 1998). According to Chatfield (2004), trend can be loosely defined as 'a long-term change in the mean level'. Using this definition, the first-order polynomial fitted to the time series is an indication of the trend, although a piecewise linear trend with two segments, corresponding to the two regimes (with transition point sometime in April 2006) can also be fitted.

Granger (1966) argues that what is considered as the trend in a short series is not necessarily considered as the trend in much longer time series, and suggests the use of the 'trend in
mean”, which defines a trend as all components whose wavelengths are at least equal to the length of the series. Also, trends are often not considered in isolation, and so-called trend-cycle components, which comprise both trend and cyclical components, are obtained by convolving time series with moving average (MA) filters (Makridakis et al, 1998).

Figure 2.2. Closing value of the FTSE 100 index from Nov. 2005 – Oct. 2006 fitted with linear trend line.

Trends can be broadly classified as being deterministic or stochastic (Virili & Freisleben, 2000) and time series with such trends are described as being trend stationary or difference stationary, respectively (Mills, 2003). Time series where stationarity can be produced by differencing, i.e. difference-stationary series, are regarded as having stochastic trend. Conversely, if the residuals after fitting a deterministic trend are stationary i.e. trend stationary, then the trend is considered deterministic (Chatfield, 2004). Traditional methods for modelling deterministic trends include the use of linear and piecewise linear fit functions, and nonlinear n-order polynomial curve fitting. However, the choice of models for trends is not trivial. Mills (2003) argues that the use of linear functions are ad hoc, the use of segmented or piecewise linear trends requires a priori choice of the terminal points of ‘regimes’, and high order polynomials may lead to overfitting. Note the use of the term ‘ad hoc’ by Mills – it is not only the ad hoc choice of a function that is of concern, but the predication that trends exist.

(ii) Seasonal and cyclical components

Time series in which similar changes in the observations are of a (largely) fixed period are referred to as being characterised by seasonality. Seasonality is often observed in economic time series, where monthly or quarterly observations reveal patterns that repeat year after year. In particular, seasonality is indicated when observations in some time periods have strikingly different patterns compared to observations from other periods (Franses, 1998). The annual timing, direction, and magnitude of seasonal effects are reasonably consistent (US Census Bureau, 2006). For example, the unadjusted women’s clothing stores sales (UWCSS) series (Figure 2.3) exhibits distinct seasonal patterns. The tendency for clothing sales to rise
around Christmas, a seasonal pattern, is clearly indicated by the peaks observed in the 12th month and its integer multiples.

Observed seasonal components often dominate a time series, obscuring the low-frequency underlying dynamics of the series. Seasonally adjusted data is used to unmask underlying non-seasonal features of the data. The seasonally adjusted UWCSS time series indicates that the non-seasonal patterns display a mild increase in the first year, and a downward trend in subsequent years.

![Figure 2.3. Women's clothing sales for January 1992 – December 1996 showing unadjusted (blue) and seasonally adjusted (red) data.](image)

The length of the seasonal pattern observed depends on the data being analysed: with economic time series, periods of interest are typically months or quarters, while for high frequency financial time series, seasonality at daily periods and higher moments would be observed. Similar to the trend component, seasonality in time series are also classified as either being stochastic or deterministic (Chatfield, 2004), although Pierce (1978) asserts that both stochastic and deterministic components may be present in the same time series. Seasonal patterns are deterministic if they can be described using functions of time, while stochastic seasonality is present if seasonal differencing is needed to attain stationarity.

Unlike seasonal components, which are considered to have fixed periods, wavelike fluctuations without a fixed period or consistent pattern are regarded as being cyclical. Whilst seasonal patterns are mainly due to the weather and artificial events such as holidays, cyclical components are usually indicative of changes in economic expansions and contractions i.e. business cycles, with rates of changes varying in different periods. If the length of a time series is short relative to the length of the cycle present in the data, the cyclical component will be observed as a trend (Granger, 1966). In general, seasonal patterns have a maximum length of one year, while repeating patterns that have a length longer than one year are referred to as cycles (Makridakis et al, 1998).
(iii) *Irregular components*

The irregular (residual or error) component of a time series describes the variability in the time series after the removal of other components. Such components are considered to have unpredictable timing, impact, and duration (US Census Bureau, 2006). Consider the women’s clothing stores series described earlier. Irregular components are obtained by (i) *first differences* of the seasonally adjusted data, assuming difference stationarity, and (ii) subtracting a linear trend component from the seasonally adjusted data, assuming trend stationarity (Figure 2.4). It appears that, in this case, the difference stationary model is appropriate, since the residual obtained from this process appears to be stationary, unlike the residual from the trend stationary approach.

![Figure 2.4. Irregular component of women’s clothing sales obtained by assuming (i) a difference stationary trend (blue) and (ii) a trend stationary model (red).](image)

**2.1.2. Nonstationarity in the Mean and Variance**

Generally, a time series is considered stationary if there is no systematic change in either the mean or the variance, and if strictly periodic fluctuations are not present (Chatfield, 2004). If the data does not fluctuate around a constant mean, and has no long-run mean to which it returns, the series is nonstationary in the mean. For example, consider the time series of Gaussian random variables with a zero mean and unit variance (Figure 2.5). The mean of the series is constant and the values of the series fluctuate about the mean, and with approximately constant magnitude. This series is stationary in both the mean and variance.

![Figure 2.5. Time series data that is stationary in the mean and variance.](image)
Nonstationarity in the mean can be due to two principal factors. First, nonstationarity may be due to a (long-term) trend. This can be visualised by adding a linear trend to the Gaussian random variable time series (Figure 2.6a). Second, nonstationarity in the mean can be caused by the presence of additive seasonal patterns.

Figure 2.6. Time series data that is nonstationary in the (a) mean and (b) variance.

Difference stationary time series are made stationary by the application of differencing, while trend stationary time series are made stationary by fitting a linear trend to the series, as earlier discussed. Statistical unit root tests, such as the Dickey-Fuller and Phillips-Perron tests, have been developed to distinguish between trend and difference stationary time series. It has however been argued that unit root tests have poor power, especially for small samples (Levin et al., 2002). The type of detrending technique used on a time series is important, since the use of improper techniques may result in poor forecast performance (Chatfield, 2004).

If the variance is not constant with time, the series exhibits nonstationarity in the variance. Nonstationarity in the variance of time series is typically caused by multiplicative seasonality, where the seasonal effect appears to increase with the mean. In the example, the alteration of the fluctuation around the mean results in a series that is nonstationary in the variance (Figure 2.6b). Conventionally, data transformations are employed in order to stabilize the variance, make multiplicative seasonal effects additive, and ensure that data is normally distributed (Chatfield, 2004). The Box-Cox family of power transformations is widely used for data transformation:

$$X'_i = \begin{cases} 
(X_i^\lambda - 1)/\lambda & \lambda \neq 0 \\
\log(X_i) & \lambda = 0
\end{cases}$$
where $X_t$ is the original data, $X_t'$ is the transformed series, and $\lambda$ is the transformation parameter, which is estimated using the value of $\lambda$ that maximizes the log of the likelihood function.

### 2.2 Time Series Models

The analysis of time series focuses on three basic goals: forecasting (or predicting) near-term progressions, modelling long-term behaviour and characterising underlying properties (Gershenfeld and Weigend, 1994). Interest in such analysis is wide ranging, dealing with both linear and non-linear dependence of the response variables on a number of parameters. A key motivation of research in time series analysis is to test the hypothesis that complex, potentially causal relationships exist between various elements of a time series. Conventional, model-based parametric methods express these causal relationships through a variety of ways. The most popular is the autoregressive model where there is an assumption that the causality connects the value of the series at time $t$ to its $p$ previous values. On the other hand, soft computing techniques such as fuzzy systems, genetic algorithms, neural networks and hybrids presumably make no assumptions about the structure of the data. These methods are referred to as 'universal approximators' that provide non-linear mapping of complex functions.

#### 2.2.1 An Overview of Conventional Approaches

Conventional statistical models for time series analysis can be classified into linear models and non-linear changing variance methods. Linear methods comprise autoregressive (AR), moving average (MA), and hybrid AR and MA (ARMA) models. Such models summarise the knowledge in a time series into a set of parameters, which, it is assumed, simulate the data, or some of its interesting structural properties. Linear models also assume that the underlying data generation process is time invariant, i.e. the process does not change in time. The assumption that time series are stable over time necessitates the use of stationary time series for linear models.

Autoregressive (AR) models represent the value of a time series $X_t$ as a combination of the random error component $\varepsilon_t$ and a linear combination of previous observations:

$$X_t = \phi_1 X_{t-1} + ... + \phi_p X_{t-p} + \varepsilon_t$$
where \( p \) is the order of the autoregressive process, \( \phi_i \)'s are autoregressive coefficients, and \( \varepsilon_i \) is a Gaussian random variable with mean zero and variance \( \sigma^2 \). AR models assume that the time series being analysed is stationary.

In contrast to AR models, where only random shocks at time \( t \) are assumed to contribute to the value of \( X_t \), moving average (MA) models assume that past random shocks propagate to the current value of \( X_t \). MA models represent time series as a linear combination of successive random shocks:

\[
X_t = \theta_0 \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q}
\]

where \( \varepsilon_t \) is a white noise process with zero mean and variance \( \sigma^2 \); \( \theta \) are parameters of the model, and \( q \) is the order of the MA process. The orders of simple autoregressive (\( p \)) and moving average (\( q \)) models are typically determined by examining the autocorrelation function (ACF) and partial autocorrelation function (PACF) plots of the data under analysis, and general rules have been devised for the identification of these models (Makridakis et al., 1998).

Box and Jenkins (1970) introduced a more general class of models incorporating both AR and MA models i.e. mixed ARMA models. A mixed ARMA model with \( p \) AR terms and \( q \) MA terms is said to be of order \( (p,q) \) or ARMA(\( p,q \)), and is defined by:

\[
X_t = \phi_1 X_{t-1} + \ldots + \phi_p X_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q}
\]

The ARMA model assumes a stationary time series. In order to take into account the fact that, in practice, most time series are nonstationary in the mean, a difference operator was introduced as part of the ARMA model to adjust the mean. The modified model is called an integrated ARMA or ARIMA model since, to generate a model for nonstationary data, the stationary model fitted to the differenced data has to be summed or integrated (Chatfield, 2004). The differenced series, \( X'_t \), is defined as

\[
X'_t = \nabla^d X_t
\]

where \( d \) is the number of differencing operations carried out to make \( X_t \) stationary. The resulting ARIMA model is given by:

\[
X'_t = \phi_1 X'_{t-1} + \ldots + \phi_p X'_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q}
\]
The ARIMA model is of the general form ARIMA \((p,d,q)\). Unlike separate AR or MA models, patterns of the ACF and PACF for ARIMA cannot be easily defined. Consequently, model identification is carried out in an iterative fashion, with an initial model identification stage, and subsequent model estimation and diagnostics stage. Typically, the accuracy of the model developed depends on the expertise of the analyst, and the availability of information about the data generating process.

In ARIMA models, the relationship between \(X_t\), past values \(X_{t-p}\) and error terms \(\epsilon_t\) is assumed to be linear. If the dependence is nonlinear, specifically if the variance of a time series increases with time i.e. the series is \textit{heteroskedastic}, it is modelled by a class of autoregressive models known as the \textit{autoregressive conditional heteroskedastic} (ARCH) models (Engle, 1982). The most commonly used variant of the ARCH model is the generalised ARCH (GARCH) model, introduced by Bollerslev (1986).

The conventional methods so far described are parametric. In the next section, soft computing approaches to time series analysis are considered.

\textbf{2.2.2 Soft Computing Models: Fuzzy Inference Systems}

Conventional methods, described in the previous sections, are well understood and commonly used. However, time series are invariably nonstationary, and structural assumptions made on the data generating process are difficult to verify (Moorthy \textit{et al}, 1998), making traditional models unsuitable for ‘even moderately complicated systems’ (Gershenfeld and Weigend, 1994). Also, real-world data may be a superposition of many processes exhibiting diverse dynamics. Increasingly, soft computing techniques such as fuzzy systems, neural networks, genetic algorithms and hybrids, have been used to model complex underlying relationships in nonlinear time series. Such techniques, referred to as \textit{universal approximators} (Kosko, 1992; Wang, 1992; Ying, 1998), are theoretically capable of uniformly approximating any real continuous function on a compact set to any degree of accuracy. Unlike conventional methods, soft computing models like neural networks, it has been argued, are ‘nonparametric’ and learn without making assumptions about the data generating process (Berardi and Zhang, 2003). However, Bishop (1995) asserts that soft computing methods do make assumptions, and can only be described as being ‘semi-parametric’.

In particular, fuzzy systems are used due to linguistic interpretability of rules generated by such methods. Fuzzy Inference Systems (FIS) are described as universal approximators that can be used to model non-linear relationships between inputs and outputs. The operation of a
FIS typically depends on the execution of four major tasks: fuzzification, inference, composition, and defuzzification (Table 2.2). The identification of a fuzzy system has close parallels with identification issues encountered in conventional systems. There are two factors that are relevant here: structure identification and parameter identification (Takagi & Sugeno, 1985). Structure identification involves selecting variables, allocating membership functions and inducing rules while parameter identification entails tuning membership functions and optimising the rule base (Emami et al., 1998).

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuzzification</td>
<td>Definition of fuzzy sets; determination of the degree of membership of crisp inputs</td>
</tr>
<tr>
<td>Inference</td>
<td>Evaluation of fuzzy rules</td>
</tr>
<tr>
<td>Composition</td>
<td>Aggregation of rule outputs</td>
</tr>
<tr>
<td>Defuzzification</td>
<td>Computation of crisp output</td>
</tr>
</tbody>
</table>

The performance of each fuzzy model on a given set of data is dependent on the specific combination of system identification and optimisation techniques employed. Two rule evaluation methods, differing in the form of the rule consequent, are generally applied in fuzzy systems: the Mamdani (Mamdani & Assilian, 1975) and Takagi-Sugeno-Kang, or TSK (Takagi & Sugeno, 1985; Sugeno & Kang, 1998), inference methods. In this thesis, the TSK method is employed, due to its computational efficiency (Negnevitsky, 2005).

### 2.3 Fuzzy Models for Time Series Analysis

Methods based on the fuzzy inference system, or fuzzy systems, and its hybrids have been used in the analysis and modelling of time serial data in a number of different application areas (Table 2.3). Modelling methods using fuzzy set theory are broadly classified into those using complex rule generation mechanisms and ad hoc data-driven models for automatic rule generation (Casillas et al., 2002). Complex rule generation mechanisms employ hybrid methods, including neuro-fuzzy (NF), genetic fuzzy (GF), genetic-neuro-fuzzy (GNF) and probabilistic fuzzy methods. Conversely, ad hoc data-driven models utilize data covering criteria in example sets.
Neuro-fuzzy models incorporate strengths of neural networks, such as learning and generalisation capability, and strengths of fuzzy systems, such as qualitative reasoning and uncertainty modelling ability. The Adaptive Network-based Fuzzy Inference System (ANFIS) proposed by Jang (1993) is one of the most commonly used neuro-fuzzy methods, with over 1,400 citations in Google Scholar as at November 2006. The ANFIS is a neural network that models TSK-type fuzzy inference systems and it comprises five layers, each layer being functionally equivalent to a fuzzy inference system. Other neuro-fuzzy models include the subsethood-product fuzzy neural inference system, SuPFuNIS (Paul & Kumar, 2002), the dynamic evolving neural-fuzzy inference system, DENFIS, (Kasabov & Song, 2002), and hierarchical neuro-fuzzy quadtree (HFNQ) models (de Souza et al, 2002). (see Mitra and Hayashi, 2000 for a review of the neuro-fuzzy approach).

Table 2.3 Exemplar application areas of fuzzy models and hybrids for time series analysis

<table>
<thead>
<tr>
<th>Application area</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Financial time series</td>
<td>Analysis of market index (Van den Berg et al, 2004); real-time forecasting of stock prices (Wang, 2003); forecasting exchange rate (Tseng et al, 2001)</td>
</tr>
<tr>
<td>Chaotic functions</td>
<td>Prediction of Mackey-Glass chaotic function (Tsekouras et al, 2005; Kasabov &amp; Song, 2002; Kasabov, 2001; Mendel, 2001; Rojas et al, 2001)</td>
</tr>
<tr>
<td>Control system</td>
<td>Electricity load forecasting (Lotti, 2001; Weizenegger, 2001)</td>
</tr>
<tr>
<td>Transportation</td>
<td>Traffic flow analysis (Chiu, 1997)</td>
</tr>
<tr>
<td>Sales</td>
<td>Forecasting (Kuo, 2001; Singh, 1998)</td>
</tr>
</tbody>
</table>

The genetic fuzzy predictor ensemble (GFPE) proposed by Kim and Kim (1997) is an exemplar genetic fuzzy (GF) system. In this model, the initial membership functions of a fuzzy system are tuned using genetic algorithms, in order to generate an optimised fuzzy rule base. Other GF methods use sophisticated genetic algorithms, such as multidimensional and multideme genetic algorithms (Rojas et al, 2001) and multi-objective hierarchical genetic algorithms, MOHGA (Wang et al, 2005), to construct fuzzy systems. A review of the genetic fuzzy approach to modelling is provided by Cordón et al (2004).

Genetic-neuro-fuzzy (GNF) hybrids like the genetic fuzzy rule extractor, GEFREX (Russo, 2000) have also been reported. Unlike the neuro-fuzzy method, which uses neural networks to provide learning capability to fuzzy systems, GEFREX uses a hybrid approach to fuzzy supervised learning, based on a genetic-neuro learning algorithm. Other GNF hybrids include evolving fuzzy neural networks, EfuNNs (Kasabov, 2001), the hybrid evolutionary neuro-fuzzy system, HENFS (Li et al, 2006), and the self-adaptive neural fuzzy network with group-
based symbiotic evolution (SANFN-GSE) method (Lin & Yu, 2006). Finally, a hybrid approach involving the use of both probabilistic and fuzzy systems frameworks was proposed by van den Berg et al. (2004). The probabilistic fuzzy system (PFS) is unique in that, unlike other hybrids, which use a combination of soft computing methods, the PFS combines the strengths of uncertainty modelling present in both probabilistic and fuzzy frameworks to model financial data.

Recall that complex rule generation mechanisms, and ad hoc data-driven models are identified as two broad classes of fuzzy models for automatic rule generation. This classification is not strict, and ad hoc data-driven models, having advantages of simplicity, speed and high performance, often serve as preliminary models that are subsequently refined using other more complex methods (Casillas et al., 2002). In the following, we present a description of ad hoc data-driven models, based on the data partitioning scheme used. In particular, we describe two of the general partitioning schemes discussed in the rule induction literature that have a significant bearing on time series analysis: grid partitioning and scatter partitioning (Jang et al., 1997; Guillaume, 2001).

### 2.3.1 Grid Partitioning

In grid partitioning, a small number of fuzzy sets are usually defined for all variables, and are used in all the induced rules (Guillaume, 2001). There are two general methods: i) models where fuzzy sets are predetermined, often defined by domain experts, and have qualitative meanings, making generated rules suited for linguistic interpretation; ii) models where fuzzy sets are dynamically generated from training data. In the following, we describe grid partitioning with pre-specified fuzzy sets (section 2.4.1.1) and dynamically generated fuzzy sets (section 2.4.1.2).

#### 2.3.1.1 Models with Pre-specified Fuzzy Sets

This method induces rules that consist of all possible combinations of defined fuzzy sets (Ishibuchi et al., 1994; Nozaki et al., 1997). Here, a non-linear system is approximated by specifying covering n-input and single-output space using fuzzy rules of the form:

\[
\text{Rule } R_j: \text{ If } x_i \text{ is } A_{ij} \text{ and } \ldots \text{ and } x_n \text{ is } A_{nj} \text{ then } y \text{ is } w_i \quad j = 1, \ldots, N \text{ and } i = 1, \ldots, n
\]

where \( R_j \) is the \( j \)-th rule of \( N \) fuzzy rules, \( x_i \) is the \( i \)-th input variable, \( A_{ij} \) is the linguistic value defined by a fuzzy set, \( y \) is the output variable, and \( w_i \) is a real number. This method is based on the zero-order TSK model. An \( n \)-dimensional input space \([0,1]^n\) is evenly partitioned into
Chapter 2

$N$ fuzzy subspaces ($N = K^n$, where $K$ is the number of pre-specified fuzzy sets in each of the $n$ dimensions) using a simple fuzzy grid and triangular membership functions (Figure 2.7).

A learning method, the gradient descent method, is then used to select the best model, based on minimising the total error. It can be argued that this method is not efficient since, depending on input space data distribution and partitioning, many rules will be generated i.e. it suffers from the curse of dimensionality problem, and some rules may never be activated. Also, the number of fuzzy sets is pre-specified. This may lead to overfitting and loss of generality if too many fuzzy sets are used, and loss of accuracy where too few fuzzy sets are defined.

Another method, the so-called Wang-Mendel (WM) method (Wang and Mendel, 1992; Wang, 2003) uses the number of training pairs to limit the number of rules generated. Consider an $n$-input single output process, given a set of input-output data pairs:

$$(x_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(1)}; y^{(1)}), (x_1^{(2)}, x_2^{(2)}, \ldots, x_n^{(2)}; y^{(2)}), \ldots \quad i = 1, \ldots, n$$

where $x_i$ are inputs and $y$ is the output, the method provides a mapping $f$: $(x_1, x_2, \ldots, x_n) \rightarrow y$.

Each input and output variable is divided into 'domain intervals' that define the region of occurrence of the variable. A user specified number of fuzzy sets with triangular membership functions is then assigned to each region. For interpretability, the fuzzy sets may have linguistic labels like small ($S_1, S_2, S_3$), centre (C), and big ($B_1, B_2, B_3$), as shown in Figure 2.8.

Membership functions are assigned to individual variables by mapping from the time series to the pre-specified fuzzy sets. Taking $(x_1, x_2)$ as inputs and $(x_3)$ as the output in Figure 2.8, an exemplar rule relating the variables is of the form:

if $x_1$ is $B_1$ or $B_2$ and $x_2$ is $S_1$ or $S_2$ then $x_3$ is $S_2$ or $S_3$
i.e. $x_1$, $x_2$, and $x_3$ respectively belong to fuzzy sets $B_1$ and $B_2$; $S_1$ and $S_2$; $S_2$ and $S_3$. Subsequently, each variable is allocated to the fuzzy set in which it has the maximum membership function:

\[
\text{if } x_1 \text{ is } B_1 \text{ and } x_2 \text{ is } S_2 \text{ then } x_3 \text{ is } S_2
\]

There may be rules that have similar antecedents and different consequents. This is addressed by a conflict resolution method where each rule is assigned a degree, $D$, a product of the membership functions of its antecedents and consequents:

\[
D_{\text{Rule}} = \mu_{A_1}(x_1) \mu_{A_2}(x_2) \cdots \mu_{A_m}(x_m) \mu_B(y)
\]

The rule with the highest degree, in each set of conflicting rules, is chosen. Selected rules are then used to populate the fuzzy rule base. This model is one of the most widely cited ad hoc data-driven methods, with over 650 Google Scholar citations as at November 2006, and several improvements have been proposed to deal with identified limitations. The most comprehensive review of the technique was carried out by one of the original authors, Wang (2003). Relevant modifications proposed include flexibility in the choice of membership functions, rule extrapolation to regions not covered by training data, model validation, input selection, and model refinement.

### 2.3.1.2 Models with Dynamically Generated Fuzzy Sets

The models described in the previous section are based on user specified fuzzy sets. In order to address limitations related to predetermined fuzzy sets, such as the curse of dimensionality due to rule explosion, and overfitting, methods that use iterative *partition refinement* have been developed. Here, a partition covering the input space with two fuzzy sets – centred at the maximum and minimum value of the input data set - is initially specified and error indices
associated with each fuzzy region and input variable are defined. Model refinement is achieved by adding fuzzy sets to the input subspace responsible for the greatest error. The iteration is stopped after the error falls below a given threshold or when it reaches a minimum. Here, although fuzzy sets are dynamically chosen and not arbitrarily specified, all possible rule combinations are still implemented, like the previous methods (section 2.3.1.1). Also, model refinement is limited to the input space region.

Rojas et al (2000) proposed an improvement, referred to as a 'self-organised fuzzy system', which involves model refinement not only in the input space region, but also at the rule level. The technique is a three-phase process. In the first phase, a simple system having membership functions and rules is initialised. In the next phase, several structures are modelled by dynamically altering the fuzzy sets defined for some input variables and evaluating system output. In a particular input subspace, there may be input-output vectors with significantly different outputs, creating conflicts. To determine the consequent part of rules, a controversy index (CI) is defined. The CI provides a measure of the difference between the observed outputs of data points that fire a specific rule and the rule conclusion provided by the method. The lower the CI value, the better the match between observed and estimated rules.

The CI is extended to membership functions in order to determine particular membership functions responsible for high CI in a region. This is achieved by using another index, the sum of controversies associated with a membership function (SCMF). A normalised SCMF is computed to facilitate comparison and more fuzzy sets are assigned to input subspaces with high controversy values. In the third and final phase, the best structure, which provides a compromise between desired accuracy and rule set complexity, is selected. Selection is carried out using another index derived from the mean square error of the approximation and the number of rules in the system.

### 2.3.2 Scatter Partitioning: Subtractive Clustering

Scatter partitioning, or clustering, aims at partitioning data into quasi-homogenous groups with intra-group data similarity greater than inter-group similarity. This approach attempts to obtain an approximation of the fuzzy model without making assumptions about the structure of the data (Jang et al, 1997). Clustering is used in fuzzy modelling for data compression and model construction. In order to divide data into groups, similarity metrics are used to evaluate the homogeneity of normalized input vectors. Comparable input-output data pairs in the training set are assembled into groups or clusters. After data partitioning, one rule is associated with each data cluster, usually leading to rules scattered in the input space at
locations with sufficient concentration of data. This results in a greatly reduced number of rules, in contrast to grid-partitioned models. Also, as opposed to models using grid partitioning, fuzzy sets are not shared by all the rules. Off-line clustering algorithms used for fuzzy modelling include the fuzzy C-means (FCM) clustering (Bezdek, 1981), the mountain clustering method (Yager & Filev, 1994) and the subtractive clustering technique (Chiu, 1997).

**FCM algorithm** partitions a time series vector $x_i$ into $g$ fuzzy groups, and finds a cluster centre in each group that minimises a cost function, typically the Euclidean distance (Vernieuwe et al., 2006). In this scheme, each data point does not belong exclusively to one cluster, but may belong to several clusters with different degrees of membership. The FCM algorithm requires the specification of the number of clusters and initial cluster centres, and the performance of the method depends on this specification (Jang et al., 1997). Various methods have been proposed to, amongst others, improve cost metric selection (Bouchachia & Pedrycz, 2006) and sensitivity to noise and outliers (Leski, 2003) for the FCM method.

The **mountain clustering** method addresses the specification of initial clusters and their location, both limitations of the FCM method. In the mountain clustering method, a grid is formed in the data space, and each grid point is deemed a candidate cluster centre. Candidate cluster centres are assigned potentials based on the distance to actual data points, and, following an iterative procedure, grid points with high potentials are selected as cluster centres. This method provides a simple and effective method for cluster estimation and is less sensitive to noise (Pal & Chakraborty, 2000), although the computational complexity increases exponentially with the dimension of the data: a problem space with $m$ variables each having $n$ grid lines results in $n^m$ grid points as candidate cluster centres. The subtractive clustering method, which is adopted in this thesis, is a modification of the mountain clustering method.

The **subtractive clustering** method defines each data point as a candidate cluster centre, limiting the number of potential cluster centres. In subtractive clustering, cluster centres are selected based on the density of surrounding data points. For $n$ data points $\{x_1, x_2, ..., x_n\}$ in an $M$-dimensional space, a neighbourhood with radius, $r_{\alpha}$, is defined and each data point is associated with a measure of its potential to be the cluster centre. The potential for point $i$ is

$$P_i = \sum_{j=1}^{n} e^{-\alpha |x_i - x_j|^2}$$  

Eq. 2.1,
where $\alpha = 4/r_a^2$ and $\|\|$ is the Euclidean distance. The data point with the highest potential, $P_i^*$, at location $x_i^*$ is selected as the first cluster centre, and, after obtaining the $k$th centre, the potential of other points $P_i$ is reduced based on their distance from the cluster centre:

$$P_i \leftarrow P_i - P_i^* e^{-\beta \|x_i - x_i^*\|^2}$$  \hspace{1cm} \text{Eq. 2.2},

where $\beta = 4/r_b^2$ and $r_b$ is a positive constant that defines the neighbourhood with significant reduction in potential. Data points close to location $x_i^*$ will have very low potential and low likelihood of being selected in the next iteration. The iteration stops when the potential of all remaining data points is below a threshold defined as a fraction of the first potential $P_1^*$. This criterion is complemented by other cluster centre rejection criteria (see Algorithm 3, section 3.4.2). For a set of $m$ cluster centres $\{x_1^*, x_2^*, \ldots, x_m^*\}$ in an $M$ dimensional space, with the first $N$ and last $M-N$ dimensions respectively corresponding to input and output variables, each selected cluster centre $x_i^*$ represents a rule of the form:

$$\text{if } \{\text{input is near } y_i^*\} \text{ then output is near } z_i^*$$

where $y_i^*$ and $z_i^*$ are components of $x_i^*$ containing the coordinates in the input and output space respectively. Given an input vector $y$, the degree of fulfilment of $y$ in rule $i$ is:

$$\mu_i = e^{-\alpha \|y - y_i\|^2}$$

The advantages of the subtractive clustering method over the mountain clustering method are that (i) the computation is proportional to the number of data points, and independent of the data dimension; (ii) there is no need to specify the grid resolution, which necessitates trade-offs between accuracy and model complexity; and (iii) the subtractive clustering method extends the cluster rejection criteria used in the mountain clustering method (Chiu, 1997).

For example, consider a simulated univariate time-series with lags at 12 units (generated using a seasonal ARIMA model, SARIMA $(1,0,0)(0,0,1)_{12}$) (Figure 2.9).
This univariate time series can be transformed into an $M$ dimensional vector by using a windowing scheme. Assuming a multiple-input single-output (MISO) model, with two inputs (i.e. window size = 2), the resulting time series is a 3-dimensional vector (Figure 2.10), with the first $N = 2$ and the last $M-N = 1$ dimensions corresponding to input and output variables, respectively.

Using the subtractive clustering algorithm, data is normalised into a unit hypercube, then cluster centres and the range of influence of each cluster centre in each dimension are computed. The distribution of clusters is such that cluster centres are located in areas where data is concentrated. A few clusters, in this case five clusters, representing five rules, cover most of the problem space (Figure 2.11).

In comparison, if mountain clustering had been used, with a resolution of three grid lines per variable, nine rules would be required to cover the same problem space. Also, if grid partitioning with pre-specified fuzzy sets (section 2.3.1) is used, portions of the hypercube without data will also be assigned fuzzy sets and rules, which will be redundant (rarely fire).
Cluster radius, $r_a = 0.5$ has been used in the example, and it can be observed that some data points do not belong to any cluster. This can be addressed by using smaller radius size. However, a smaller radius size might result in overfitting, leading to poor generalisation and degraded out-of-sample forecast performance.

2.3.3 Criticism of Fuzzy-based Soft Computing Techniques

Fuzzy models described in sections 2.3.1 and 2.3.2 have been used for the analysis of time series. Recall that these data-driven models, having advantages of simplicity, speed and high performance, often serve as initial models, and are refined using other more complex methods. The refinements involve the use of fuzzy models, combined with other soft computing approaches, such as neural networks and genetic algorithms, and are meant to improve the forecast accuracy of fuzzy models, by enhancing the system identification and optimisation technique employed. A critical examination of the literature suggests that using more complex models does not necessarily result in significant improvement in the forecast accuracy of fuzzy models. This may be because of the so-called bias/variance dilemma (Geman et al, 1992), in which bias and variance components of simple and complex models have different impacts - complex models tend to exhibit low bias and high variance, while simple models are typically characterised by high bias and low variance.

For instance, an observation of the forecast accuracy of models reported in the literature in the past seven years, and the impact of the use of more sophisticated models on forecast accuracy, shows a mixed picture. The comparison of different hybrid fuzzy models is complicated, since most models reported use different data sets to evaluate the effectiveness of such models, and, model performance may depend on the characteristics of the underlying data generating system (Schiavo & Luciano, 2001). There exists a data set, the Mackey Glass (MG) chaotic time series (Mackey Glass, 1977), that is generally used as a benchmark for comparing soft computing time series models, although it can be argued that this time series is not characterised by all the interesting features and patterns that are of interest in real-world time series. In the following, we review the forecast performance of hybrid fuzzy models on the MG data set in order to trace the evolution of forecast performance achieved using increasingly complex hybrid models. We emphasise that this review is limited in that it focuses on:

i) forecast models in the literature that report results on the MG data set;

ii) hybrid models that have fuzzy systems as one of the components;
iii) results reported in journal papers, which are deemed to have undergone more stringent review processes; and

iv) results reported in the past seven years, which represent the state-of-the-art.

We use Jang's (1993) model, one of the most cited hybrid models, as a baseline for performance comparison. Hybrid fuzzy techniques reported in the literature, which used the MG data set for evaluation, can be classified into three broad categories: neuro-fuzzy (NF) models that use a combination of neural networks and fuzzy models; genetic-fuzzy (GF) models that employ a hybrid of genetic algorithms and fuzzy systems; and genetic-neuro-fuzzy (GNF) systems, which are combinations of genetic algorithms, fuzzy models, and neural networks. In order to provide a fair basis for comparison, all the methods reported are compared on the non-dimensional error index (NDEI), defined in Lapedes and Farber (1987) and reported in Jang (1993). The NDEI, also referred to as the normalised root mean square error (NRMSE), is defined as root mean square error divided by the standard deviation of the target series:

$$\text{NDEI} = \frac{\text{RMSE}}{\sigma} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - x'_i)^2}$$

where RMSE is the root mean square error, $\sigma$ is the standard deviation of the test set, $N$ is the number of data points in the test set, and $x_i$ and $x'_i$ denote the $i$th observed and predicted value determined by the forecast model, respectively. Since the standard deviation of the test set is available, all results reported using the (R)MSE statistic are converted to the NDEI metric.

The forecast accuracy of models based on these methods for the MG data set is reported (Table 2.4). The results indicate that, compared to the benchmark ANFIS model, more sophisticated hybrid models do not necessarily result in improved forecast performance.

First, we compare the forecast performance of each class of hybrid model to the ANFIS benchmark model. Out of all the neuro-fuzzy models proposed as improvements to the adaptive neuro-fuzzy inference system (ANFIS) proposed by Jang (1993), only the neuro-fuzzy inference method for transductive reasoning, NFI (Song & Kasabov, 2006) results in better forecast performance than the ANFIS method. However, this improved performance is at a cost – for each test data sample, a local transductive model is generated; for the Mackey-Glass data set with 500 test samples, 500 local models will be generated. The computational complexity of this model is significant when compared to the ANFIS method in which a single model with 16 rules is generated for the test data set. The use of one model per sample is indicative of overfitting.
Similarly, fuzzy hybrid models with genetic algorithms do not result in significantly improved forecast models, compared to the ANFIS model - the fuzzy model with constrained evolutionary optimization, F-CEO (Kim et al, 2004), and the multi-objective hierarchical genetic algorithm (MOHGA) model, (Wang et al, 2005), only show results that are comparable to ANFIS. The same is true for GNF models, where both the hybrid evolutionary network fuzzy system, HENFS, (Li et al, 2006), and the self-adaptive neural fuzzy network with group-based symbiotic evolution (SANFN-GSE) method (Lin & Yu, 2006) show comparable results to the ANFIS method.

Table 2.4. Forecast results of different hybrid models on Mackey-Glass data set.

<table>
<thead>
<tr>
<th>Model</th>
<th>Name</th>
<th>NDEI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benchmark model: ANFIS (Jang, 1993)</td>
<td></td>
<td>0.007</td>
</tr>
<tr>
<td>Neuro-Fuzzy</td>
<td>NFI (Song &amp; Kasabov, 2005)</td>
<td>0.004</td>
</tr>
<tr>
<td></td>
<td>SuPfuNIS (Paul &amp; Kumar, 2002)</td>
<td>0.014</td>
</tr>
<tr>
<td></td>
<td>DENFIS (Kasabov &amp; Song, 2002)</td>
<td>0.016</td>
</tr>
<tr>
<td></td>
<td>HFNQ (de Souza et al, 2002)</td>
<td>0.032</td>
</tr>
<tr>
<td></td>
<td>D-FNN (Wu &amp; Er, 2000)</td>
<td>0.065</td>
</tr>
<tr>
<td>Genetic-Fuzzy</td>
<td>F-CEO (Kim et al, 2004)</td>
<td>0.007</td>
</tr>
<tr>
<td></td>
<td>MOHGA (Wang et al, 2005)</td>
<td>0.009</td>
</tr>
<tr>
<td></td>
<td>MMGF (Rojas et al, 2001)</td>
<td>0.158</td>
</tr>
<tr>
<td>Genetic-Neuro-Fuzzy</td>
<td>HENFS (Li et al, 2006)</td>
<td>0.006</td>
</tr>
<tr>
<td></td>
<td>SANFN-GSE (Lin &amp; Yu, 2006)</td>
<td>0.008</td>
</tr>
<tr>
<td></td>
<td>GEFREX (Russo, 2000)</td>
<td>0.030</td>
</tr>
<tr>
<td></td>
<td>EfujNN (Kasabov, 2001)</td>
<td>0.046</td>
</tr>
</tbody>
</table>

Second, we examine the forecast performance of all the models, compared to the ANFIS benchmark model (Figure 2.12). The figure indicates that no hybrid method consistently outperforms the ANFIS model, and only the HENFIS and NFI models have better forecast performance. Apart from the architectural complexity of the models, the computational complexity of models is another consideration. Out of all the models that provide comparable (or better) forecast performance to the ANFIS model, i.e. NFI, HENFIS, F-CEO, SANFN-GSE and MOHGA, only the HENFIS and F-CEO have similar computational complexity: F-CEO has 16 tuneable parameters, population size of 100 and maximum generation of 300; HENFIS has 16 rules, 104 parameters and the reported results were based on 100 epochs. These compare favourably with ANFIS' 16 rules, 104 parameters and 500 epochs.
In contrast, NFl generates a local model for each testing sample, and SANFN-GSE uses 500 generations of training, repeated 50 times. Also, although MOHGA reports an impressive fuzzy model configuration, with three fuzzy sets and one optimised rule, there is no indication of the population size and number of epochs used to achieve this optimised model. The simulation time provides a measure of the complexity – MOHGA is reported to run for 300mins, as opposed to ANFIS, which takes less than 5mins to run on a standard PC. If the complexity of the ANFIS model is increased, by using four membership functions, rather than the two membership functions reported in Jang (1993), an NDEI of 0.002 is obtained (red circle in Figure 2.12). With this configuration, the ANFIS model significantly outperforms more sophisticated techniques, although the model now has 256 rules – another case of overfitting. These results suggest that models developed from complex structure identification and parameter optimisation techniques may overfit or, at best, produce incremental improvement in the forecast performance of such models, relative to simpler models.

![Figure 2.12](image.png)

Figure 2.12. Forecast accuracy (NDEI) of hybrid models plotted on a log scale.

In addition, fuzzy systems are typically generated from raw time series. This, perhaps, is because of the universal approximation property, which suggests that fuzzy models can be used to directly model real-world time series. Mathematical proofs that support the universal approximation capability of Takagi-Sugeno fuzzy systems have been provided (see, for example Ying, 1998; Zeng et al, 2000). However, such proofs place a lower limit on the number of fuzzy sets required for each input variable, in order to guarantee a required approximation capability (Ying, 1994). As the desired approximation error decreases and approaches zero, the number of fuzzy sets required increases and approaches infinity (Ying, 1998):

$$\lim_{n \to \infty} F_n(x) = P_n(x)$$
where $F_n(x)$ is a fuzzy system with $n$ fuzzy sets, and $P_h(x)$ is a polynomial of order $h$. This has practical implications for the configuration of fuzzy systems for time series forecasting: models with a high number of fuzzy sets and rules overfit training data, but exhibit significantly degraded performance when applied to out-of-sample data (Guillaume, 2001) – an instance of low bias and high variance. Such single global systems will be complex, and have a questionable value in real-world applications. It has been argued that the universal approximation property ceases to be valid if, as a practical limitation, the number of rules and fuzzy sets are bounded (Tikk and Baranyi, 2003). This has led some to argue that soft computing techniques are best able to model pre-processed data, and that forecast accuracy of such techniques are degraded when nonstationary data is used (Zhang et al, 2001). Instructively, none of the models discussed in the this section uses any form of pre-processing. All the analysis are carried out on the raw data.

An alternative approach to the generation of better forecast accuracy might be to reduce data complexity via pre-processing, rather than develop newer methods. Various strategies have been used for pre-processing data in order to make data stationary, or to obtain so-called components of time series (section 2.1.1). Some of these techniques are described in section 3.2. In particular, wavelet analysis has been widely used for decomposing time serial data, prior to the application of modelling techniques. Wavelets are powerful analysis tools that provide both temporal and frequency representations of a time series, by decomposing data into different frequency components with the temporal resolution being matched to the scale. In the next section, we provide an overview of time and frequency domain analysis of time series, and discuss the use of the wavelet analysis for the decomposition of time series.

### 2.4 Multiscale Wavelet Analysis of Time Series

#### 2.4.1 Time and Frequency Domain Analysis

Conventional and soft computing methods so far described use time domain properties of data for the generation of models. However, it has been argued that hidden structures may be present in time series that are not readily apparent in the time domain, but can be detected in frequency domain analysis of such series (Chatfield, 2004). Conventionally, a spectral plot is used to examine such hidden structures, particularly the cyclic structure of time series, in the frequency domain. The spectral plot is able to determine the number of frequency components and to detect the dominant cyclic frequency, if any, which is embedded in a time
series, even in the presence of noise. The discrete Fourier transform (DFT), which reveals periodicities present in time series and their relative strengths, can be defined as:

\[ X_k = \sum_{t=0}^{N-1} x_t e^{-j2\pi ft}, \quad k = 0, 1, \ldots, N - 1 \]

where \( X_k \) is the \( k \)th spectral sample, frequency \( f_k = k/N \), \( N \) is the number of samples. Also, the inverse DFT approximates a discretely sampled time series \( X_t \) using a linear combination of sines and cosines:

\[ X_t = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{j2\pi ft}, \quad t = 0, 1, \ldots, N - 1 \]

For example, consider a random signal and a noisy signal with three periodic components, with periods 4, 8 and 16. The time plot provides no clear indication as to the presence of periodic components in both the random signal and the periodic signal (Figure 2.13).

However, the corresponding frequency analysis clearly shows that there is no dominant frequency component in the random signal, and that there are three distinct components in the periodic signal. Although the Fourier transform enables the detection of spectral components, two limitations of the Fourier analysis are critical:

i) the Fourier transform requires the use of data that are stationary in the mean – the presence of trends should be removed before using Fourier transforms (Croarkin & Tobias, 2006).

ii) the Fourier transform has full frequency information but no temporal resolution - it provides no information about when, in time, frequency components exist. The
Fourier method therefore assumes that all frequency components are present at all times, i.e. the frequency content of the signal is stationary.

In time series where frequency components are present only in specific time segments, the Fourier analysis is unable to provide information about the temporal location of the frequency components.

For example, if the three periodic components in the previous example do not exist simultaneously, but sequentially i.e. periods 4, 8, 16 respectively occurring at time intervals 1-50, 51-130 and 131-200, the frequency analysis correctly indicates the presence of these components (Figure 2.14). However, there is no information about where in time these components occur, and the frequency plot is strikingly similar to that of the periodic signal where the components are present at all times (Figure 2.13).

The literature on wavelet analysis suggests that, in order to provide time information for the Fourier transform, the Short-Time Fourier Transform (STFT) or Gabor transform was proposed (see Gençay et al, 2002 for details). The STFT assumes that a portion of any signal is stationary i.e. frequency components in a fixed interval of the signal exist at all times in that interval. The STFT then uses a fixed-width sliding window, and computes the Fourier transform for each window. The limitation of this approach is that, for events falling within the fixed width window, the time resolution problem inherent in the Fourier transform is present. Moreover, following Heisenberg’s uncertainty principle, time and frequency resolution cannot be simultaneously achieved (Walker, 1999).

2.4.2 The Discrete Wavelet Transform (DWT)

The wavelet transform was proposed to address limitations of the Fourier transform by using basis functions (mother wavelets) that are translated and dilated to provide good time resolution for high-frequency events, and limited time resolution for low frequency events. Proponents of wavelet analysis suggest that wavelets are mathematical functions that ‘cut’ up
data into different frequency components, and study each component with a resolution matched to its scale (Daubechies, 1992; Daubechies, 1996; Graps, 1995). Wavelets have been described as robust parameter-free tools (Pen, 1999) for identifying the deterministic dynamics of complex financial processes (Gençay et al., 2002). Ramsey (1999) asserts that wavelets can approximately decorrelate long memory processes, represent complex structures without knowledge of the underlying function, and locate structural breaks and isolated shocks. Gençay et al. (2001) have argued that wavelets can separate intraday seasonal components of high frequency, non-stationary time series, leaving the underlying non-seasonal structure intact. A schematic representation of the effect of using time, Fourier, STFT and wavelet analysis is presented in Figure 2.15.

As indicated in the figure, the Fourier transform can achieve high frequency resolution, without time resolution, while, with the time domain representation, no frequency resolution is present, but the time resolution is very good. The STFT analyses only a small windowed segment of the signal at a time, eventually providing a mapping of the signal into two dimensional function of time and frequency (Figure 2.15c). The STFT is an improvement on both the Fourier (frequency) and time representations, because it provides a measure of time and frequency resolutions. However, the use of a fixed window size at all times and for all frequencies is a limitation of this method, since the resolution is the same at all locations in the time-frequency plane. The wavelet representation addresses this limitation, by adaptively partitioning the time-frequency plane, using a range of window sizes (Figure 2.15d). At high
frequencies, the wavelet transform gives up some frequency resolution compared to the Fourier transform (Gencay et al., 2002).

Fourier basis functions - sines and cosines - are, by definition, smooth or regular, nonlocal and stretch to infinity. Consequently, such functions poorly approximate sharp spikes and other local behaviour. On the other hand, the wavelet transform uses an analysing or mother wavelet as the basis function. Wavelet basis functions have compact support i.e. they exist only over a finite time limit, and are typically irregular and asymmetric. Such functions, it has been suggested, are better suited for the analysis of sharp discontinuities, nonstationarity and other transient or local behaviour (Graps, 1995). Good temporal representation of high frequency events is achieved by using contracted versions of the prototype wavelet, while good frequency resolution is achieved with dilated versions.

For example, consider a square-wave function, based on the simplest wavelet basis function - the Haar wavelet filter (Figure 2.16a). The wavelet function can be shifted or translated forward in time (Figure 2.16b) in order to capture events at a particular location. Also, to capture low frequency events, the filter can be stretched or dilated (Figure 2.16c), and to capture high frequency events, it can be constricted or negatively dilated (Figure 2.16d).

![Wavelet Diagram](image)

Figure 2.16. (a) Square-wave function mother wavelet (b) wavelet positively translated in time (c) wavelet positively dilated in time and (d) wavelet negatively dilated in time (Gencay, 2001).

Conceptually, the generation of wavelet coefficients for a time series involves five steps (The Mathworks, 2005):

i) Given a signal $X_t$ and a wavelet function $\psi_{j,k}$, compare the wavelet to a section at the start of the signal (Figure 2.17a).
ii) Compute the coefficient, $c_{j,k}$, which is an indication of the correlation of the wavelet function with the selected section of the signal.

iii) Shift the wavelet to the right and repeat steps (i) and (ii) until all the signal is covered (Figure 2.17b).

iv) Dilate (scale) the wavelet and repeat steps (i) through (iii) (Figure 2.17c).

v) Repeat steps (i) through (iv) for all scales to obtain coefficients at all scales and at different sections of the original signal.

Figure 2.17. Generating wavelet coefficients from a time series.

The set of scales and positions of the analysing wavelet determines the type of analysis obtained: for a continuous wavelet transform (CWT), the wavelet is shifted smoothly over the full domain of the analysed function; for a discrete wavelet transform (DWT), the analysis is made more efficient by using only a subset of scales and positions — this choice is based on the powers of two of the form $2^{j-1}$, $j = 1, 2, 3, \ldots$, and are referred to as dyadic scales (Percival & Walden, 2000).

Discrete wavelet transforms (DWTs) can be developed from various wavelet families. A wavelet family comprises wavelet basis functions, derived from a single prototype wavelet filter, the mother wavelet, and obtained over all scales and translations or positions. Examples
of families of wavelet basis functions include Haar, Daubechies, Biorthogonal, Coiflets, Symlets, Morlet and the Mexican Hat (The Mathworks, 2005). Basis functions or filters used for wavelet analysis are characterised by a set of properties. By definition, a wavelet filter \( h_t = (h_0, \ldots, h_{L-1}) \), of even and finite length \( L \) must have unit norm or energy:

\[
\sum_{l=0}^{L-1} h_l^2 = 1; \quad \text{Eq. (2.3)},
\]

the wavelet filter must integrate or sum up to zero:

\[
\sum_{l=0}^{L-1} h_l = 0; \quad \text{Eq. (2.4)},
\]

and it must be orthogonal to its even shifts:

\[
\sum_{l=0}^{L-1} h_l h_{l+2n} = \begin{cases} 1 & n = 0 \\ 0 & \text{otherwise} \end{cases} \quad \text{Eq. (2.5)},
\]

where \( n \) is a non-negative integer. The Daubechies wavelet, \( D(4) \), which is used in this thesis, has high-pass or wavelet filter coefficients defined as:

\[
h_0 = \frac{1 - \sqrt{3}}{4\sqrt{2}}, \quad h_1 = \frac{-3 + \sqrt{3}}{4\sqrt{2}}, \quad h_2 = \frac{3 + \sqrt{3}}{4\sqrt{2}}, \quad \text{and} \quad h_3 = \frac{-1 - \sqrt{3}}{4\sqrt{2}}, \quad \text{Eq. (2.6)},
\]

and satisfy the conditions defined in Eqs. (2.3)–(2.5). Given the wavelet coefficients defined for the wavelet or high-pass filter in Eq. (2.6), the related low-pass or scaling coefficients, \( g_t \), are determined using the quadrature mirror relationship (Gençay et al, 2002):

\[
g_l = (-1)^{l+1} h_{L-1-l} \quad \text{for } l = 0, \ldots, L - 1
\]

For the Daubechies \( D(4) \) wavelet filter, \( L = 4 \), and the corresponding scaling filters to the wavelet filters defined in Eq. (2.6), are \( g_0 = -h_3, g_1 = h_2, g_2 = -h_1 \) and \( g_3 = h_0 \):

\[
g_0 = \frac{1 + \sqrt{3}}{4\sqrt{2}}, \quad g_1 = \frac{3 + \sqrt{3}}{4\sqrt{2}}, \quad g_2 = \frac{3 - \sqrt{3}}{4\sqrt{2}}, \quad \text{and} \quad g_3 = \frac{1 - \sqrt{3}}{4\sqrt{2}}, \quad \text{Eq. (2.7)}.
\]

Next, we describe the computation involved in obtaining the DWT of a finite-length time series. Let \( X \) be a column vector of real-valued time series, with dyadic length \( N = 2^J \) (\( J \) is a positive integer). A length \( N = 2^J \) column vector of level \( J \) discrete wavelet coefficients, \( w \), can be obtained from:
where $W$ is a $N \times N$ real-valued matrix defining the DWT, and $W^T W = I_N$. Vector $w$ contains the transform coefficients, and its first $N - N / 2^J$ and last $N / 2^J$ elements are, respectively, the wavelet and scaling coefficients. The vector in Eq. (2.8) can be reorganized as:

$$w = [w_1, w_2, \ldots, w_J, v_J]^T$$

where $w_j$ is a length $N / 2^J$ vector of wavelet coefficients, and $v_j$ is a length $N / 2^J$ vector of scaling coefficients. This implies that, given a dyadic length $N$ vector, for each scale of length $\lambda_j = 2^{j-1}$, there are $N_j = N / 2^J$ wavelet coefficients, $w_j$. For example, given a length $N = 64$ vector, with $J = 4$, $w_1$, $w_2$, $w_3$ and $w_4$ are respectively vectors of length 32, 16, 8 and 4; and $v_4$ is a vector with a length of 4.

The matrix $W$ comprises wavelet and scaling filter coefficients arranged on a row-by-row basis:

$$W = [W_1, W_2, \ldots, W_J, V_J]^T$$

where $W_j$ is an $N/2^J \times N$ dimensional matrix of zero-padded, circularly shifted (by factors of $2^J$) wavelet filter coefficients in reverse order. For scale 1 i.e. $J = 1$ wavelet filter coefficients,

$$W_1 = [h_1^{(2)}, h_1^{(4)}, \ldots, h_1^{(N-2)}, h_1],$$

and

$$h_1 = [h_{1,N-1}, h_{1,N-2}, \ldots, h_{1,1}, h_{1,0}]^T$$

$$h_1^{(2)} = [h_{1,1}, h_{1,0}, h_{1,N-1}, h_{1,N-2}, \ldots, h_{1,3}, h_{1,2}]^T$$

$$\vdots$$

$$h_1^{(N-2)} = [h_{1,N-3}, h_{1,N-4}, \ldots, h_{1,1}, h_{1,0}, h_{1,N-1}, h_{1,N-2}]^T$$

The coefficients $h_{1,0}, \ldots, h_{1,L-1}$ are even length $L$ wavelet filters padded with $N - L$ zeros, i.e. for time series length $N = 16$, and filter length $L = 4$, the matrix is padded with 12 zeros:
and so on, where $h^{(2)}_1$ is the circularly shifted version of $h_1$. In general, for each scale $j$, zero-padded wavelet filter coefficients, $h_j$, are circularly shifted by factors of $2^j$, and $V_j$ is a row vector with all its elements equal to $N_0^{0.5}$. Further details of the methods for explicitly computing the wavelet ($h_j$) and scaling ($g_j$) filter coefficients for scales $j = 1, ..., J$ are available in the literature (Percival & Walden, 2000; Gençay et al, 2002).

### 2.4.2.1 DWT Implementation using a Pyramidal Algorithm

In practice, Mallat's pyramidal algorithm (Mallat, 1989) is used to achieve efficient implementation of the DWT (Figure 2.18).

![Flow diagram illustrating the pyramidal method for decomposing $X_i$ into wavelet coefficients $w_j$ and scaling coefficients $v_j$.](image)

Figure 2.18. Flow diagram illustrating the pyramidal method for decomposing $X_i$ into wavelet coefficients $w_j$ and scaling coefficients $v_j$. 
Given a time series $X_t$ of length $N$, wavelet (high-pass) filters, $h_j$, and scaling (low-pass) filters, $g_j$, are used to compute wavelet and scaling coefficients $w_1$ and $v_1$ by convolving the time series with $h_1$ and $g_1$, and subsampling the filter outputs to obtain $N/2$ coefficients. Downsampling by 2 or dyadic decimation involves keeping only even indexed elements. Next, the downsampled outputs $w_1$ are kept as wavelet coefficients, and the subsampled output ($v_1$) of the $g_1$ filter is again convolved with the wavelet and scaling filters, and the outputs are downsampled to obtain wavelet and scaling coefficients $w_2$ and $v_2$, and so on. This process is repeatable up to $J = \log_2(N)$ times, and gives the vector of wavelet and scaling coefficients, $w$.

We illustrate DWT decomposition by considering a time series obtained from the summation sine waves (Figure 2.19). This series has length $N = 1000$, and the number of decomposition levels $J = \log_2(N) \leq 9$. Recall that the DWT requires the use of a dyadic length time series. Where $X_t$ is not a dyadic length time series, as in this case, the series is zero-padded in order to compute the wavelet transform. This is one of the limitations of the DWT and is addressed using a variant of the DWT, the maximal overlap DWT (MODWT), discussed in Chapter 3. A partial DWT of the time series, with $J_p = 3$ is computed, and the resulting scaling coefficients ($v_3$) and wavelet coefficients ($w_1 - w_3$) are plotted.

![Figure 2.19. Plot of original time series (top) and its wavelet decomposition structure (bottom).](image-url)
As can be seen from the example, the wavelet decomposition has resolved the original time series into four components: a scaling signal \((v_3)\) and three wavelet signals \((w_1 - w_3)\). The original time series \(X_t\) can be recovered from these components without loss of information. This is achieved by upsampling (in this case inserting a zero between adjacent values) the final coefficients, \(w_1\) and \(v_2\), convolving with the respective filters, and adding up the filtered vectors. Reconstruction filters \(h'\) and \(g'\) are used to recover the signal from its wavelet components.

### 2.4.2.2 DWT Multiresolution Analysis

Although the wavelet decompositions described so far have interesting characteristics, it is still not suitable for predictive analysis of time series, since, by downsampling, half of the data is ‘lost’ at each stage, and the components are not aligned in time with the original signal. What is needed is time series decomposition where the components have the same number of samples \(N\) as the original signal, with coefficients at time \(t\) aligned across scales, and where the original series can be recovered via simple addition of the components. Also, for predictive purposes, we need to be able to reconstruct approximations or smooths \(S_t'\) and details \(D_t'\) from the associated wavelet and scaling coefficients. This is achieved by convolving upsampled wavelet or scaling coefficients with a vector of zeros, which has a length equal to the length of the coefficient (Figure 2.20).

![Flow diagram illustrating the pyramidal method for reconstructing wavelet approximations \(S_t\) and details \(D_t\) from wavelet coefficients \(w_t\) and scaling coefficients \(v_t\).](image)

**Figure 2.20.** Flow diagram illustrating the pyramidal method for reconstructing wavelet approximations \(S_t\) and details \(D_t\) from wavelet coefficients \(w_t\) and scaling coefficients \(v_t\).
In the figure, wavelet approximation $S_j'$ is reconstructed from the associated scaling coefficient, $v_j$, using reconstruction filter $h'$. Similarly, wavelet detail $D_j'$ is reconstructed from the associated wavelet coefficient, $w_j$, using reconstruction filter $g'$. The wavelet approximation $S_j'$ and wavelet detail $D_j'$ can be used to define a wavelet multiresolution analysis (MRA), where $j$th level wavelet detail $D_j'$ characterises the high frequency components at scale $j$, and the wavelet approximation (or wavelet smooth) $S_j$ characterises low frequency components. Given a time series $X$, the DWT-based MRA can be defined as:

$$X = \sum_{j=1}^{J} D_j' + S_j'.$$

The time series of sum of sine waves previously decomposed using wavelet and scaling coefficients can be similarly analysed using wavelet approximations and details (Figure 2.21).

Figure 2.21. Five-level multiscale wavelet decomposition of time series $X$, showing the wavelet approximation, $S_j'$, and wavelet details $D_j' - D_j'$. 
Due to the complex and uncertain nature of real-world data, soft computing techniques have found increased use in time series analysis. Specifically, fuzzy systems appear to facilitate time series mining, and provide qualitative interpretation of input/output data behaviour. Such models however have to cope with a compromise between accuracy and rule interpretability. Two classes of fuzzy models are generally identified: complex rule generation mechanisms and ad hoc data driven models. The former employs hybrid systems while the latter uses data partitioning techniques.

Generally, fuzzy models reported in the literature do not use data pre-processing to improve forecast performance. Instead, performance improvements are usually based on developing more sophisticated models, by enhancing the system identification and optimisation technique employed. A review of the literature indicates that the use of increasingly sophisticated fuzzy and hybrid-fuzzy models does not necessarily result in significant improvement in forecast accuracy, due to the classic bias-variance dilemma. An alternative approach, based on reducing data complexity through data pre-processing, may be beneficial. The multiscale wavelet transform facilitates the exploration of events that are local in time and helps in the identification of deterministic dynamics of complex processes. The use of wavelets, it has been argued, formalizes the notions of data decomposition. In the next chapter, we propose the use of a wavelet-based framework for pre-processing data prior to the application of a fuzzy model, and provide methods for testing the suitability of wavelet-based data pre-processing.
Chapter 3

Fuzzy-Wavelet Method for Time Series Analysis

3.1 Introduction

Fuzzy systems are amongst the class of soft computing methods, referred to as universal approximators, which are theoretically capable of uniformly approximating any real continuous function on a compact set to any degree of accuracy (Kosko, 1992; Wang, 1992). In particular, Takagi-Sugeno (TS) models approximate nonlinear systems using a fuzzy mixture of locally linear models (Takagi & Sugeno, 1985). In this scheme, fuzzy regions or ‘patches’ are defined in the input space, and each region is characterised by a linear input-output sub-model. The overall representation of the nonlinear system is obtained via a fuzzy aggregation of locally valid linear models. Unlike piecewise linear relations, this method attempts to facilitate smooth transition between local linear models, with a view to preventing discontinuities at the boundaries of local models.

Fuzzy systems have been used in the analysis and modelling of time serial data in a number of different application areas. Typically, such global models approximate nonlinear functions by combining local linear models into a single global model, and are built on the assumption that the variables under consideration are stationary i.e. vary in a uniform manner. However, there has been considerable debate, on the presumed ability of the resultant ‘monolithic global models’ (Zhang et al., 2001) to directly analyse real-world time series, which are often characterized by complex local behaviour – mean and variance changes, seasonality and other features. Song and Kasabov (2005) argue that developing global models, which are valid for the whole problem space, is a difficult and often unnecessary task.

Limitations of global models have been addressed through the use of various pre-processing strategies, which have been adopted to extract time series components, prior to the application of soft computing methods. Ramsey (1999) suggests that traditional methods for decomposing time series into components are informal. Such methods may be useful for
analysing time series with dominant trend and seasonality (Chatfield, 2004). However, even in such cases, defining and modelling dominant components will depend on the experience and expertise of the analyst, and on the availability of information about the data generating process. This presents a significant challenge as underlying characteristics of most astrophysical phenomena and real world processes, such as financial trading, are not completely understood. Modelling is further complicated when dealing with noisy, high frequency data, like financial market data, where complex interactions occur at different time scales. To address these limitations, Ramsey (1999) advocates the adoption of a wavelet-based approach, which formalizes notions of decomposing time series into components, 'without knowing the underlying functional form' (ibid. pp. 2594). The wavelet transform provides a local representation of a signal, in both time and frequency domain. Motivated by Ramsey's (1999) assertion that time series decomposition is formalized by using wavelets, we classify pre-processing methods into two categories: (i) conventional, ad hoc or 'informal' techniques, and (ii) 'formal', wavelet-based techniques.

Whilst time series pre-processing may be beneficial, it may also introduce artefacts into time series, which have neither trends nor seasonality. Given the strong data-dependent nature of fuzzy systems, the introduction of artefacts through data pre-processing may result in significantly degraded forecast performance. We argue that it is vital that analysis methods involving data pre-processing have a measure of 'intelligence' to ascertain the suitability of the pre-processing method of choice on the data under analysis (Popoola et al, 2005; Popoola & Ahmad, 2006a). Furthermore, in contrast to the ad hoc approach to data pre-processing currently used in soft computing literature, we propose a systematic method for selecting the pre-processing strategy, making automation possible (Popoola & Ahmad, 2006b). The selective use of pre-processing techniques is intended to prevent the introduction of artefacts into time series, and the attendant degradation in model accuracy. We show in Chapter 4, by an analysis of synthetic and real-world time series, the effects of different pre-processing methods and the advantage of using a framework for systematic selection.

In this chapter, we use autocorrelation functions to understand the autocorrelations of different time series, and propose a structured approach, consistent with Occam's razor, for determining the suitability of specific data pre-processing techniques. We present a fuzzy-wavelet model, which uses wavelet-based pre-processing to decompose time series data prior to the application of a fuzzy model. In addition, the (wavelet) variance plot of a time series is used as an exploratory device for graphical assessment of the suitability of wavelet-based pre-
processing; and an automatic method for detecting variance breaks in time series is used for testing the suitability of wavelet-based pre-processing.

The structure of the chapter is as follows. We provide a discussion of data pre-processing models (3.2) including informal models (3.2.1) and formal models, specifically wavelet-based time series decomposition (3.2.2). This is followed by a description of diagnostic tests (3.3) based on the existence of autocorrelations (3.3.1) and variance homogeneity (3.3.2) in time series. A fuzzy-wavelet model for time-series analysis is then presented with the dictum of Occam that has inspired us: to apply complex (pre-processing) methods only when the time series is suited to it (3.4). We conclude the chapter by providing a summary of important points (3.5).

3.2 Data Pre-processing for Fuzzy Models

It can be argued that the forecast performance of data-driven techniques such as fuzzy systems depends critically on the type of data used to generate the model. Given the strong data-dependent nature of such techniques, there is a need to investigate the desirability, and methods, for reducing data complexity through data pre-processing. The effectiveness of data pre-processing on the prediction performance of neural networks, another class of universal approximators, has been investigated (Nelson et al., 1999; Virili & Freisleben, 2000; Zhang, 2003; Zhang & Qi, 2005). Some studies report a consistent improvement in forecast performance for models trained on pre-processed data, while others indicate conflicting performance. This, perhaps, implies that whilst forecast performance may improve with data pre-processing, this is not true for all data sets and all pre-processing strategies. The conflicting conclusions reported in the literature about the necessity and efficacy of data pre-processing may be due to inconsistent approaches to data pre-processing (Nelson et al., 1999: 360). Inevitably, sophisticated techniques are selected where simpler approaches might be sufficient.

In this research, we intend to extend previous work on the effects of data pre-processing to fuzzy models. We investigate the single-step forecast performance of subtractive clustering TSK fuzzy models for non-stationary time series, and examine the effect of different pre-processing strategies on the performance of the model. We argue that the use of appropriate data pre-processing techniques reduces data complexity, and enables fuzzy models generated on pre-processed data to exhibit better forecast performance. Also, we argue that there is a need for a more structured approach for determining the suitability of data pre-processing
techniques. Such conventional methods have the added benefit of being consistent with Occam's razor – simpler and potentially more effective pre-processing methods precede more sophisticated approaches.

### 3.2.1 Informal Approaches

A common assumption in time series analysis is that time series data have constant mean and variance i.e. they are stationary. This is normally true except when shocks are administered to the system generating the series, resulting in nonstationary in the variance, or there is a trend in the series, resulting in nonstationary in the mean. Pre-processing techniques, which facilitate stabilization of the mean and variance, and seasonality removal, are often applied to remove non-stationarity in data used to build soft computing models. Traditional data pre-processing methods generally focus on transformations and decomposition of time serial data. Transformations are used to i) stabilize the variance; ii) make seasonal effects additive; and iii) make data to be normally distributed. Such transformations usually involve logarithmic and square root conversions, which are special cases of Box-Cox power transformations. However, transformations introduce bias when data has to be 'transformed back'. Also, such transforms often have no physical interpretation (Chatfield, 2004). Data decomposition, on the other hand, typically involves separating trend, seasonal/cyclical and irregular components from a time series. The components obtained from time series are either specified by a well-informed modeller, or dictated by the decomposition technique.

We have investigated the effect of different pre-processing strategies on the single-step forecast performance of TSK fuzzy models, and explored the ability of such models to directly analyse non-stationary time series. Pre-processing techniques have been used to remove non-stationarity in data, prior to the application of the subtractive clustering fuzzy model: first difference for trend removal and seasonal difference for the removal of seasonality. In the following, we discuss the pre-processing techniques investigated in our work, limitations of an ad hoc approach, and the need for a systematic method for data pre-processing.

#### 3.2.1.1 Trend Removal

First difference is typically used for eliminating stochastic trends. For example, first difference is employed to stabilize the mean in the widely used ARIMA method (section 2.2.1). Differencing simply creates new series from an initial series. The backshift operator, $B$, provides a useful notation for representing $n$th-order difference:
where

\[ Bx_t = x_{t-1}. \]

Using this notation, we can represent the first difference as:

\[ \nabla x_t = x_t - x_{t-1} = (1 - B)x_t; \]

It has been argued that differencing may not always be appropriate for modelling trend, and that for deterministic trends, polynomial trend fitting may be more suitable (Virili & Freisleben, 2003). For example, a first order polynomial trend fitted to a time series \( x_t \), follows a simple regression model:

\[ x_t = \alpha + \beta t + \epsilon_t, \]

where \( \alpha \) and \( \beta \) are unknown intercept and slope parameters of the polynomial, and \( \epsilon_t \) is the residual error time series. In practice, it is difficult to distinguish between stochastic and deterministic trends, since standard statistical tests suffer from low power in differentiating between unit root and near unit processes (Zhang & Qi, 2005). Typically, a trial-and-error approach is used to determine the best method for trend removal.

### 3.2.1.2 Seasonality Removal

Seasonality is encountered in many time series. Seasonality in a time series may be multiplicative or additive. Removal of seasonality from monthly data is commonly achieved using a 12-month centred moving average (MA):

\[
\text{Sm}(x_t) = \frac{\frac{1}{2}x_{t-6} + x_{t-5} + x_{t-4} + ... + x_{t+5} + \frac{1}{2}x_{t+6}}{12}
\]

and \( x_t - \text{Sm}(x_t) \) eliminates additive seasonality, while \( x_t/\text{Sm}(x_t) \) removes multiplicative seasonality. Another method for removing seasonality is via seasonal differencing. Seasonal differencing removes stochastic seasonality from data that have fixed cycle lengths (Makridakis et al., 1998):

\[ \nabla_{12} x_t = x_t - x_{t-12} = (1 - B^{12})x_t \]

In some cases, after a seasonal difference, there still exists a trend in the time series. A further first difference is computed to remove this trend. Seasonal difference followed by first difference, which is equivalent to first difference followed by seasonal difference, can then be represented as:
\[ \nabla_{12,1} x_t = (x_t - x_{t-12}) - (x_{t-1} - x_{t-13}) = (x_t - x_{t-1}) - (x_{t-12} - x_{t-13}) = (1 - B)(1 - B_{12}^{12})x_t \]

### 3.2.1.3 Removing Trends and Seasonal Components: An Example

To illustrate the removal of trend and seasonal components, consider a univariate SARIMA(1,0,0)(0,1,1)$_{12}$ model, earlier discussed in section 2.3.2 (Figure 3.1).

![Figure 3.1. Time series generated from SARIMA(1,0,0)(0,1,1)$_{12}$ model.](image)

The time series exhibits an upward trend. A 'detrended' time series is obtained using first difference (Figure 3.2a) and first-order polynomial trend fitting (Figure 3.2b).

![Figure 3.2. Synthetic data 'detrended' using (a) first difference (b) first-order polynomial curve fitting.](image)

Whilst first differenced data seems to be stationary with respect to the mean, it appears that this is not the case with polynomial detrending, where a higher order polynomial may be more appropriate. This is an example of the subjective, trial-and-error nature of ad hoc data pre-processing.
The detrended time series (Figure 3.2) exhibits seasonal fluctuations, as the SARIMA(1,0,0)(0,1,1)_{12} model used to generate the series has a seasonal component. Moreover, the seasonal fluctuations are of increasing magnitude, indicating the presence of multiplicative seasonality. To obtain 'deseasonalised' series from the raw data, we apply seasonal difference (Figure 3.3a) using Eq. (3.2), and remove multiplicative seasonality (Figure 3.3b) using a 12-month centred MA Eq. (3.1).

On one hand, it can be observed that seasonal differencing appears to have removed not only the seasonal component, but also most of the trend component. The presence of a mild downward linear trend in the data (red line in Figure 3.3a) suggests that a further first difference may be beneficial (this is indeed the case, as we discuss later in section 4.3.2). On the other hand, data processed using a 12-month centred MA appear to retain some seasonal effects, and a trend component is clearly present. Theoretically, this should not be the case i.e. (multiplicative) seasonality ought to have been eliminated, since a 12-month centred MA was used (Makridakis et al., 1998; Chatfield, 2004). Here again, even though the use of a 12-month centred MA for dealing with multiplicative seasonality is well documented in the literature, in practice, a trial and error approach appears necessary in order to adequately deal with non-stationary time series.

3.2.1.4 Classical Decomposition

In previous sections, we have discussed pre-processing methods that involve the 'removal' of specific components - trend and seasonality. However, there are situations in which we may be interested not only in the time series generated by a process, but also in its constituents or components. So-called decomposition models have been developed to address this issue. Decomposition-based approaches enable forecasters to separate known elements of economic activity such as seasonal changes, in order to uncover changes that may be obscured by trend or seasonal changes.
Recall that time series are deemed to have components – trend \((T)\), seasonal \((S)\), cyclical \((C)\) and irregular \((I)\) components. In order to separate these components from a time series, a mathematical relationship between the components and the original series is often assumed, and time series are typically modelled as having additive or multiplicative components. The classical decomposition method is the basis for most decomposition methods currently in use. Due to the difficulty in defining and separating cyclical components, in this method, only three components are assumed: (i) the trend-cycle component, which represents the trend and cyclical component, (ii) the seasonal and the (iii) irregular components. The classical decomposition method comprises four steps (Makridakis et al., 1998):

i) Compute the trend-cycle using a centred 12 MA defined in Eq. (3.1)

ii) Compute the detrended series \(x'\) by subtracting the trend-cycle from the original series (additive model):

\[
x' = x - T = S_i + I_i
\]

or by dividing the original series by the trend cycle (multiplicative model):

\[
x' = \frac{x}{T} = S_i 	imes I_i
\]

iii) Estimate the seasonal component, which is assumed to be constant year in year out, by computing seasonal indices. The seasonal index for each month is obtained by averaging the detrended values for the month over all the years represented in the data. These 12 indices form a sequence that estimates the seasonal component for each year.

iv) Compute the irregular components by subtracting the seasonal component from detrended data (additive model)

\[
I_i = x_i - T_i - S_i
\]

or by dividing detrended data by the seasonal component:

\[
I_i = \frac{x_i}{T_iS_i}
\]

Components computed from the simulated series (Figure 3.1), using the additive model of the classical decomposition approach, is shown (Figure 3.4). In this case, the use of classical decomposition still leaves some structure related to the volatility in the irregular component, an indication that ad hoc application of the classical decomposition method is not suitable. There are variants of the classical decomposition method, such as the US Census Bureau X-12-ARIMA method, that have been reported in the literature. It has however been argued that
decomposition methods such as the X-12-ARIMA are ad hoc (Zhang & Qi, 2005). Mills (2003) argues that, in addition to being ad hoc, decomposition methods are designed primarily for ease of computation rather than the statistical properties of the data. Proponents of wavelet analysis suggest that wavelets address some of the limitations of ad hoc methods by providing a robust, parameter-free framework (Pen, 1999) for decomposing a time series without prior knowledge of the underlying process (Ramsey, 1999; Gençay et al, 2002); like the classical decomposition method, wavelet-based decomposition preserves the components of the original series being modelled.

![Image](image_url)

Figure 3.4. Trend-cycle, seasonal and irregular components of simulated data computed using the additive form of classical decomposition method (the ‘irregular’ component still has some structure related to the volatility).

### 3.2.2 Formal Approach: Multiresolution Analysis with Wavelets

Methods based on the multiscale wavelet transform provide powerful analysis tools for decomposing time series into coefficients associated with time and a specific frequency band (or scale). Wavelets decompose a time series into several sub-series, and each series is associated with particular time scales. On each scale, the time series is described by wavelet coefficients of approximation and details. It has been argued that the interpretation of features in complex financial time series is made easy by first applying the wavelet transform, and then interpreting individual sub-series (Zhang et al., 2001). The DWT has a number of limitations (Percival & Walden, 2000), two of which affect its suitability as a pre-processing tool for predictive analysis:
i) The length $N$ of the time series processed by the DWT into $J$ levels must be an integer multiple of $2^J$, although most real-world time series are of nondyadic length. To analyse a nondyadic length time series with the DWT, a partial DWT has to be computed, using only a portion of the available data.

ii) Events in the original time series do not have temporal alignment with the corresponding DWT detail ($D_j$) and approximation coefficients ($S_j$) i.e. events at time $t$ in the original series are not associated with coefficients at time $t$ in the $D_j$ and $S_j$ coefficients.

These limitations are addressed by using the maximal overlap DWT (MODWT), a variant of the DWT. The MODWT is derived just like the DWT (section 2.4.2), but without subsampling filtered outputs, using rescaled scaling and wavelet ($h/2^j$) filters, and circular shifting of filters by integer units, rather than dyadic shifts used for the DWT. Similar to the DWT, MODWT-based multiresolution analysis (MRA) for a time series $X_n$ can be defined as:

$$X_t = \sum_{j=1}^{J} D_j + S_j,$$

Eq. (3.3)

where $S_j$ is the wavelet approximation and $D_j$ are the wavelet details. Recall that the set of coefficients $\{D_j\}$ in Eq. (3.3) are expected to capture local fluctuations over the whole period of a time series at each scale; and the set of values $S_j$ provide a “smooth” or overall “trend” of the original signal. Successive “smooth” and “detail” coefficients at different resolutions are obtained using Mallat’s pyramidal algorithm, as discussed in section 2.4.2. Starting from signal $X_n$, smooth and detail coefficients are obtained by iteratively convolving $X_t$ with low- (G) and high-pass (H) filters respectively (Figure 3.5).

Figure 3.5. Mallat’s pyramidal algorithm for wavelet multilevel decomposition.
A 4-level \((J_0=4)\) wavelet transform for the simulated series \(X_i\) (discussed in section 3.2.1, Figure 3.1) is shown below (Figure 3.6). The top panel is the original data, \(X_i\), and other plots are wavelet components of the raw signal. Lower level wavelet decompositions, with \(D_1\) being the lowest level, represent high frequency components. As the wavelet level increases, corresponding coefficients typically become smoother. Starting from \(D_1\), successive components represent highest to lowest frequency component of the original signal, with \(S_1\) representing the “smooth” or lowest frequency component. The additive form of reconstruction in Eq. (3.3) allows us to predict each wavelet sub-series separately and add the individual predictions to generate an aggregate forecast.

![Wavelet Components](image)

Figure 3.6. Simulated time series \((X_i)\) and its wavelet components \(D_1, D_4, S_4\).

The use of wavelets for pre-processing is based on the supposition that wavelets are capable of separating the different components of time serial data. In the following, we illustrate the ability of wavelets to extract seasonal components from noisy time series that are nonstationary in the variance, whilst leaving the underlying components intact.

3.2.2.1 Wavelet-based Extraction of Seasonal Components

The presence of seasonal components in a time series results in positive autocorrelations that are considerably higher for time lags that are integer multiples of the seasonal period than for
other lags. Such components dominate the autocorrelation plot and make it difficult to detect and model the underlying data generation process. In order to reveal other dynamics present in the time series, seasonal components need to be filtered out i.e. the series needs to be deseasonalised without distorting low-frequency components.

The method proposed in this thesis exploits the capability of wavelets to decompose time series into constituent components, prior to the application of a fuzzy model. The method therefore depends on the supposition that wavelets are capable of extracting seasonal and other components from time series. This assertion is examined in this section. Following Gençay et al (2001), we describe simulations that indicate that wavelets extract seasonal components from noisy time series that are nonstationary in the variance, whilst leaving the underlying components intact. The dataset consists of an AR(1) process with periodic components $C_i$, defined by:

$$X_t = 0.95 X_{t-1} + e_t + C_i$$

where

$$C_i = \sum_{j=1}^{4} [3 \sin\{\frac{2\pi}{P_j} t\} + \eta \nu_i].$$

$C_i$ has four periodic components $P_1=2$, $P_2=4$, $P_3=8$, $P_4=16$; $e_t$ and $\nu_i$ are zero mean, unit variance random variables, and $\eta$, the signal-to-noise ratio in each seasonal component, is set to 0.30 in order to mask the periodic components. A 1000-sample realisation of this model (Figure 3.7a) is used in the analysis. In the first part of the experiment, autocorrelograms of the AR(1) model with and without seasonal components are examined and compared to the wavelet smooth obtained from wavelet-filtered data.

![Image](image_url)

Figure 3.7. (a) Time plot of AR(1) model with seasonality components (b) sample autocorrelogram for AR(1) process (solid line) and AR(1) process with seasonality components (dashed line). Adapted from Gençay et al. (2001).
Autocorrelograms of the AR(1) model with \( (X_t) \) and without the seasonal component \( (X_t - C_t) \) (Figure 3.7b) indicate that the presence of seasonal components distorts the autocorrelation structure. The presence of seasonal patterns depresses the autocorrelations, and effectively obscures the persistence observed in the autocorrelations of the aperiodic AR(1) model. ‘Deseasonalisation’ of the periodic time series should result in a filtered series with an autocorrelation structure that is similar to the aperiodic AR(1) model.

The simulated data has periodic components \( P_1, P_2, \ldots, P_4 \) with periods between 2 and 16, and wavelet detail \( D_j \) captures time series dynamics associated with frequencies, \( f \), such that \( 2^{j+1} \leq f \leq 2^j \) i.e. periodic oscillations \( P \) in the range \( 2^j \leq P \leq 2^{j+1} \). Based on the length of the periodic components, a four-level MODWT decomposition on the seasonal data can be used i.e. the data is decomposed into a wavelet smooth, \( S_4 \), and four wavelet details \( D_1, D_2, \ldots, D_4 \). This implies that wavelet details \( D_1 - D_4 \) capture oscillations with periods \( 2 - 32 \), and wavelet smooth \( S_4 \) is expected to be free from periodic components. This is in fact the case: \( S_4 \) has no oscillatory component and is similar to the AR(1) model without seasonal components (Figure 3.8a). The result indicates that the wavelet-based method has been able to isolate the AR model in the presence of stochastic seasonality.

Furthermore, the autocorrelogram of the wavelet smooth is similar to that of the AR(1) model without seasonal components (Figure 3.8b). This indicates that periodic components have been automatically filtered out by the wavelet method, leaving the underlying structure intact, and lends credence to the claim that wavelet-based filtering can be used for the extraction of seasonality in time series data.

![Figure 3.8.](image_url)

Figure 3.8. (a) Time plots of AR(1) model without seasonality components (blue) and wavelet smooth \( S_4 \) (red); (b) sample autocorrelograms for AR(1) process (solid line), AR(1) process with seasonality components (dashed line), and wavelet smooth \( S_4 \) (red dotted line).
The data model used in the previous example assumes stationarity in the variance, with error terms $e_t$ and $\nu_i$, having (approximately) unit variance. However, many real-world data, particularly financial market data, exhibit nonstationarity in the variance. In the second part of the simulations, the ability of wavelets to extract seasonal components in the presence of variance change is examined. The seasonal AR(1) model is modified by introducing variance change between data points 500 and 750. The variance change in the modified AR(1) model, $X_t^*$, can be observed in the shaded portion of the time plot (Figure 3.9a). A wavelet decomposition scheme similar to that used for the constant variance time series was implemented for the modified data in order to obtain the wavelet approximation, $S_4$ (Figure 3.9b).

It can be observed that, in the region with variance change, the plot of the wavelet smooth $S_4$ is noticeably different from that of the aperiodic AR(1) model. This is because, for $S_4$, the variance change has been filtered out and retained in lower frequency components $D_t$ of the wavelet decompositions. This appears to be a distortion of the original signal, since $S_4$ does not maintain high fidelity with $X_t^*$ in the shaded region, unlike what obtains in the model without variance change (Figure 3.8a). However, the temporary effect, distortion or ‘noise’ in the region with variance change is not ‘lost’, it has been captured and preserved in lower scale wavelet components. Importantly, the underlying correlation structure captured by the sample autocorrelograms for both $X_t^*$ and the wavelet smooth $S_4$ are similar (Figure 3.9b): wavelet-based filtering has enabled the isolation of the seasonal component, leaving low-frequency dynamics intact, even in the presence of nonstationarity in the variance.

It can be inferred from these simulations that the use of wavelets to uncover underlying dynamics of data is robust to the presence of noise, periodic components and variance change.

Figure 3.9. (a) Time plots of aperiodic AR(1) model with variance change between 500-750 (blue) and corresponding wavelet smooth $S_4$ (red); (b) sample autocorrelograms for AR(1) process with variance (solid blue line), and wavelet smooth $S_4$ (red dotted line).
3.3 Diagnostics for Time Series Pre-processing

Learning and out-of-sample generalization capabilities are two critical issues in developing fuzzy systems: models that are not sufficiently complex may fail to capture key characteristics in a complicated time series, resulting in underfitting. Conversely, the use of complex models on data with simple structures may lead to overfitting the training set, and poor out-of-sample forecast performance. This is the bias/variance dilemma (Geman et al., 1992; Bishop, 1995).

The goal of pre-processing method diagnostics described in this section is to examine the statistical properties of time series, and only recommend complex pre-processing strategies like wavelet decomposition when data with complex structures are being analysed.

Furthermore, the assumption that fuzzy models are not constrained by nonstationarity has an impact on the analysis of a time series. A number of diagnostic tests to check for stationarity exist, and the application of these tests should be considered before a time series is pre-processed. In this thesis, we use two tests: the partial autocorrelation function (PACF) for checking for nonstationarity in the mean, particularly for autocorrelations at lags of 12 for monthly data; and wavelet-based variance analysis for checking for nonstationarity in the variance. The flowchart (Figure 3.10) shows a comparison of formal and informal approaches to time series pre-processing.

![Flowchart of 'informal' and 'formal' pre-processing methods.](Image)
3.3.1 Testing the Suitability of Informal Approaches

In order to test for the suitability of informal approaches to time series pre-processing, we examine the correlation structure of such time series. The autocorrelation function (ACF) is a statistical tool for examining the correlation structure of time series. The autocorrelation coefficient, $r_s$, of a time series with length $n$ and mean $\bar{x}$, lagged $s$ periods apart, can be defined as:

$$r_s = \frac{\sum_{t=s+1}^{n} (x_t - \bar{x})(x_{t-s} - \bar{x})}{\sum_{t=1}^{n} (x_t - \bar{x})^2}$$

In particular, in order to determine whether to use first or seasonal differencing, we examine the PACF of the time series and consider if the coefficient at lag 12 is significant. Seasonal difference (SD) is computed if the partial autocorrelation coefficient at lag 12 is positive and significant, first difference (FD) is taken otherwise. If the time series is not stationary after computing the seasonal difference, then an additional first difference should be computed (SD+FD). The rules stated above provide a more systematic approach to the selection of pre-processing techniques. For example, consider the time series of a zero mean and unit variance Gaussian random variable and the associated PACF (Figure 3.11). As expected, the coefficient at lag 12 is not significant.

![Figure 3.11. Time plot of random series and corresponding PACF plot. None of the coefficients has a value greater than the critical values (blue dotted line)](image)

As a matter of fact, since this is a random time series, none of the autocorrelation coefficients are significant, and all have values below the critical value (Table 3.1). However, for the simulated series (discussed in section 3.2.1, Figure 3.1), the plot of the partial autocorrelation coefficients indicates that there are significant, positive partial autocorrelations at lags 1, 3, 4, 5, 7, 9, 10, 11 and 12 (Figure 3.12 and Table 3.2).
We are interested in the partial autocorrelations at lags 1 and 12, since the proposed pre-processing selection method recommends seasonal difference only when the coefficient at lag 12 is positive and significant. In this case, the coefficient is positive and significant, indicating that seasonal difference, rather than first difference, will be beneficial. We show in Chapter 4 that, for this simulated time series, seasonal difference is beneficial, and that making the 'right' choice between first and seasonal difference has a significant impact on the forecast accuracy of the fuzzy model.

We emphasize that the test only distinguishes between using of first or seasonal difference, and does not test whether to use polynomial fitting for trend removal, or a 12-month centred MA for the removal of seasonality. Moreover, autocorrelation coefficients have some limitations: they measure only linear relationships (Chatfield, 2004), exhibit instability with small samples (n<30), and are influenced by outliers (Makridakis et al, 1998).

<table>
<thead>
<tr>
<th>Lags</th>
<th>PACF values</th>
<th>Lags</th>
<th>PACF values</th>
<th>Lags</th>
<th>PACF values</th>
<th>Lags</th>
<th>PACF values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0196</td>
<td>6</td>
<td>0.0455</td>
<td>11</td>
<td>-0.029</td>
<td>16</td>
<td>0.0287</td>
</tr>
<tr>
<td>2</td>
<td>-0.0501</td>
<td>7</td>
<td>0.0222</td>
<td>12</td>
<td>0.0766</td>
<td>17</td>
<td>-0.0474</td>
</tr>
<tr>
<td>3</td>
<td>0.0222</td>
<td>8</td>
<td>0.028</td>
<td>13</td>
<td>-0.0109</td>
<td>18</td>
<td>0.0312</td>
</tr>
<tr>
<td>4</td>
<td>-0.0371</td>
<td>9</td>
<td>0.076</td>
<td>14</td>
<td>-0.0251</td>
<td>19</td>
<td>0.0113</td>
</tr>
<tr>
<td>5</td>
<td>-0.0697</td>
<td>10</td>
<td>-0.0087</td>
<td>15</td>
<td>0.0372</td>
<td>20</td>
<td>0.0422</td>
</tr>
</tbody>
</table>

Table 3.1. PACF values at different lags 1-20, and critical values ±0.0885
Table 3.2. PACF values at different lags 1-20 showing positive significant values (boldface) at critical values ±0.0885

<table>
<thead>
<tr>
<th>Lags</th>
<th>PACF values</th>
<th>Lags</th>
<th>PACF values</th>
<th>Lags</th>
<th>PACF values</th>
<th>Lags</th>
<th>PACF values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.9910</td>
<td>6</td>
<td>0.0337</td>
<td>11</td>
<td>0.6093</td>
<td>16</td>
<td>-0.0263</td>
</tr>
<tr>
<td>2</td>
<td>0.0578</td>
<td>7</td>
<td>0.4045</td>
<td>12</td>
<td>0.9881</td>
<td>17</td>
<td>0.0651</td>
</tr>
<tr>
<td>3</td>
<td>0.3569</td>
<td>8</td>
<td>-0.1155</td>
<td>13</td>
<td>-0.3637</td>
<td>18</td>
<td>-0.0420</td>
</tr>
<tr>
<td>4</td>
<td>0.2661</td>
<td>9</td>
<td>0.0997</td>
<td>14</td>
<td>0.0530</td>
<td>19</td>
<td>0.0771</td>
</tr>
<tr>
<td>5</td>
<td>0.1009</td>
<td>10</td>
<td>0.3286</td>
<td>15</td>
<td>0.0268</td>
<td>20</td>
<td>-0.0089</td>
</tr>
</tbody>
</table>

### 3.3.2 Testing the Suitability of Wavelet Pre-processing

Hybrid models that use a combination of wavelets and other time series modelling tools have been reported in the literature. Ho et al. (2001) describe a fuzzy wavelet network (FWN) where wavelet functions are used as the activation function in the hidden layer, and a fuzzy model is used to improve the accuracy of the wavelet sigmoid function. In addition, Thuillard (2001) describe “wavelet networks, wavenets and fuzzy wavenets”, using different combinations of wavelets and soft computing techniques. The scaling function of wavelets is employed to determine membership functions of fuzzy models. Fuzzy systems and wavelets have also been used to model multiscale processes (Zhang et al., 2003), in which data collected at different sampling rates are decomposed using wavelets to facilitate multivariate analysis.

Wavelet analysis has been used for data ‘filtering’ prior to the application of fuzzy systems (Popoola et al., 2004), neural networks (Aussem & Murtagh, 1997; Zhang et al., 2001; Soltani, 2002; Murtagh et al., 2004), and autoregressive (AR) models (Renaud et al., 2003; Renaud et al., 2005). In these studies, models built from wavelet-processed data consistently resulted in better model performance. However, our study on Takagi-Sugeno-Kang (TSK) fuzzy models of time series that exhibit seasonal changes, and structural breaks indicates that, depending on the variance profile of the time series under analysis, models built from wavelet-processed data may underperform compared to models trained on raw data (Chapter 4).

Wavelets are better suited for modelling time series that exhibit local behaviour or structural changes (Percival & Walden, 2000). Also, periods of high volatility that result in variance changes in economic and financial time series occur in localized regions or clusters (Franses & van Dijk, 2000). Time series that exhibit variance changes and volatility clustering require
pre-processing, and wavelet-based pre-processing offers a ‘natural’, parameter-free method for decomposing such time series. Conversely, for time series with homogeneous variance, the use of universal approximation models like fuzzy systems may be appropriate and sufficient; any pre-processing leads to worse results compared to an equivalent analysis carried out using raw data. One possible explanation could be that wavelet pre-processing is well suited for analysing time series with structural breaks and local behaviour. If there are no such discontinuities or local behaviour, then (i) there is no need to use the pre-processing and (ii) there is a possibility that the use of wavelet pre-processing may add artefacts to processed series, thereby worsening the fit.

We propose two methods, which test wavelet coefficients of a time series on a level-by-level basis, for assessing the suitability of wavelet-based pre-processing. First, the (wavelet) variance plot of a time series is used as an exploratory device for graphical assessment of the suitability of wavelet-based pre-processing. Second, a statistical test, based on a method for detecting multiple variance breaks in time series, is used as an indicator as to whether wavelet-based pre-processing is required. The methodology uses formal hypothesis testing to determine a priori whether wavelet pre-processing will improve forecast performance.

### 3.3.2.1 The Wavelet Variance Plot

The wavelet variance decomposes the variance of a time series \( x_t \) on a scale-by-scale basis, thereby replacing ‘global’ variability with variability over (local) scales:

\[
\text{Var}(x_t) = \sum_{j=1}^{\infty} \sigma^2_j(\lambda_j)
\]

where level \( j \) wavelet coefficients are associated with scale \( \lambda_j = 2^{j-1} \), and

\[
\sigma^2_j(\lambda_j) = \frac{1}{2\lambda_j} \text{Var}(w_{j,s})
\]

The wavelet variance plot can be used for visually exploring and detecting any variance changes in time series data, and hence the suitability of wavelet processing for the time series. For example, consider a synthetic time series defined by:

\[
X_t = 3 + 0.95X_{t-1} + \epsilon_t
\]

This represents an AR(1) process, where \( \epsilon_t \) is a zero mean and unit variance Gaussian random variable (Figure 3.13a). The plot shows that the variance of this time series is constant for all values of \( t \). The corresponding wavelet variance, plotted against wavelet scale \( \lambda_s \), is shown in Figure 3.13b. The relationship between the wavelet scale and the variance is approximately linear, indicating that there is no significant variance change across all scales.
When variance change is introduced in the model at $t_{3,500}$ (Figure 3.14a), the linear relationship no longer holds (Figure 3.14b).

The wavelet variance plot reveals that the variance in the time series is not constant, with a structural break noticeable at higher scales. The wavelet variance plot can be used to detect the presence of variance breaks, and this serves as an indicator as to whether wavelet-based data pre-processing will be beneficial. The use of the wavelet variance plot for diagnosing the suitability of wavelets for pre-processing real-world time series is explored in Chapter 4.

### 3.3.2.2 Wavelet-Based Test for Homogeneity of Variance

In the previous section, the use of the wavelet variance plot for detecting variance change was discussed. It was argued that the variance profile of a time series may be used to diagnose its suitability for wavelet-based pre-processing. However, the use of a wavelet variance plot is subjective, and does not easily lend itself to automation, since it requires visual inspection and the necessity for human intervention. In this section, we describe a method inspired by literature on tests for variance homogeneity of time series.
Two properties of the DWT are of particular relevance for this test. First, the variance of a time series is preserved and captured in the variance of its wavelet coefficients (Percival & Walden, 2000). The wavelet variance, obtained by decomposing the variance of a time series on a scale-by-scale basis, can be used to partition and summarize the properties of a time series at different scales. Second, the DWT can effectively dissolve the correlation structure of heavily autocorrelated time series, using a recursive two-step filtering and downsampling method (Gencay et al., 2002). Coefficients resulting from wavelet-filtered data therefore form a near independent Gaussian sequence. These properties form the basis of the test for homogeneity of variance.

The variance preservation and approximate decorrelation properties of the DWT have been used for constructing a statistical test for variance homogeneity in long memory processes (Whitcher et al., 2000). The test depends on the hypothesis that, for a given time series $X_1, \ldots, X_N$, each sequence of wavelet coefficients $W_{ij}$ for $X_i$ approximates samples of zero mean independent Gaussian random variables with variances $\sigma_1^2, \ldots, \sigma_N^2$. The null hypothesis for this test is:

$$H_0 : \sigma_1^2 = \sigma_2^2 = \ldots = \sigma_N^2$$

and alternative hypotheses are of the form:

$$H_1 : \sigma_1^2 = \ldots = \sigma_c^2 \neq \sigma_{c+1}^2 = \ldots = \sigma_N^2$$

where $c$ is an unknown variance change point. The test statistic is based on the use of the normalized cumulative sum of squares, $\eta_c$:

$$\eta_c = \frac{\sum_{j=1}^{c} w_j^2}{\sum_{j=1}^{N} w_j^2}$$

where $w_j$ are scale $j$ DWT coefficients. $\eta_c$ measures variance accumulation in a time series, as a function of time. A plot of the cumulative variance can provide a means of studying the time dependence of the variance of a series: if the variance is stationary over time, i.e. the null hypothesis is not rejected, only small deviations (of $\eta_c$) from zero should be observed, and $\eta_c$ should increase linearly with $c$, at approximately $45^\circ$, with each random variable contributing the same amount of variance. Conversely, if $H_0$ is rejected, (i) relatively larger divergence of $\eta_c$ from zero may exist, and (ii) considerable divergence of the cumulative variance plot from the $45^\circ$ line will occur.
For example, consider a time series $\varepsilon_t$ of $N(10,1)$ random variables with constant variance $\sigma^2_{1:N} = 1.0$ (Figure 3.15). A plot of the cumulative variance for the first level wavelet component of this series shows a maximum deviation (from zero) of about 0.06, indicating, as expected, that the variance is stationary.

![Figure 3.15. Time plot of random variables with homogeneous variance (top) and associated normalized cumulative sum of squares (bottom).](image)

The variance structure can be altered by introducing variance changes at $n_{400}$ and $n_{700}$ such that $\sigma^2_{1:400} = 1.0$, $\sigma^2_{401:700} = 3.0$, and $\sigma^2_{701:1024} = 1.0$ (Figure 3.16). Here, $\eta_c$ increases linearly at approximately 45° until it is at location $n_{400}$ (Figure 3.16 - bottom). Subsequently, considerable divergence of the variance plot from the 45° line is observed. From location $n_{700}$, when the variance is 1.0, $\eta_c$ again varies linearly with time. Also, the cumulative variance plot shows that $\eta_c$ has a significant deviation from zero, with a maximum value of about 0.40, almost a sevenfold increase from the previous value of 0.06. This is an indication that the variance in the series is not homogenous.

![Figure 3.16. Time plot of random variables with variance change at $n_{400}$ and $n_{700}$ (top) and associated normalized cumulative sum of squares (bottom).](image)
The test statistic, $D$, for detecting inhomogeneity of variance measures variance accumulation in a time series as a function of time. $D$ is defined as the maximum vertical deviation from the 45° line, with critical levels of $D$ at 1%, 5% and 10% significance levels under $H_0$ empirically generated using Monte Carlo simulations, based on 10,000 replicates (Whitcher et al., 2002). $D$ is defined in terms of its components $D^+$ and $D^-:

$$D^+ = \max_{1 \leq c < N-1} \left( \frac{c}{N-1} - \eta_c \right),$$

Eq. (3.4)

and

$$D^- = \max_{1 \leq c < N-1} \left( \eta_c - \frac{c-1}{N-1} \right),$$

Eq. (3.5)

then

$$D = \max[D^+, D^-]$$

Eq. (3.6)

3.3.2.3 Test Algorithm

Our method tests candidate time series for homogeneity of variance, and selects only time series that exhibit variance change(s) for wavelet pre-processing (Algorithm 1). We are interested in $j=1$ level of wavelet coefficients (step iiia), which is reported to be the most sensitive to the presence of variance change in a series (Whitcher et al., 2000). The effect of zero padding on variance change is eliminated by only selecting coefficients that are related to the actual time series (step iiib).

i. Given a time series $\{x_i\}, i=1...N$, create
   training $\{x_a\}, a=1...T, T<N$
   test $\{x_b\}, b=T+1...N$ data sets
ii. If $T > m*2^j$, where $m, j \in \mathbb{Z}$
    Add $k$ zeros such that
    $T + k = m*2^j$;
iii. Compute partial DWT (order $J$) of $\{x_a\}$,
   (a) Retain only coefficients for $j=1$
   (b) Select first $N_{j=1} = (N/2)$ coefficients;
   (c) Discard boundary coefficients;
iv. Test for suitability of wavelet pre-processing
   Calculate $D$, using Eqs. (3.4)-(3.6)
   If $D >$ critical value at 5% significance level,
   $H_0$ is not rejected
   Then use raw data to generate fuzzy model
   Else
   $H_0$ is rejected
   Then use wavelet-processed data to generate fuzzy model

Algorithm 1. Testing the suitability of wavelet pre-processing.
3.4 Fuzzy-Wavelet Model for Time Series Analysis

The proposed framework provides single-step time series predictions based on components obtained from multiscale decomposition of a time series. The approach can be regarded as a way of decomposing a large problem into smaller and specialized ones, with each sub-problem analysed by an individual fuzzy model. The method comprises the diagnosis, pre-processing and model configuration phases (Figure 3.17).

In the diagnosis phase, the time series to be analysed is tested for suitability of wavelet-based pre-processing (section 3.3.2). If deemed suitable, the second phase, in which the components of the time series are generated using MODWT-based multiresolution analysis, is executed. Otherwise, the second stage is omitted. The diagnosis phase provides a measure of 'intelligence' in the system - the suitability of the time series for wavelet-based processing is evaluated before proceeding to the pre-processing stage. In the pre-processing phase, shift invariant wavelet components are obtained from raw data, using the MODWT. In the model configuration stage, fuzzy models are generated from raw data, or wavelet components.

The following packages have been used in this research:

i) Waveslim package (Whitcher, 2005) is utilized in the testing phase.

ii) MODWT analysis is carried out using the Wavelet Methods for Time Series Analysis (WMTSA) toolkit for Matlab (Percival & Walden, 2000).

iii) Subtractive clustering is performed by using the Fuzzy Logic toolbox of Matlab™.
3.4.1 Pre-processing: MODWT-based Time Series Decomposition

The schematic representation of the proposed framework is presented (Figure 3.18). In the pre-processing stage, the time series is decomposed into different scales using the MODWT. Since our intention is to make one-step-ahead predictions, we should perform the MODWT in such a way that the wavelet coefficients (for each level) at time point $t$ should not be influenced by the behaviour of the time series beyond point $t$. Thus, we must perform the MODWT incrementally where a wavelet coefficient at a position $t$ is computed using samples at time points less than or equal to, but never beyond point $t$. This will give us the flexibility of dividing the wavelet coefficients for training and testing and making one-step-ahead predictions in the same way as we would for the original signal.

![Figure 3.18. Schematic representation of the wavelet/fuzzy forecasting system.](image)

$D_1, \ldots, D_j$ are wavelet coefficients, $S_j$ is the signal "smooth".

To accomplish this, we make use of the time-based à trous filtering scheme proposed in Shensa (1992). Given time series $\{X_t: t = 1, \ldots, n\}$, where $n$ is the present time point, we perform the steps detailed below (Algorithm 2), following Zhang et al. (2001).

i. For index $k$ sufficiently large, (we use $k = 10$)
   Compute MODWT transforms on $\{X_t: t = 1, \ldots, n\}$.

ii. For $J$ resolution levels,
   Retain $D_{1,k}, D_{2,k}, \ldots, D_{J,k}, S_{J,k}$, for the $k$th time point only

iii. If $k < n$,
   Set $k = k + 1$;
   Go to Step 1.

iv. The summation of $D_{1,k}, D_{2,k}, \ldots, D_{J,k}, S_{J,k}$ gives $X_{t,k}$, as indicated in Eq. (3.3).

Algorithm 2. Time-based à trous filtering scheme for MODWT.

Other implementation issues that have to be considered in the use of DWT-based pre-processing for time series analysis include (i) selecting the wavelet family and (ii) dealing
Chapter 3

with boundary conditions. The MODWT is deemed to be less sensitive to the choice of wavelet functions (Gençay et al., 2002), and we have used the Daubechies $D(4)$ basis function as the mother wavelet in our method. In addition, we have used reflection to address boundary conditions.

### 3.4.2 Model Configuration: Subtractive Clustering Fuzzy Model

During model configuration, coefficients from each wavelet scale are divided into in-sample training and validation sets, and out-of-sample test sets. Different fuzzy models $f_{i,w}$ are automatically generated (Algorithm 3) from the training data, where $j$ and $w$ are the decomposition level and window size, respectively. For each decomposition level, an optimal model is selected by checking the models’ performance on the validation data set, which is assumed to be representative of the characteristics of the underlying process.

**Algorithm 3. Subtractive clustering (adapted from Chiu, 1997).**

The selected optimal fuzzy model $f_i$ is used to provide single step forecasts for each wavelet component: for time series $X_i$, decomposed into $J$ level wavelet components with details $D_j$ at
different levels of decomposition and smooth $S_J$ fuzzy models for single-step forecast are generated for each wavelet component. This results in $J+1$ fuzzy models:

\[
\begin{align*}
  f_1 &: D_{1_{1-n}} , D_{1_{1-n+1}}, \ldots , D_{1_{m}} \rightarrow D_{1_{m+1}} \\
  f_2 &: D_{2_{1-n}} , D_{2_{1-n+1}}, \ldots , D_{2_{m}} \rightarrow D_{2_{m+1}} \\
  & \ldots \\
  f_J &: D_{J_{1-n}} , D_{J_{1-n+1}}, \ldots , D_{J_{m}} \rightarrow D_{J_{m+1}} \\
  f_{J+1} &: S_{J_{1-n}} , S_{J_{1-n+1}}, \ldots , S_{J_{m}} \rightarrow S_{J_{m+1}}
\end{align*}
\]

In the third and final stage, single step forecasts for all the wavelet components are combined to give the next step forecast for time series $X$:

\[X_{t+1} = D_{1_{m+1}} + D_{2_{m+1}} + \ldots + D_{J_{m+1}} + S_{J_{m+1}}\]

### 3.5 Summary

In this chapter, we discussed the universal approximation property of TSK fuzzy models in relation to the analysis of real-world nonstationary time series. We highlighted the limitations of global models in analysing data with complex local behaviour, discussed the importance of data complexity in generating fuzzy models, and emphasised the need for data pre-processing.

Many pre-processing methods exist in the literature. However, pre-processing methods are typically selected arbitrarily, depend on the expertise of the analyst, and the availability of knowledge about the underlying data generating process. In this study, we propose the use of shift invariant wavelet transforms as data pre-processing tools. The use of wavelet analysis not only eliminates the need for an ad hoc approach to data pre-processing, as currently practised, but also removes the need for knowledge about the data generating process. Moreover, wavelet analysis is deemed to be a parameter free method, since parameters are not imposed on the time series as is the case for autoregressive models.

In our scheme, time series predictions are generated by fuzzy models, which are built from components obtained from multiscale decomposition of a time series. This method generally results in improved forecast accuracy compared to models generated from raw data. However, whilst forecast performance may improve with wavelet-based data pre-processing, this is not true for all data sets. There are cases where wavelet-based pre-processing leads to worse results compared to an equivalent analysis carried out using raw data. The method described in this chapter therefore incorporates a measure of intelligence, whereby an automatic method
for detecting variance breaks in time series is used as an indicator as to whether or not wavelet-based pre-processing is required.

In the next chapter, we investigate the effects of data pre-processing on the forecast performance of the subtractive clustering fuzzy model by comparing the performance on raw and pre-processed time series. Also, simulations and experiments carried out to evaluate the proposed pre-processing method are described, and a detailed discussion of the results obtained is provided.
Chapter 4

Simulations and Evaluation

4.1 Introduction

In chapter 2, time series analysis methods, with a focus on fuzzy models, were discussed. We highlighted criticisms of fuzzy models related to improving forecast performance and the universal approximation property. In chapter 3, we argued that suitably chosen data pre-processing techniques can improve the forecast accuracy of fuzzy models, and presented a wavelet-based method for pre-processing data for fuzzy systems. In this chapter, we describe simulations carried out to evaluate our approach. The following issues are addressed:

i) We carry out simulations that examine the effects of data pre-processing on fuzzy systems. Specifically, we investigate the effect of data pre-processing on the forecast performance of subtractive clustering fuzzy systems.

ii) Different pre-processing strategies have been used in the literature on soft computing techniques like neural networks. Typically, selection of pre-processing techniques is carried out in ad hoc, trial-and-error manner. This has resulted in inconsistent, apparently conflicting results on the effect of pre-processing on such models. We evaluate our proposal that a systematic or ‘intelligent’ data pre-processing selection strategy, using well-established statistical methods, is beneficial.

iii) Finally, we evaluate the proposed fuzzy-wavelet framework for time series analysis. The results indicate that wavelet pre-processing improves forecast accuracy for time series that exhibit variance changes and other complex local behaviour. Conversely, for time series that exhibit no significant structural breaks or variance changes, fuzzy models trained on raw data perform better than hybrid fuzzy-wavelet models. Limitations of the proposed scheme are also discussed.
4.2 Rationale for Experiments

In line with established research practice, we endeavour to assess the method on simulated data with known characteristics. These characteristics are such that they mimic relevant properties observed in real-world data. For example, to investigate the effect of preprocessing on real-world trend and seasonal time series, a simulated autoregressive time series with trend and seasonal components is tested, and then the method(s) are applied to real-world data sets. In addition, whilst there is a wide variety of real-world datasets in the literature to choose from, we have limited our choice of data to a subset with a mix of what might be some interesting characteristics – trend and seasonal components, discontinuities and/or variance changes. In the following, we provide an overview of datasets and evaluation methods used in our experiments.

4.2.1 Simulated Time Series

The simulated time series is based on the seasonal ARIMA (SARIMA) model, ARIMA\((p,d,q)(P,D,Q)_s\), where \((p,d,q)\) and \((P,D,Q)\) respectively represent the nonseasonal and seasonal part of the model, and \(s\) is the length of the season. This model incorporates characteristics of interest: increasing trend i.e. nonstationarity in the mean, and the presence of seasonal variation. Since the real world series used in our analysis (described in section 4.2.2) comprises monthly data, we have set the length of the season, \(s=12\) for the simulated time series. The synthetic data is the SARIMA model \((1, 0, 0)(0, 1, 1)_12\) with a non-seasonal AR term, a seasonal MA term, and one seasonal difference, used in Chatfield (2004). This model is given by

\[
(1 - \phi B)(1 - B^{12})X_i = (1 + \theta B^{12})\epsilon_i,
\]

where \(\epsilon_i\) are random variables with \(\mu = 0\) and \(\sigma^2 = 1\). We have set \(\phi = 0.4\) and \(\theta_1=0.7\). The time series generated by this model exhibits strong seasonal patterns, and nonstationarity in the mean (Figure 4.1).

![Figure 4.1. Simulated SARIMA model \((1, 0, 0)(0, 1, 1)_12\) time series.](image-url)
4.2.2 Real-World Time Series

Real-world data set used in our analysis comprises two groups of monthly time series (Table 4.1) used by Zhang and Qi (2005). These are six US Census Bureau (USCB) retail sales data and four industrial production series from the Federal Reserve Board (FRB). Each series records the price or production volume of a good/service stored on a monthly basis. Monthly series are used since they exhibit stronger seasonal patterns than quarterly series. These series are characterized, in varying degrees, by trend and seasonal patterns, as well as discontinuities. FRB fuels and USCB clothing (Figure 4.2) time series are prototypes of each class of time series.

<table>
<thead>
<tr>
<th>#</th>
<th>Data Sets</th>
<th>Sample number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>FRB Durable Goods</td>
<td>660</td>
</tr>
<tr>
<td>2</td>
<td>FRB Consumer goods</td>
<td>384</td>
</tr>
<tr>
<td>3</td>
<td>FRB Total production</td>
<td>660</td>
</tr>
<tr>
<td>4</td>
<td>FRB Fuels</td>
<td>576</td>
</tr>
<tr>
<td>5</td>
<td>USCB Book stores</td>
<td>120</td>
</tr>
<tr>
<td>6</td>
<td>USCB Clothing stores</td>
<td>120</td>
</tr>
<tr>
<td>7</td>
<td>USCB Department stores</td>
<td>120</td>
</tr>
<tr>
<td>8</td>
<td>USCB Furniture stores</td>
<td>120</td>
</tr>
<tr>
<td>9</td>
<td>USCB Hardware stores</td>
<td>120</td>
</tr>
<tr>
<td>10</td>
<td>USCB Housing Start</td>
<td>516</td>
</tr>
</tbody>
</table>

4.2.3 Evaluation Method

Various error measures have been proposed in the literature, and a critical survey of error metrics has been provided (see, for example, Hyndman and Koehler, 2005). According to Makridakis et al (1998: 42), 'standard' statistical measures of forecast accuracy include the mean error (ME), the mean square error (MSE), and the mean absolute error (MAE). The MSE, and its variant, the root MSE (RMSE), are particularly useful when comparing various methods on the same set of data. However, the RMSE statistic is sensitive to the dimension of the data, and the presence of outliers, and is consequently not recommended for forecast accuracy evaluation (Armstrong, 2001). For example, USCB clothing data has values in the order of 10E+04, while the values of FRB data are in the order of 10E+02 (Figure 4.2).
(R)MSE values for USCB clothing data will therefore appear deceptively higher relative to that of FRB data. The mean absolute percentage error (MAPE), a variant of the MAE, is generally recommended for evaluating forecast accuracy (Bowerman et al, 2004), provided that all the data sample have non-zero values (Makridakis et al, 1998).

\[
\text{MAPE} = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{X_i - X_i^{'}}{X_i} \right|
\]

In some experiments, multiple simulations are carried out and the average MAPE is generated (section 4.3). Although the average MAPE provides a measure of the central tendency of errors, it provides no information about the variability of errors. In such cases, box-plots are

---

Figure 4.2. (a) USCB clothing stores data exhibits strong seasonality and a mild trend; (b) FRB fuels data exhibits nonstationarity in the mean and discontinuity at around the 300\(^{th}\) month.

In this thesis, we compare methods across different data sets, which have different dimensions, and have non-zero values. Consequently, the metric of choice is the MAPE. Given a test set of length \(n\), single-step predictions, \(X_i^{'}\), are evaluated against the target (original) series, \(X_i\), and the MAPE statistic, \(E_{\text{MAPE}}\) is defined as:

\[
E_{\text{MAPE}} = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{X_i - X_i^{'}}{X_i} \right|
\]
used to provide a graphical view of the spread of errors. Outliers are excluded from the computation of the mean and standard deviation by using Tukey’s outlier filter (Hoaglin et al., 1983).

4.3 Informal Pre-processing for Fuzzy Models

In this section, we report experiments conducted to investigate the effects of data pre-processing on forecast performance of subtractive clustering TSK fuzzy models. The models were trained and validated with three differently pre-processed data sets: (i) data detrending using first difference (FD); (ii) deseasonalisation with seasonal difference (SD); and (iii) both seasonal and first difference (SD+FD). In all the experiments, forecasts on pre-processed data were converted back to the original time series before computing prediction errors.

Following Chiu (1994), Angelov and Filev (2004), we have used a cluster radius of 0.5 to build first-order TSK fuzzy models from training data, without iterative optimisation. In order to reduce the effect of window size selection on model performance, we have used window sizes between 1 and 40 in all cases, with out-of-sample test set comprising the last one year (12 months), as in Zhang and Qi (2005). Reported results are based on the mean and standard deviation computed over all the window sizes. For each data set, forecast performance on raw data is compared to results obtained from pre-processed data.

Recall that, according to our framework for testing informal pre-processing methods, partial autocorrelation functions (PACF) are used as a pre-processing method selection for nonstationary time series: seasonal difference (SD) is computed if the partial autocorrelation coefficient at lag 12 is positive and significant, first difference (FD) is taken otherwise. If the time series is not stationary after computing the seasonal difference, then an additional first difference should be computed (SD+FD). Our intuition is that these rules provide a more objective basis for choosing pre-processing techniques. In this section, for each time series, we use PACF-based recommendations to select a pre-processing technique, and compare the results to actual (best) results generated by any of the FD, SD, SD+FD methods.

To evaluate the PACF-based recommendations vis-à-vis actual (best) results, two criteria are examined:

i) accuracy, defined as the minimum average MAPE;

ii) robustness, defined as the minimum (most compact) inter-quartile range.
The robustness of the models developed from data pre-processed in a particular manner provides an indication of the stability or reliability of the model in the problem space: a method that results in low average MAPE but has wide error variability, i.e. poor robustness, is unreliable. An ideal model will exhibit both high accuracy and robustness. In practice, models need to achieve a balance between high accuracy, or learning capability, and robustness, or generalisation ability. This is related to the classic bias-variance trade-off (Geman et al., 1992; Bishop, 1995).

### 4.3.1 Results and Discussion

#### 4.3.1.1 Simulated Data

We first discuss results for simulated data (Figure 4.1). In this time series, the PACF at lag 12 is positive and significant (Figure 4.3), and the PACF-based recommendation is to use seasonal difference (SD) to pre-process this time series.

![PACF plot for simulated time series](image)

Figure 4.3. PACF plot for simulated time series.

Next, we compare this recommendation to empirical results for the simulated data, which are presented in Table 4.2. The results indicate that, generally, data pre-processing appears to be beneficial. In particular, models derived from both SD and SD+FD data pre-processing techniques, which involve seasonal differencing, provide markedly better accuracy compared to that obtained from raw data, confirming the recommendation of our method.

<table>
<thead>
<tr>
<th>Pre-Processing Method</th>
<th>MAPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>3.51 ± 2.00</td>
</tr>
<tr>
<td>FD</td>
<td>2.84 ± 0.99</td>
</tr>
<tr>
<td>SD</td>
<td>0.60 ± 0.06</td>
</tr>
<tr>
<td>SD+FD</td>
<td>0.59 ± 0.04</td>
</tr>
</tbody>
</table>
This is to be expected, since the seasonal ARIMA model used to generate the data has a seasonal difference component, and removal of this component is important. On the other hand, FD pre-processing results in worse performance relative to SD and SD+FD methods because the simulated data does not have a non-seasonal difference component. Note that whilst a time plot of the series indicates the presence of a linear trend (Figure 4.1), which in an ad hoc framework suggests taking a first difference, the results indicate that (i) the type of pre-processing method applied affects prediction accuracy; (ii) the choice between seasonal difference (SD) and first difference (FD) is non-trivial, and ad-hoc use of first difference (or any other pre-processing technique) may worsen forecast performance.

To examine the robustness of the model generated by the recommended pre-processing method, the box plots for these results (Figure 4.4) are plotted on a log scale (the arrow at the top indicates that outliers exist outside the range shown). The error spreads for models derived from both R and FD are considerable, indicating poor robustness of models derived from these data. Error variability for SD and SD+FD are compact, and reasonably similar. This signifies that fuzzy models generated from SD and SD+FD are robust i.e. reliable. Moreover, the medians for R and FD are undesirably high, while medians for SD and SD+FD data are similar, and have low values.

\[
\begin{array}{c}
\text{Log MAPE} \\
\hline
10^0 \\
10^{-1} \\
10^{-2} \\
10^{-3} \\
10^{-4} \\
10^{-5} \\
\end{array}
\]

Figure 4.4. Model error for raw and pre-processed simulated data.

4.3.1.2 Real-world Data

In the following, we discuss results obtained using ten real-world time series described in section 4.2.2. Forecast errors obtained for the best (and worst) results for each time series, across all pre-processing methods, are reported (Table 4.3). Here again, the results indicate that data pre-processing is generally beneficial: in six out of the ten time series, the use of raw time series results in the worst MAPE values. SD+FD, SD and FD pre-processing respectively result in minimal MAPE in six, three and one case(s); SD and FD pre-processing
each result in the maximum MAPE in two cases. There was no case where SD+FD gave the worst result.

We observe that none of the pre-processing techniques provides consistently superior performance relative to other techniques, across all time series. This corroborates the assertion that the type of pre-processing method applied affects prediction accuracy, and more pre-processing (SD+FD) does not necessarily result in improved model performance. Also, ad hoc selection of pre-processing method, for example, taking seasonal difference simply because monthly data is being analysed, may result in worse performance. For time series with trend, like most of the time series analysed, simply taking the first difference to ‘make the data stationary’ results in the best forecast in only one out of the ten series.

Table 4.3. Minimum and maximum MAPE for each of the ten series and the pre-processing technique resulting in minimum error.

<table>
<thead>
<tr>
<th>Data Sets</th>
<th>MAPE (%)</th>
<th>‘Best’ Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Best</td>
<td>Worst</td>
</tr>
<tr>
<td>1. USCB Furniture</td>
<td>4.23 ± 1.26</td>
<td>5.22 ± 0.51 b</td>
</tr>
<tr>
<td>2. FRB Fuels</td>
<td>1.42 ± 0.33</td>
<td>4.16 ± 4.26</td>
</tr>
<tr>
<td>3. USCB Department</td>
<td>2.95 ± 0.25</td>
<td>5.06 ± 2.70 *</td>
</tr>
<tr>
<td>4. FRB Hardware</td>
<td>1.90 ± 0.45</td>
<td>4.10 ± 0.28</td>
</tr>
<tr>
<td>5. FRB Durable Goods</td>
<td>2.37 ± 0.35</td>
<td>3.11 ± 0.93</td>
</tr>
<tr>
<td>6. FRB Consumer goods</td>
<td>0.93 ± 0.17</td>
<td>2.49 ± 0.92</td>
</tr>
<tr>
<td>7. FRB Total production</td>
<td>0.72 ± 0.13</td>
<td>2.06 ± 2.08 b</td>
</tr>
<tr>
<td>8. USCB Book Store</td>
<td>6.32 ± 0.68</td>
<td>10.44 ± 3.78 *</td>
</tr>
<tr>
<td>9. USCB Clothing</td>
<td>3.16 ± 0.35</td>
<td>4.92 ± 0.09</td>
</tr>
<tr>
<td>10. USCB Housing Start</td>
<td>6.06 ± 1.48</td>
<td>8.35 ± 5.70</td>
</tr>
</tbody>
</table>

*FD is worst performing method; bSD is worst performing method; all other (worst) results are for raw data.

Recall that, for each time series, four different data types were used to generate fuzzy models - raw (R), first differenced (FD), seasonal differenced (SD) and seasonal and first differenced (SD+FD) data. Next, we compare actual (best) results to the PACF-based recommendations of our method for these ten time series (Table 4.4). In all but one case (series 1 i.e. USCB Furniture), the PACF-recommended method results in models that provide the best (score = 1) or ‘almost best’ (score = 2) for both accuracy and robustness metrics; and in two cases, pre-processing techniques resulting in the best accuracy and robustness are recommended.
It is not in all cases that the PACF-based method recommends pre-processing methods that result in the best accuracy or robustness. However, the approach provides consistently helps in avoiding pre-processing methods that result in the worst performance in terms of both accuracy and robustness i.e. the method was able to address the classic bias-variance trade-off (Geman et al., 1992). For instance, although the recommended pre-processing method (SD+FD) for series 1 does not result in the minimum mean error, it results in the model with the most compact error range (discussed later in this section). We note that, whilst partial autocorrelation functions are routinely utilized in model identification for statistical techniques, to our knowledge, this work represents the first attempt to use it for selecting pre-processing methods for soft computing models.

### Table 4.4. PACF-based recommendations and actual results

<table>
<thead>
<tr>
<th>Data Sets</th>
<th>PACF-based Recommendation</th>
<th>Evaluation Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Accuracy</td>
</tr>
<tr>
<td>1. USCB Furniture</td>
<td>SD+FD</td>
<td>3*</td>
</tr>
<tr>
<td>2. FRB Fuels</td>
<td>FD</td>
<td>2</td>
</tr>
<tr>
<td>3. USCB Department</td>
<td>SD+FD</td>
<td>2</td>
</tr>
<tr>
<td>4. USCB Hardware</td>
<td>SD+FD</td>
<td>2</td>
</tr>
<tr>
<td>5. FRB Durable Goods</td>
<td>SD+FD</td>
<td>1</td>
</tr>
<tr>
<td>6. FRB Consumer goods</td>
<td>SD+FD</td>
<td>1</td>
</tr>
<tr>
<td>7. FRB Total production</td>
<td>FD</td>
<td>2</td>
</tr>
<tr>
<td>8. USCB Book Store</td>
<td>SD</td>
<td>2</td>
</tr>
<tr>
<td>9. USCB Clothing</td>
<td>SD+FD</td>
<td>1</td>
</tr>
<tr>
<td>10. USCB Housing Start</td>
<td>FD</td>
<td>2</td>
</tr>
</tbody>
</table>

*Scores indicative of performance compared to actual (best) results: 1(best) and 4(worst).

In order to better understand why specific pre-processing techniques result in poor forecast performance, we investigate the characteristics of the time series. We discuss three cases where each pre-processing technique - FD, SD, FD+SD - was appropriate. For each of the cases, we computed the partial autocorrelation function, PACF, and use PACF plots to investigate the properties of the time series, and box plots to examine the robustness of different pre-processing techniques.

**Best result with seasonal differenced data**

The best accuracy for series 3 (USCB Department) is from models developed from seasonal differenced data. The raw series has a high positive value at lag 12, and comparatively lower
values at other lags (Figure 4.5). This suggests that the application of seasonal difference is beneficial, which is indeed the case. Note that whilst a time plot of the series indicates the presence of a mild linear trend that in an ad hoc framework may suggest taking a first difference, the PACF indicates that the coefficient at lag 1 is not significant; the use of first differenced data results in the worst performance.

![Figure 4.5. Time plot of series 3 (USCB Department) and corresponding PACF plot for raw data.](image)

Next, we examine the error distribution of the processed data (Figure 4.6). The inter-quartile range for FD processed data is similar to that of raw data, indicating that an ad hoc application of first difference is not beneficial in this case. This supports the argument that specific pre-processing methods may be unsuitable for some time series. Conversely, relative to raw data, models built from SD and SD+FD pre-processed data show lower spread, or variability, of the error measure. SD processed data has the most compact error range, or best model robustness. This suggests that SD is an effective pre-processing strategy for this time series, confirming the PACF-based recommendation.

![Figure 4.6. Model error for raw and pre-processed USCB Department.](image)

**Best result with first differenced data**

Series 1 (USCB Furniture) has the best accuracy using FD data. The highest positive value for
the PACF is at lag 1, indicating that first difference may be beneficial (Figure 4.7). Seasonal
difference computed on the data results in the worst average error forecast (Table 4.3). This,
perhaps, highlights a limitation of the PACF-based method, with the recommended SD+FD
pre-processing resulting in a comparatively low accuracy score.

![Figure 4.7. Time plot of series 1 (USCB Furniture) and corresponding PACF plot for raw data.](image)

However, although FD results in the best (accurate) models, errors generated by models
developed from FD data have the widest inter-quartile range – poor robustness (Figure 4.8),
which is undesirable. In contrast, SD processed data results in a robust model. The PACF-
based recommendation framework enables us to select, in this case, a pre-processing method
that generates robust fuzzy models.

![Figure 4.8. Model error for raw and pre-processed series 1 (USCB Furniture) data.](image)

**Best result with seasonal and first differenced data**

Series 5 (FRB Durable Goods) has the best accuracy with fuzzy models developed from
SD+FD data. The PACF has high positive values at lags 1 and 12, indicating that seasonal
and first difference may be beneficial (Figure 4.9). Although the order in which seasonal and
first differences are applied makes no difference, it is recommended that seasonal difference
be applied first, since resulting series may not require a further first difference (Makridakis et
al, 1998). It is necessary to check the PACF plot after seasonal differencing. In this case, after the seasonal difference has been taken, the PACF still shows a large positive coefficient at lag 1 (Figure 4.9), necessitating a further first difference.

The error distribution for this time series (Figure 4.10) confirms that SD+FD is a good choice of pre-processing method: SD+FD, as recommended by our method, has the best robustness. Although SD and FD separately do not result in consistently low error values, a combination of both methods results in the best performance in terms of both accuracy and robustness.
4.3.1.3 Fuzzy Rule Clusters of Real-World Data

In this section, we examine the rule structure of models derived from pre-processed and unprocessed data. In particular, we are interested in finding out if, for a given time interval (window size), more rule clusters are needed to characterise the problem space when processed data is used to generate fuzzy models. This provides an indication as to whether pre-processing reduces data complexity, which can be captured using simple fuzzy models i.e. models with less rule clusters compared to models generated from raw data.

To accomplish this, and enable fair comparison across the different pre-processing methods, we select a window size of 12, which represents an annual cycle for monthly data. Note that an input window size of 12 implies, for a multiple input single output (MISO) system, that the data has 12+1 dimensions. Since only three dimensions can be displayed at a time, in all scatter plots, we limit ourselves to illustrating with two of the input dimensions, and the output dimension. Setting the cluster radius to 0.5, we use the selected window size to construct fuzzy models for each time series, and examine the rule clusters generated using Algorithm 3 (section 3.4.2).

Fuzzy Rule Clusters for Raw Data

Consider series 5 (FRB Durable Goods) in Table 4.4. For this time series, the MAPE result for raw data obtained with a model having a window size of 12 is 4.72%, and the subtractive clustering algorithm automatically generates four rule clusters to characterise the problem space (Figure 4.11). These clusters are located where the data are more concentrated i.e. near the origin of the input axes, and data points that are farthest from the origin are not located in any of the clusters generated.

Figure 4.11. (a) 3-dimensional scatter plot of series 5 using raw data (b) Four rule clusters automatically generated to model the data.
**Fuzzy Rule Clusters for First Differenced (FD) Data**

For FD data, a window size of 12 gives a MAPE of 1.94%. Using FD data, we observe that the data distribution has been significantly altered, with data concentrated in the middle of the hypercube (Figure 4.12). In this case, a single rule cluster, rather than four, is generated. Although a single cluster is used to characterise the problem space, the MAPE is significantly lower than that of the model generated from raw data (4.72%), which have four rule clusters. This illustrates the benefit of using a *suitable* pre-processing method. In this case, there is the added advantage of reduced complexity, since only one rule cluster is needed.

![Figure 4.12. (a) 3-dimensional scatter plot of series 5 using FD data (b) One rule cluster automatically generated to model the data.](image)

**Fuzzy Rule Clusters for Seasonal Differenced (SD) Data**

Next, SD processed data is used to generate a fuzzy model (Figure 4.13). The model with a window size of 12 gives a MAPE of 2.59%.

![Figure 4.13. (a) 3-dimensional scatter plot of series 5 using SD data (b) Six rule clusters automatically generated to model the data.](image)
The processed data in this case are not as concentrated compared to FD processed data, although more concentrated compared to raw data. Due to the sparse nature of the data, six rule clusters are required in the fuzzy model. Even then, not all data are in clusters, and this model has a worse MAPE, compared to FD processed data. This corroborates the assertion made earlier in this section that, although pre-processing is generally beneficial, some methods are better than others on specific data sets, and indiscriminate use of pre-processing methods may lead to degraded model accuracy. The SD model however offers a significant improvement in terms of error reduction, albeit at the cost of increased model complexity, relative to the model generated from raw data.

**Fuzzy Rule Clusters for Seasonal and First Differenced (SD+FD) Data**

For SD+FD data, a window size of 12 results in a MAPE of 2.07%. SD+FD processed data (Figure 4.14a) appear to be more concentrated than SD data (Figure 4.13a), but not as concentrated as FD data (Figure 4.12a). A single rule cluster is generated for SD+FD data, similar to what obtains for FD data. Although just one cluster is used for SD+FD data, compared to six clusters for SD data, the error for the SD+FD model is lower than that generated using SD data. This suggests that model complexity, in terms of the number of generated rule clusters, does not necessarily translate to a lower MAPE. In addition, the single cluster generated using FD processed data results in marginally lower error compared to SD+FD data.

![Figure 4.14](image)

Using a similar experimental method, i.e. window size of 12, the other nine time series were tested. A summary of the number of rule clusters generated for each of the ten time series is provided (Table 4.5). For a given time series, each value presented in the table represents the ratio of the number of rule clusters using a specific pre-processing method, to the maximum...
number of clusters generated using any (R, FD, SD, SD+FD) data. A value equal to 1.0 means that the specific pre-processing method results in the greatest number of rule clusters. For example, for FRB Durable Goods (series 5), SD results in the highest number of rule clusters (six clusters), and has a value of 1.0; Raw, FD and SD+FD respectively result in four, one and one rule clusters, and have values of 0.7, 0.2 and 0.2 respectively. The associated score (superscript) indicates relative forecast performance. For series 5, FD results in the best result (score = 1) and R results in the worst result (score = 4).

It can be observed that, overall, SD and SD+FD results in higher number of rule clusters: SD+FD and SD generate the maximum rule clusters in five and six cases respectively. Conversely, FD and R result in the maximum number of rule clusters only in one and two cases respectively. This suggests that, for some time series, particular types of pre-processing may result in more complex fuzzy models, with higher number of rule clusters.

### Table 4.5. Ratio of the number of rule clusters using specific pre-processing method to the maximum number of clusters generated using any data, and corresponding MAPE forecast performance.

<table>
<thead>
<tr>
<th>Data Sets</th>
<th>R</th>
<th>FD</th>
<th>SD</th>
<th>SD+FD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. USCB Furniture</td>
<td>0.44</td>
<td>0.31</td>
<td>0.82</td>
<td>1.03</td>
</tr>
<tr>
<td>2. FRB Fuels</td>
<td>1.04</td>
<td>0.82</td>
<td>0.33</td>
<td>0.31</td>
</tr>
<tr>
<td>3. USCB Department</td>
<td>0.34</td>
<td>0.13</td>
<td>1.01</td>
<td>1.02</td>
</tr>
<tr>
<td>4. USCB Hardware</td>
<td>0.44</td>
<td>0.33</td>
<td>1.02</td>
<td>1.01</td>
</tr>
<tr>
<td>5. FRB Durable Goods</td>
<td>0.74</td>
<td>0.21</td>
<td>1.03</td>
<td>0.22</td>
</tr>
<tr>
<td>6. FRB Consumer goods</td>
<td>0.03^4</td>
<td>0.14</td>
<td>0.02^1</td>
<td>1.0^2</td>
</tr>
<tr>
<td>7. FRB Total production</td>
<td>0.94</td>
<td>0.73</td>
<td>1.0^2</td>
<td>0.31</td>
</tr>
<tr>
<td>8. USCB Book Store</td>
<td>0.62</td>
<td>0.2^4</td>
<td>1.0^4</td>
<td>0.7^3</td>
</tr>
<tr>
<td>9. USCB Clothing</td>
<td>0.33</td>
<td>0.12</td>
<td>1.04</td>
<td>1.01</td>
</tr>
<tr>
<td>10. USCB Housing Start</td>
<td>1.0^3</td>
<td>1.0^4</td>
<td>0.2^2</td>
<td>0.3^1</td>
</tr>
</tbody>
</table>

^1-4Score indicative of forecast performance: 1(best) and 4(worst).

However, we observe that, while R generally results in less complex models, in none of the time series did R result in the most accurate model. In fact, R results in the worst forecast (score = 4) in six cases, and ‘almost worst’ (score = 3) in three cases. Conversely, SD+FD results in the best forecast (score = 1) in three cases, ‘almost best’ (score = 2) in two cases, and never resulted in a worst (score = 4) performing model. Instructively, complex models with high number of rule clusters do not necessarily provide the best forecast performance: in all
but two instances (series 3 and 4), models with small cluster numbers provide the best or 'almost best' forecast performance. Also, no single pre-processing method consistently results in the best model. The implication is that pre-processing methods need to be matched to the time series under analysis. The following is a summary of inferences that can be made on the usage of data pre-processing methods from these experiments:

(i) generally, data pre-processing appears to be beneficial, although unsuitable methods may result in more complex fuzzy models;

(ii) specific data pre-processing techniques 'match' or are more suitable for particular time series;

(iii) model complexity does not necessarily result in improved accuracy, and suitably pre-processed data can result in less complex but more accurate fuzzy models.

4.3.2 Comparison with Naïve and State-of-the-Art Models

So far, we have presented results based on the average of 40 window sizes, in order to discount the effect of window size selection on forecast performance. In this section, we present a comparison of results obtained using the subtractive clustering fuzzy system to those obtained using:

i. a naïve random walk model, which assumes a Gaussian (white-noise) distribution

ii. state-of-the-art models reported by Zhang and Qi (2005), using artificial neural networks (ANN) and the ARIMA method, and Taskaya-Temizel and Ahmad (2005), using time delay neural networks and AR models

Following the data partition used in the state-of-the-art models, validation data comprises the last 12 months of in-sample data, while the remaining data are used for model training. Out-of-sample test data consists of the last 12 months' data of each data set. In order to facilitate fair comparison, rather than using the average error from 40 different window sizes, as was done in the previous section, we test window sizes between 1 and 20 on the validation set, and lags with minimum error are used on the test set to obtain single-step forecasts. This is similar to the method used in Taskaya-Temizel and Ahmad (2005). Also, RMSE errors are reported here, since Taskaya-Temizel and Ahmad (2005) only report RMSE errors.
4.3.2.1 Comparison with Naïve Random Walk Model

The random walk hypothesis asserts short-term unpredictability of future time series values. In order to investigate the short-term predictability of economic time series, forecast performance of a naïve random walk model is compared to the prediction of the fuzzy model trained on raw data (Table 4.6). Note that the naïve model used in our analysis defines the single-step future value as the additive combination of a Gaussian (white noise) variable to the current value of the series.

<table>
<thead>
<tr>
<th>Data Sets</th>
<th>Naïve</th>
<th>TSK Fuzzy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. FRB Durable Goods</td>
<td>7.79</td>
<td>3.25</td>
</tr>
<tr>
<td>2. FRB Consumer goods</td>
<td>3.33</td>
<td>2.69</td>
</tr>
<tr>
<td>3. FRB Total production</td>
<td>2.55</td>
<td>2.29</td>
</tr>
<tr>
<td>4. FRB Fuels</td>
<td>2.74</td>
<td>2.35</td>
</tr>
<tr>
<td>5. USCB Housing Start</td>
<td>12.61</td>
<td>11.13</td>
</tr>
<tr>
<td>6. USCB Hardware</td>
<td>134.37</td>
<td>71.64</td>
</tr>
<tr>
<td>7. USCB Book Store</td>
<td>439.10</td>
<td>163.73</td>
</tr>
<tr>
<td>8. USCB Furniture</td>
<td>307.18</td>
<td>215.29</td>
</tr>
<tr>
<td>9. USCB Clothing</td>
<td>3201.06</td>
<td>537.69</td>
</tr>
<tr>
<td>10. USCB Department</td>
<td>6588.58</td>
<td>715.84</td>
</tr>
</tbody>
</table>

The results indicate that the subtractive clustering fuzzy model consistently provides significantly better forecast performance relative to the naïve random walk model. This implies that short-term predictability is possible for economic time series. We note that, as opposed to high frequency financial time series, economic time series on which our studies are based are aggregated i.e. monthly data where noisy, possibly random components arising from daily or other short-term fluctuations have been averaged out.

4.3.2.2 Comparison with State-of-the-Art Models

Forecast performance of all the models generated with pre-processed data, as well as error reductions due to data pre-processing for the ARIMA-NN and fuzzy models, are reported (Table 4.7). Note that the TDNN (time-delay neural network) and AR-TDNN methods reported in Taskaya-Temizel and Ahmad (2005) inherently feature data pre-processing.
Table 4.7 Comparison of RMSE on AR-TDNN\textsuperscript{a}, TDNN\textsuperscript{b} (Taskaya-Temizel and Ahmad, 2005) ARIMA\textsuperscript{c}, ARIMA-NN\textsuperscript{d} (Zhang and Qi, 2005), and fuzzy models\textsuperscript{e}.

<table>
<thead>
<tr>
<th>Data Sets</th>
<th>AR-TDNN\textsuperscript{a}</th>
<th>TDNN\textsuperscript{b}</th>
<th>ARIMA\textsuperscript{c}</th>
<th>ARIMA-NN\textsuperscript{d}</th>
<th>TSK Fuzzy\textsuperscript{e}</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. FRB Durable Goods</td>
<td>1.48</td>
<td>2.91</td>
<td>5.61</td>
<td>3.63 (44*)</td>
<td>2.85 (12*)</td>
</tr>
<tr>
<td>2. FRB Consumer goods</td>
<td>0.98</td>
<td>1.07</td>
<td>3.96</td>
<td>0.68 (57)</td>
<td>0.97 (64)</td>
</tr>
<tr>
<td>3. FRB Total production</td>
<td>0.83</td>
<td>1.07</td>
<td>8.94</td>
<td>0.85 (48)</td>
<td>0.82 (64)</td>
</tr>
<tr>
<td>4. FRB Fuels</td>
<td>1.52</td>
<td>1.64</td>
<td>1.62</td>
<td>0.81 (59)</td>
<td>1.35 (43)</td>
</tr>
<tr>
<td>5. USCB Housing Start</td>
<td>-</td>
<td>-</td>
<td>5.18</td>
<td>4.23 (74)</td>
<td>5.78 (48)</td>
</tr>
<tr>
<td>6. USCB Hardware</td>
<td>25.39</td>
<td>35.70</td>
<td>100.71</td>
<td>49.17 (73)</td>
<td>31.56 (56)</td>
</tr>
<tr>
<td>7. USCB Book Store</td>
<td>99.33</td>
<td>91.51</td>
<td>98.17</td>
<td>88.74 (73)</td>
<td>101.66 (38)</td>
</tr>
<tr>
<td>8. USCB Furniture</td>
<td>173.03</td>
<td>173.10</td>
<td>124.44</td>
<td>99.45 (79)</td>
<td>172.05 (20)</td>
</tr>
<tr>
<td>9. USCB Clothing</td>
<td>381.76</td>
<td>372.50</td>
<td>519.60</td>
<td>315.43 (82)</td>
<td>378.99 (30)</td>
</tr>
<tr>
<td>10. USCB Department</td>
<td>726.56</td>
<td>628.70</td>
<td>1005.41</td>
<td>975.55 (55)</td>
<td>726.26 (-1)</td>
</tr>
</tbody>
</table>

\*percentage RMSE reduction compared to models developed from raw data

First, we compare the forecast performance of the conventional ARIMA method (column 4), which includes data pre-processing in model identification, to the fuzzy models generated using pre-processed data (column 6). For the ten time series analysed, soft computing methods developed from pre-processed data generally result in better forecast performance than the corresponding ARIMA model: in all but three time series, fuzzy models developed from pre-processed data result in better forecast performance than the ARIMA model; similarly, the hybrid ARIMA-ANN method results in better accuracy than the ARIMA method. Next, we compare the two soft computing techniques – TDNN (column 3) and fuzzy models (column 6). Generally, the fuzzy model provides comparable forecast performance to the ANN model, with the fuzzy model resulting in marginally better forecast accuracy in five out of nine cases reported (results for only nine time series are reported in Taskaya-Temizel and Ahmad, 2005).

Moreover, we observe that the more sophisticated AR-TDNN model (column 2) does not consistently result in better forecast performance, compared to the simpler TDNN model (column 3). The AR-TDNN model provides marginal accuracy improvements in six cases, but results in worse accuracy in three of the nine cases reported. This corroborates our assertion that more sophisticated hybrids do not necessarily result in better models, and, at best, provides incremental improvements in forecast accuracy (section 2.3.3 in Chapter 2). In contrast, pre-processing methods typically result in significant improvements in the forecast performance of soft computing models. A comparison between accuracy improvements
recorded for sophisticated models, as opposed to models developed from pre-processed data, is presented (columns 5 and 6 in Table 4.7 – percentage error reduction in parentheses). The results indicate that, in all but one case (for the fuzzy model), neural networks and fuzzy models generated from pre-processed data result in significantly improved forecast performance, relative to models generated from raw data.

4.4 Formal Wavelet-based Pre-processing for Fuzzy Models

In the previous section, we discussed the effects of data pre-processing on fuzzy model forecast performance. The results indicate that the use of appropriate pre-processing techniques results in improved forecast accuracy. On the other hand, inappropriate choice of pre-processing techniques typically results in degraded performance. The choice of pre-processing techniques was influenced by our knowledge of the data model for the case of simulated data, and the possibility of having 12-month seasonality in the case of real-world economic time series. Consequently, seasonal differencing with a season length of 12 was used for both data sets. However, the availability of information about the data generating process is not always guaranteed and, as we discussed in the previous section, an ad hoc selection strategy may result in degraded forecast performance.

In this section, we describe simulations carried out using multiscale wavelet analysis as a pre-processing tool, as discussed in Chapter 3. We use wavelets to analyse real-world economic time series, and provide a test to determine the suitability of wavelet pre-processing. This is followed by a critique of the wavelet-based method.

4.4.1 Fuzzy-Wavelet Model for Time Series Analysis

Previously, we discussed the need for pre-processing of data prior to the application of the subtractive clustering fuzzy model (section 4.3). We have also argued that wavelet-based filtering is an effective pre-processing method, since it is capable of automatically extracting the underlying non-seasonal structure of a seasonal time series, even in the presence of variance change (section 3.2.2.1 in Chapter 3). In this section, we discuss forecast results obtained using wavelet-based pre-processing for fuzzy models (4.4.1.1); the effect, on forecast performance, of fuzzy models generated from each wavelet component (4.4.1.2); and the observed deterioration in forecast performance for fuzzy-wavelet models of some time series (4.4.1.3).
4.4.1.1 Experiments and Results

To evaluate the proposed method, we carried out simulations using ten experimental dataset described in section 4.2.2. The subtractive clustering fuzzy model was used to build first-order TSK models from training data, without iterative optimisation. Following Chiu (1994), Angelov and Filev (2004), we use a cluster radius, $r_a$ of 0.5, neighbourhood radius of 1.5 $r_u$, thresholds for acceptance and rejection of candidate cluster centres, respectively, of 0.5 and 0.15 of the first potential $P_1$. Each of the ten time series was decomposed into five wavelet resolution levels using the $a$ trous filtering scheme discussed in section 3.4.2. The fuzzy system was trained separately on each wavelet sub-series (resolution) to generate single step forecasts. Time lags between 1 and 20 are tested on the validation set, and lags with minimum MAPE are used on the test set to obtain single-step forecasts. Forecast performances of the fuzzy model using wavelet-processed data are obtained on all ten series. This is compared to results obtained from (i) raw, unprocessed data; (ii) ad hoc (informal) pre-processing methods discussed in the previous section. In all cases, MAPE values were computed after converting the pre-processed series back to the original. The results on fuzzy models generated from raw, ad hoc-, and wavelet-processed data are obtained on all ten series. This is compared to results obtained from (i) raw, unprocessed data; (ii) ad hoc (informal) pre-processing methods discussed in the previous section. In all cases, MAPE values were computed after converting the pre-processed series back to the original. The results on fuzzy models generated from raw, ad hoc-, and wavelet-processed data are presented in Table 4.8. Note that, here, MAPE results are presented (recall that RMSE was reported in Table 4.7 to facilitate comparison with results reported in the literature).

Table 4.8. Comparison of MAPE on raw, informal (ad hoc) and formal (wavelet processed) data using fuzzy clustering model

<table>
<thead>
<tr>
<th>#</th>
<th>Data Sets</th>
<th>Pre-processing Method</th>
<th>Error Reduction (%)**</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>None (Raw)</td>
<td>Ad hoc*</td>
</tr>
<tr>
<td>1</td>
<td>FRB Durable Goods</td>
<td>2.37</td>
<td>1.95</td>
</tr>
<tr>
<td>2</td>
<td>FRB Consumer goods</td>
<td>2.26</td>
<td>0.76</td>
</tr>
<tr>
<td>3</td>
<td>FRB Total production</td>
<td>1.94</td>
<td>0.59</td>
</tr>
<tr>
<td>4</td>
<td>FRB Fuels</td>
<td>1.89</td>
<td>1.03</td>
</tr>
<tr>
<td>5</td>
<td>USCB Book Store</td>
<td>8.43</td>
<td>6.00</td>
</tr>
<tr>
<td>6</td>
<td>USCB Clothing</td>
<td>4.26</td>
<td>2.90</td>
</tr>
<tr>
<td>7</td>
<td>USCB Department</td>
<td>2.45</td>
<td>3.06</td>
</tr>
<tr>
<td>8</td>
<td>USCB Furniture</td>
<td>3.88</td>
<td>3.32</td>
</tr>
<tr>
<td>9</td>
<td>USCB Hardware</td>
<td>5.05</td>
<td>1.97</td>
</tr>
<tr>
<td>10</td>
<td>USCB Housing Start</td>
<td>6.14</td>
<td>3.74</td>
</tr>
</tbody>
</table>

*The best results for ad hoc processed data; **comparison of error reduction between models built from raw and wavelet-processed data
Generally, forecast accuracy for wavelet-processed data is better for FRB data than for USCB data, similar to results based on ad hoc pre-processing methods in section 4.3.2. Also, compared to models generated from raw data, model performance improved with wavelet pre-processing in seven out of the ten time series studied, while there was degradation in performance for the remaining three time series (last column in Table 4.8).

The best results from handcrafted ad hoc pre-processing are better than results obtained for raw data in all but one (series 7) case, and wavelet-based models are better than ad hoc methods in three cases. Note that the results reported for ad hoc methods are the best out of three different pre-processing methods – first difference, seasonal difference and both seasonal and first difference. If wavelet-based results are compared to individual ad hoc pre-processing methods, then the fuzzy-wavelet method results in better forecast performance in five cases compared to models based on each of first- and seasonal-differenced data, and four cases compared to models based on both seasonal and first differenced data. This indicates that, considered separately, ad hoc pre-processing methods do not necessarily result in better forecast accuracy than wavelet-based models.

It can be observed from the results that models trained on wavelet-processed data do not necessarily exhibit superior performance, relative to models trained on raw data: the fuzzy-wavelet model performs worse in three of the ten series, compared to the fuzzy-raw model. Why is it that, for these three series, the hybrid model performs worse than the fuzzy-raw model? This issue is discussed in section 4.4.1.3.

4.4.1.2 Effect of Each Wavelet Component on Forecast Performance

The performance of each fuzzy model \( f_j (j = 1, 2, \ldots, J+1) \), derived from wavelet detail \( D_j \) or wavelet smooth \( S_j \), relative to the aggregate performance of all the fuzzy models, is examined in this section. As earlier described, five wavelet resolution levels were computed i.e. \( J = 5 \), resulting in five wavelet details \( (D_1 - D_5) \) and one wavelet smooth \( (S_5) \) for each time series. This implies that the fuzzy-wavelet model for each time series required a set of six fuzzy models.

When the fuzzy model from each wavelet component is separately evaluated, there is significant performance degradation – the MAPE of single and aggregate fuzzy model forecasts differ by almost two orders of magnitude (except for the wavelet smooth, \( S_5 \), where the errors are of the same order). This is expected, because the wavelet smooth, \( S_5 \), captures the underlying low-frequency dynamics of the time series, and the exclusion of the forecast from the fuzzy model generated from this component results in the performance deterioration...
observed. An alternative approach to evaluate the effect of fuzzy model is to observe the forecast performance when the contribution from $f_j$ is excluded from the aggregate forecast performance of the fuzzy-wavelet scheme (Table 4.9).

Table 4.9. Aggregate forecast performance (MAPE) when the contribution of fuzzy models generated from each wavelet component is excluded

<table>
<thead>
<tr>
<th>Data Sets</th>
<th>Wavelet component excluded</th>
<th>None (D1-D5, A5)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>D1</td>
<td>D2</td>
</tr>
<tr>
<td>1 FRB Durable Goods</td>
<td>2.01</td>
<td>2.08</td>
</tr>
<tr>
<td>2 FRB Consumer goods</td>
<td>1.36</td>
<td>1.53</td>
</tr>
<tr>
<td>3 FRB Total production</td>
<td>1.21</td>
<td>1.18</td>
</tr>
<tr>
<td>4 FRB Fuels</td>
<td>1.14</td>
<td>1.62</td>
</tr>
<tr>
<td>5 USCB Book Store</td>
<td>9.66</td>
<td>15.36</td>
</tr>
<tr>
<td>6 USCB Clothing</td>
<td>10.84</td>
<td>13.53</td>
</tr>
<tr>
<td>7 USCB Department</td>
<td>6.25</td>
<td>8.85</td>
</tr>
<tr>
<td>8 USCB Furniture</td>
<td>4.36</td>
<td>5.66</td>
</tr>
<tr>
<td>9 USCB Hardware</td>
<td>4.55</td>
<td>7.05</td>
</tr>
<tr>
<td>10 USCB Housing Start</td>
<td>3.73</td>
<td>5.10</td>
</tr>
</tbody>
</table>

Again, it can be observed that, in all cases, the elimination of fuzzy models derived from the wavelet smooth $S_s$ results in significantly degraded aggregate forecast performance. To further explore this, we investigate the contribution of the wavelet smooth to the energy profile of an exemplar time series, FRB Durable goods time series (Figure 4.15).

Figure 4.15. (a) Cumulative energy profile of 5-level wavelet transform for FRB Durable goods time series (b) A closer look at the energy localisation in $S_s (t = 0, 1, \ldots, 27)$. 

96
The energy profile provides a summary of energy accumulation in the signal, with time (Walker, 1999). The energy profile of the time series indicates that most of the energy of the signal is localised in the wavelet smooth component, which accounts for 99.6% of the total energy of the signal. This perhaps explains why fuzzy models derived from this component have such significant impact on the accuracy of the aggregate forecast.

Furthermore, we observe that, generally, the elimination of forecasts derived from wavelet components results in worse forecast performance for relatively long time series (all FRB series and USCB Housing start series). This indicates that all such wavelet components capture inherent characteristics in the time series and fuzzy models generated from the wavelet components add value to the aggregate prediction. On the other hand, for shorter-length series (all USCB time series with length of 120, except USCB Housing start), the use of more wavelet scales results in marginal (<15%) degradation of forecast performance of fuzzy models. This indicates that the decomposition level needs to be appropriately matched to the length of the time series, and that fuzzy models derived from higher level components, in this case $D_3$, component, may result in degraded aggregate performance.

We note that, although the use of a subset of fuzzy models does not generally result in performance improvement for the economic time series studied, this may not be the case for 'noisy' high-frequency financial time series, where low level wavelet components are deemed to isolate high frequency 'noise' in the original signal. In such cases, the use of a subset of fuzzy models derived from wavelet components (excluding models derived from components capturing high-frequency noise in the data), may be beneficial.

4.4.1.3 Performance Deterioration in Fuzzy-Wavelet Model

In this section, we discuss the observed deterioration in performance of the hybrid fuzzy-wavelet model for series 6, 7 and 8 (Table 4.8). Recall that, using the MODWT, the variance of a time series is preserved and captured in the variance of the wavelet coefficients (section 3.4.1). Thus, the wavelet variance, obtained by decomposing the variance of a time series on a scale-by-scale basis, can be used to partition and summarize the properties of a time series at different scales. Wavelet variance plots for the three best performing wavelet-processed time series (series 2, 4, 10 in Table 4.8) and the three worst performing wavelet-processed time series (series 6, 7, 8) are presented in Figure 4.16.

It can be observed that for series 6, 7 and 8 (right column of Figure 4.16), the variance exhibits an approximately linear relationship over all the time scales i.e. the variance structure is homogeneous with respect to time scale. This means that there are no significant structural
breaks or changes in these time series. Conversely, for series 2, 4 and 10, the variance structure is not homogeneous, indicating the presence of structural breaks and local behaviour. Notice how the wavelet variances in the left column of Figure 4.16 fluctuate with the time scale, indicating variance breaks, whereas wavelet variances in the right column are much more stable over different time scales.

Figure 4.16. Multiscale wavelet variance plots for wavelet-processed data showing best (left column) and worst (right column) performing series.

One possible explanation for the observed results could be that wavelet pre-processing is well suited for analysing time series with structural breaks and local behaviour. The ability of the MODWT to decompose the sample variance of a time series on a scale-by-scale basis is beneficial for forecasting, since each of the wavelet sub-series characterizes some local behaviour of the original signal. Hence, modelling each sub-series separately and combining individual predictions results in superior aggregate forecasts.

Conversely, if there are no discontinuities or local behaviour in the time series, then (a) there is no need to use wavelet-based pre-processing and universal approximators like fuzzy models are appropriate and sufficient (b) there is a possibility that the use of wavelet pre-processing may result in a complicated (and overfitted) model with low in-sample error but high out-of-sample generalisation capability. This is another case of the bias-variance dilemma (Geman et al, 1992; Bishop, 1995). In the next section, we use the presence of variance breaks as an indicator as to the suitability of wavelet-based pre-processing.
4.4.2 Testing the Suitability of Wavelet Pre-processing

In this section, the test algorithm (described in section 3.4.1) is used to test the suitability of wavelet-based pre-processing on the ten time series. Tests for homogeneity of variance in the ten series signify that five of the series have homogeneous variance, while the other five are characterized by one or more variance changes. Therefore, using inhomogeneity of variance in the time series as the criterion for selecting time series suitable for wavelet-based processing, five of the series are selected as requiring wavelet pre-processing (column 2 in Table 4.10). In comparison, empirically determined actual (best) results using both raw and wavelet-processed data indicate that seven out of the ten series benefit from wavelet-based processing (column 3). The best results obtained from using raw (F-Raw) and wavelet-based processing (F-W), and recommended pre-processing methods using the proposed algorithm, are reported. In all but two (shown in boldface in Table 4.10) of the ten cases under consideration, the proposed method correctly identifies time series that are best suited for wavelet-based pre-processing.

<table>
<thead>
<tr>
<th>Data Sets</th>
<th>Algorithm Recommendation Method</th>
<th>Actual (Best) Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. FRB Durable Goods</td>
<td>F-W</td>
<td>F-W</td>
</tr>
<tr>
<td>2. FRB Consumer goods</td>
<td>F-W</td>
<td>F-W</td>
</tr>
<tr>
<td>3. FRB Total production</td>
<td>F-W</td>
<td>F-W</td>
</tr>
<tr>
<td>4. FRB Fuels</td>
<td>F-W</td>
<td>F-W</td>
</tr>
<tr>
<td>5. USCB Book Store</td>
<td>F-Raw</td>
<td>F-W</td>
</tr>
<tr>
<td>6. USCB Clothing</td>
<td>F-Raw</td>
<td>F-Raw</td>
</tr>
<tr>
<td>7. USCB Department</td>
<td>F-Raw</td>
<td>F-Raw</td>
</tr>
<tr>
<td>8. USCB Furniture</td>
<td>F-Raw</td>
<td>F-Raw</td>
</tr>
<tr>
<td>9. USCB Hardware</td>
<td>F-Raw</td>
<td>F-W</td>
</tr>
<tr>
<td>10. USCB Housing Start</td>
<td>F-W</td>
<td>F-W</td>
</tr>
</tbody>
</table>

4.4.2.1 Wavelet Variance Profile of Time Series

In order to better understand these results, and the effect of the variance structure of time series on its suitability for wavelet-based processing, we examine the wavelet variance...
profiles of all ten series studied. Plots of the wavelet variances of the ten series are presented in Figure 4.17.

Figure 4.17. Multiscale wavelet variance for time series plotted on a log scale. Plots 1-4 and 10 indicate inhomogeneous variance structure; 5-9 exhibit homogeneous structure, though with noticeable discontinuities between scales 1 and 2 for plots 5 and 9.

The numbering on the plots corresponds to specific time series in Table 4.10. The plots show that series 6, 7 and 8 exhibit homogeneous variance structure with respect to time. This suggests that there are no significant structural breaks or changes in these time series. Hence, universal approximators like fuzzy models by themselves are appropriate and sufficient for
modelling the behaviour of such simple structures. Wavelet-based pre-processing results in a significantly complicated model, with more variance and less generalization capability for out-of-sample test data. This, perhaps, explains the deterioration in performance for the fuzzy-wavelet model as compared to the fuzzy-raw model for these three time series.

Conversely, for five of the remaining seven series (Figure 4.17, nos 1-4, and 10), the variance structure is not homogeneous, indicating the presence of structural breaks and local behaviour. For these series, wavelet-based pre-processing helped in improving forecast performance. These results are consistent with the observation in the literature that wavelets are better suited for data with significantly varied behaviour across various time scales (Gençay et al., 2002).

For the two cases (Figure 4.17, nos 5 and 9), where the method failed to correctly prescribe the best processing method, the variance profiles are similar, with fairly homogeneous variance in scales 2-5, and noticeable discontinuity between scales 1 and 2. The inability of the method to detect inhomogeneity in these time series, i.e. acceptance of the null hypothesis of constant variance, is a type II error, and may be addressed by increasing the power of the test (Sendur et al., 2005; Barrow, 2006).

4.4.2.2 Comparison of Wavelet and Power Transformed Data

It can be argued that, if wavelet pre-processing shows better performance for time series with nonstationarity in the variance, then (i) according to Occam’s razor, one should apply relatively simpler pre-processing methods for stabilizing the variance of time series, rather than wavelets. Such methods include logarithmic and square root transformations, which are special cases of Box-Cox power transformations (see section 2.1.2); (ii) the effects on forecast performance for these transformations should be similar to those observed for wavelet-processed data. To further evaluate the fuzzy-wavelet method, we compare the prediction performance of fuzzy models generated from (Box-Cox) power transformed data and wavelet-processed data (Table 4.11). The maximum likelihood method was used to estimate the transformation parameter ($\lambda$) for power transformations, and MAPE values were computed after converting transformed series back to the original.

We observe that models developed from wavelet decomposed data exhibit better performance for all time series, except USCB bookstore and housing start, where models from power transformed data are marginally better. Thus, for the time series analysed, wavelet processing was better able to deconstruct the variance structure. Furthermore, the results show strikingly similar effects on forecast performance, relative to raw data, for both power transformed and
wavelet processed data: better results in series with variance change (except series 2), and worse results in series 6, 7 and 8, where there is variance homogeneity. This corroborates the assertion that wavelets are beneficial due to variance inhomogeneity in some of the data.

Table 4.11. Comparison of Forecast Performance (MAPE) of Fuzzy Models Derived from Wavelet and Box-Cox Transformed Data (Worse Results Relative to Raw Data shown in boldface)

<table>
<thead>
<tr>
<th>Data Sets</th>
<th>Pre-processing Method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>None (Raw)</td>
</tr>
<tr>
<td>1. FRB Durable Goods</td>
<td>2.37</td>
</tr>
<tr>
<td>2. FRB Consumer goods</td>
<td>2.26</td>
</tr>
<tr>
<td>3. FRB Total production</td>
<td>1.94</td>
</tr>
<tr>
<td>4. FRB Fuels</td>
<td>1.89</td>
</tr>
<tr>
<td>5. USCB Book Store</td>
<td>8.43</td>
</tr>
<tr>
<td>6. USCB Clothing</td>
<td>4.26</td>
</tr>
<tr>
<td>7. USCB Department</td>
<td>2.45</td>
</tr>
<tr>
<td>8. USCB Furniture</td>
<td>3.88</td>
</tr>
<tr>
<td>9. USCB Hardware</td>
<td>5.05</td>
</tr>
<tr>
<td>10. USCB Housing Start</td>
<td>6.14</td>
</tr>
</tbody>
</table>

4.4.3 Critique of the Fuzzy-Wavelet Model

4.4.3.1 Model Configuration: Increased Complexity

One of the consequences of using the fuzzy-wavelet framework is that a greater number of fuzzy models have to be developed than is the case if raw data is used: if \( J \)-level wavelet decomposition is carried out according to (Eq. 3.12), \( J+1 \) fuzzy models need to be developed. This results in a system with significantly more rules than the corresponding model generated from raw data. For example, the fuzzy model generated from FRB durable goods time series has four rule clusters. To illustrate the increase in rule complexity resulting from using wavelet pre-processing, we consider the same time series, and examine the rule structure for fuzzy models generated from wavelet-processed data.

A 3-level MODWT transform was generated for this time series (Figure 4.18), and fuzzy models generated for each of the components, following the method described in section 3.4. Typically, the number of clusters generated depends on the level \( J \) of decomposition used – the higher the value of \( J \), the higher the number of rule clusters generated per component, and the higher the total number of rule clusters. Here, with \( J=3 \), and a window size of 12, one,
two, three and six rule clusters are respectively generated for wavelet components $D_1, D_2, D_3,$ and $S_3$ (Figure 4.19). This results in a total of 12 rule clusters, compared to four (4) rule clusters generated for the raw time series.

However, an advantage of this is that it is possible to have a system that provides rules that can be interpreted in terms of time scales, rather than just presenting rules that represent the global picture. This functionality might be of use to, say, market traders, where investors participate with different timescales: there are traders that have an investment horizon of just a few days, even a few hours, while there are those with more long term investment horizons. By providing rules that are associated with specific time scales, the use of wavelets assures that rules matching the investment horizons of different investors can be generated.

Moreover, the number of rules generated in a fuzzy-wavelet model is not a $J+1$ multiple of that generated from a single, ‘global’ fuzzy model. In the example above, the use of raw data results in four rule clusters. With $J=3$, a $J+1$ multiple would result in 16 rules. However, as shown in the figure above, only 12 rule clusters are generated. This is because, wavelets decompose the problem space so that rules are localized to a particular time scale, and the number of clusters or rules required to model each wavelet component is not the same as the
number required for the raw, unprocessed data. Instructively, half of the rule clusters (six clusters) generated by the fuzzy-wavelet model is due to the wavelet smooth, $S_3$. This suggests that it may be beneficial to characterize the wavelet smooth using a simple linear model, rather than a fuzzy model.

![Figure 4.19](image)

**Figure 4.19.** Scatter plot of series FRB durable goods series and associated rule clusters for $D_1$ (a); $D_2$ (b); $D_3$ (c); and $S_3$ (d).

### 4.4.3.2 Hypothesis Testing: Type II Errors

Another limitation of the fuzzy-wavelet scheme is that, as stated in section 4.4.2, the formal hypothesis testing method used for detecting variance homogeneity may be affected by type II errors, where a false null hypothesis is not rejected because of insufficient evidence. Recall that, using formal hypothesis testing, our method was able to correctly diagnose the suitability of wavelet-based processing in eight out of ten cases. In the following, we examine the reason for the failure to recommend the correct pre-processing method for two of the ten time series.

Three time series are considered, where the hypothesis test: (i) correctly detects homogeneity of variance; (ii) correctly detects variance inhomogeneity and (iii) fails to detect variance
inhomogeneity (Figure 4.20). It can be observed that, where the time series is characterised by variance homogeneity i.e. there is an approximately linear relationship across all scales, with no variance breaks (Figure 4.20a), the hypothesis test is able to correctly diagnose that wavelet pre-processing is not suitable. Similarly, where there is variance inhomogeneity, with breaks at scales 2, 4 and 6 (Figure 4.20b), the test correctly recommends wavelet-based pre-processing. However, for the third case, the wavelet variance profile presents a mixed picture: it indicates both variance inhomogeneity (break at scale 2) and homogeneous variance at higher scales (Figure 4.20c). The resultant insufficient evidence of variance inhomogeneity perhaps explains the inability of the hypothesis test to reject the false null hypothesis, although empirical results indicate that sufficient inhomogeneity exists for wavelet-based pre-processing to be beneficial.

Figure 4.20. Wavelet variance profiles of time series where hypothesis test: (a) correctly detects homogeneity of variance; (b) correctly detects variance inhomogeneity and (c) fails to detect variance inhomogeneity.

4.4.3.3 Window Size Selection: Trial-and-Error Approach

The fuzzy models described in this thesis are configured using a trial-and-error approach to window size selection. This is because the determination of appropriate window sizes for
fuzzy models is still an outstanding research issue, although window size selection may be an important factor in getting improved forecast performance. This means that it may be possible to obtain significantly improved results if an appropriate window size is used for the fuzzy-wavelet scheme.

4.5 Summary

In this chapter, we have investigated the effects of pre-processing time series, which act as an input to subtractive clustering TSK fuzzy models. The results indicate that, whilst informal pre-processing methods are generally beneficial, none of the strategies adopted consistently provides improvements in forecast performance as measured by MAPE. Also, the use of data pre-processing does not necessarily result in more complex fuzzy models. We have proposed the use of a systematic approach, based on partial autocorrelation functions, for selecting the most appropriate method for stabilizing a time series, prior to the application of a fuzzy model. The proposed scheme generally leads to a method that pre-processes the series such that a balance of accuracy and robustness are obtained.

Wavelet pre-processing has been used to improve the forecasting capability of typical universal approximators like neural networks. We have extended the scope of these studies to subtractive clustering fuzzy systems, and have evaluated our fuzzy-wavelet framework for time series analysis. The results obtained on carefully selected time series (USCB and FRB series) indicate that time series that have homogeneous variance structures do not require (wavelet) pre-processing. Conversely, time series that exhibit non-homogeneity in their variance structures benefit from wavelet pre-processing.

We have described a method, based on hypothesis testing for variance homogeneity, for determining the suitability of wavelet pre-processing for subtractive clustering fuzzy models. The use of hypothesis testing, it appears, makes possible the automatic selection of candidate time series for wavelet processing. Initial results based on the analysis of economic time series are encouraging: in all but two of the ten cases investigated, the method successfully identifies candidate time series for wavelet pre-processing.
Chapter 5

Conclusions and Future Work

This thesis aimed at examining the forecast performance of subtractive clustering fuzzy models on nonstationary time series characterised by trends, seasonal patterns, and variance changes. Generally, fuzzy models have been deemed capable of directly analysing real-world time series without need for pre-processing and, in cases where pre-processing is implemented, an ad hoc approach has been used. The objectives of this study were to:

i) Empirically examine the capability of fuzzy systems to directly model data with trends, patterns, discontinuities and other complex behaviour.

ii) Investigate the effects of different informal data pre-processing techniques, selected in an ad hoc fashion, on the forecast performance of fuzzy models on such time series; and determine if a systematic selection of data pre-processing techniques is beneficial.

iii) Explore the use of a formal, wavelet-based technique for data decomposition prior to the application of fuzzy models, and evaluate the performance of the wavelet-based model.

iv) Examine whether wavelet-based processing consistently results in improved forecast performance and, if not, propose a method for testing the suitability of wavelet-based pre-processing.

In this chapter, an overview of the main findings of this research is presented and discussed in the context of related research in the literature, and in relation to stated research motivations. We also make suggestions for future research.

5.1 Main Research Findings

First, we recapitulate the work presented in this thesis, and then describe the main findings. After a brief introduction in chapter 1, a discussion of different fuzzy models for time series analysis was presented in chapter 2, including a critique of the state-of-the-art, where we argued that the use of more sophisticated fuzzy models for improving the structure
identification and parameter optimisation capability of fuzzy models seem to produce incremental improvements in forecast performance of fuzzy models. We suggested that, rather than focusing on producing sophisticated models, an alternative approach to improving the forecast performance of fuzzy models, based on reducing the complexity of data used in generating fuzzy models, requires investigation.

In chapters 2 and 3, we discussed different strategies that can be used for data pre-processing, and classified such approaches into two broad categories: (i) informal ad hoc methods, based on traditional pre-processing methods and (ii) formal wavelet-based methods for data pre-processing. We also examined the issue of systematic selection of data pre-processing method in both informal and formal frameworks, and proposed methods for testing and systematically selecting data pre-processing methods. A fuzzy-wavelet framework was then proposed for the analysis of nonstationary time series.

The novelty in this work is that, unlike wavelet-based schemes reported in the literature, our method takes the statistical properties of the time series into consideration, and only recommends wavelet-based pre-processing when the properties of the data indicates that such pre-processing is appropriate. The wavelet variance profile of a time series, and an automatic method for detecting variance breaks in time series, are used as indicators as to the suitability of wavelet-based pre-processing. This approach facilitates the application of our framework to the different characteristics exhibited in real-world time series.

In chapter 4, we described simulations and experiments used to evaluate the proposed methods. We have evaluated the forecast characteristic of fuzzy models generated from pre-processed data, by examining both the mean absolute percentage error (MAPE) statistic, and equally importantly, the forecast error distribution resulting from the use of such pre-processing methods. Also, we investigated the effect of data pre-processing on the number and distribution of clusters generated by the fuzzy model under different pre-processing conditions. Using both simulated time series, designed with characteristics of interest, and real-world economic time series, we have been able to arrive at the following conclusions:

i) Based on empirical evidence, subtractive clustering fuzzy models are generally unable to directly model data with trends, patterns, discontinuities and other complex behaviour. Our study on simulated data and ten economic time series with trend, seasonal patterns and discontinuities indicates that, where such characteristics are present, the forecast performance of the time series worsens. This result corroborates the findings of a large body of the literature, where neural networks have been found
to be incapable of modelling nonstationary time series data without data pre-processing (Nelson et al., 1999; Zhang et al., 2001; Zhang & Qi, 2005; Taskaya-Temizel & Ahmad, 2005). This aspect of the thesis can therefore be viewed as an extension of previous work on neural networks, another class of universal approximators.

ii) Typically, the use of data pre-processing results in improved forecast performance for fuzzy models. Using ten economic time series, and informal pre-processing methods for making nonstationary time series stationary through first and seasonal differencing, it was observed that fuzzy models generated from pre-processed data typically resulted in better forecast performance. However, no single pre-processing method consistently resulted in improved forecast performance across all the time series data investigated, unlike results reported for neural networks in Zhang & Qi (2005). Also, specific ad hoc data pre-processing methods appear to match or be suitable for specific time series. The use of the partial autocorrelation functions provides an indication as to the suitability of ad hoc pre-processing techniques. Based on the use of the PACF, pre-processing choices were made, which maintains a balance between forecast performance in terms of low MAPEs, and model robustness, described in terms of the spread of forecast errors about a median value.

iii) In general, wavelet-based pre-processing is beneficial for analysing nonstationary time series, and is robust to the presence of noise, seasonal and trend patterns, and inhomogeneous variance. Wavelets provide an automatic method that formalises the pre-processing of time series, and captures transient events. In the fuzzy-wavelet framework, time series that exhibit changes in variance require pre-processing, and wavelet-based pre-processing provides a formal, parameter-free method for decomposing such time series. However, for cases where the variance structure of a time series is homogeneous, wavelet pre-processing leads to worse results compared to an equivalent analysis carried out using raw data. An automatic method for detecting variance breaks in time series can be used as an indicator as to whether or not wavelet-based pre-processing is required.

5.2 Suggested Directions for Future Work

There are several enhancements that can be made to this work:

i) The focus in this thesis is on wavelet-based pre-processing for fuzzy models, which are a class of universal approximators. There is a need to investigate whether similar
results will be obtained with other classes of universal approximators, such as neural networks, and hybrids of fuzzy systems with neural networks and/or genetic algorithms, using a mixture of time series with homogeneous variance profiles, and time series with discontinuities in the variance across time scales. Also, rather than reporting only overall performance measures like the MAPE and RMSE, bias-variance analysis of modelling error is needed to provide insight into the learning and generalizing behaviour of models trained on raw and pre-processed data.

ii) The choice of the time lag to be included in the antecedents of the fuzzy rules, i.e. window sizes have been selected by testing over a number of different window sizes. This is an inefficient, trial-and-error approach. The automatic determination of optimal window sizes for a given time series is potentially useful for generating fuzzy models with improved forecast performance, with the added advantage of reducing the computational requirement of the technique.

iii) The use of fuzzy models to analyse all the wavelet components, before the aggregation stage, may not be the best option. A possible alternative, especially for nonstationary time series with dramatic variance changes, is to use different analysis methods suited to the specific component. For example, high frequency level-1 wavelet details can be tested for the presence of GARCH effects, and modelled using the GARCH framework, rather than a soft computing approach.

iv) The number of rule clusters generated using the fuzzy-wavelet method is more than that generated using raw data. A more parsimonious selection of wavelet coefficients for model generation may help in reducing the number of rule clusters.

v) Many time series, especially in financial markets, are regarded as ‘noisy’, and the noise component is deemed to impair forecast accuracy. Wavelets have been used to denoise such series i.e. to separate permanent, global events from local non-recurrent events. The application of wavelet-based denoising in the proposed framework may lead to a more robust time series analysis method. Also, the use of the energy profile of wavelet components may aid in the application of ‘hard thresholding’ method for noise removal.

vi) In this work, univariate time series analysis has been carried out. This assumes that all the available and relevant information have been incorporated in the time series. However, in financial markets, sentiment analysis is becoming increasingly important due to the widespread access to real-time news items. It has been argued that investors have a tendency to speculate and that specific market news items appear to be strongly correlated with price shifts. A time series model incorporating sentiment analysis in a multivariate analysis framework may be beneficial.
Bibliography


Bibliography


Bibliography


## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACF</td>
<td>autocorrelation function</td>
</tr>
<tr>
<td>ANFIS</td>
<td>adaptive network-based fuzzy inference system</td>
</tr>
<tr>
<td>ANN</td>
<td>artificial neural networks</td>
</tr>
<tr>
<td>AR</td>
<td>autoregressive</td>
</tr>
<tr>
<td>ARCH</td>
<td>autoregressive conditional heteroskedasticity</td>
</tr>
<tr>
<td>ARIMA</td>
<td>autoregressive integrated moving average</td>
</tr>
<tr>
<td>ARMA</td>
<td>autoregressive moving average</td>
</tr>
<tr>
<td>CI</td>
<td>controversy index</td>
</tr>
<tr>
<td>CWT</td>
<td>continuous wavelet transform</td>
</tr>
<tr>
<td>DFT</td>
<td>discrete Fourier transform</td>
</tr>
<tr>
<td>DWT</td>
<td>discrete wavelet transform</td>
</tr>
<tr>
<td>EFUNNS</td>
<td>evolving fuzzy-neural networks</td>
</tr>
<tr>
<td>F-CEO</td>
<td>fuzzy model with constrained evolutionary optimization</td>
</tr>
<tr>
<td>FCM</td>
<td>fuzzy C-Means</td>
</tr>
<tr>
<td>FD</td>
<td>first difference</td>
</tr>
<tr>
<td>FIS</td>
<td>fuzzy inference systems</td>
</tr>
<tr>
<td>FRB</td>
<td>Federal Reserve Board</td>
</tr>
<tr>
<td>FWN</td>
<td>fuzzy wavelet network</td>
</tr>
<tr>
<td>GARCH</td>
<td>generalised autoregressive conditional heteroskedasticity</td>
</tr>
<tr>
<td>GEFREX</td>
<td>genetic-fuzzy rule extractor</td>
</tr>
<tr>
<td>GF</td>
<td>genetic-fuzzy</td>
</tr>
<tr>
<td>GFPE</td>
<td>genetic-fuzzy predictor ensemble</td>
</tr>
<tr>
<td>GNF</td>
<td>genetic-neuro-fuzzy</td>
</tr>
<tr>
<td>HENFS</td>
<td>hybrid evolutionary neuro-fuzzy system</td>
</tr>
<tr>
<td>MA</td>
<td>moving average</td>
</tr>
<tr>
<td>MAE</td>
<td>mean absolute error</td>
</tr>
<tr>
<td>MAPE</td>
<td>mean absolute percentage error</td>
</tr>
<tr>
<td>ME</td>
<td>mean error</td>
</tr>
<tr>
<td>MG</td>
<td>Mackey-Glass</td>
</tr>
<tr>
<td>MISO</td>
<td>multiple-input single-output</td>
</tr>
<tr>
<td>MODWT</td>
<td>maximal overlap discrete wavelet transform</td>
</tr>
<tr>
<td>MOHGA</td>
<td>multi-objective hierarchical genetic algorithms</td>
</tr>
<tr>
<td>MRA</td>
<td>multiresolution analysis</td>
</tr>
<tr>
<td>MSE</td>
<td>mean square error</td>
</tr>
<tr>
<td>NDEI</td>
<td>non-dimensional error index</td>
</tr>
<tr>
<td>NF</td>
<td>neuro-fuzzy</td>
</tr>
<tr>
<td>NRMSE</td>
<td>normalised root mean square error</td>
</tr>
<tr>
<td>OED</td>
<td>Oxford English Dictionary</td>
</tr>
<tr>
<td>PACF</td>
<td>partial autocorrelation function</td>
</tr>
<tr>
<td>R</td>
<td>raw</td>
</tr>
<tr>
<td>RMSE</td>
<td>root mean square error</td>
</tr>
<tr>
<td>SANFN-GSE</td>
<td>self-adaptive neural fuzzy network with group-based symbiotic evolution</td>
</tr>
<tr>
<td>SARIMA</td>
<td>seasonal autoregressive integrated moving average</td>
</tr>
<tr>
<td>SCMF</td>
<td>sum of controversies associated with a membership function</td>
</tr>
<tr>
<td>SD</td>
<td>seasonal difference</td>
</tr>
<tr>
<td>SD+FD</td>
<td>seasonal and first difference</td>
</tr>
<tr>
<td>STFT</td>
<td>short-time Fourier transform</td>
</tr>
</tbody>
</table>
### Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDNN</td>
<td>time delay neural networks</td>
</tr>
<tr>
<td>TS</td>
<td>Takagi-Sugeno</td>
</tr>
<tr>
<td>TSK</td>
<td>Takagi-Sugeno-Kang</td>
</tr>
<tr>
<td>USCB</td>
<td>US census Bureau</td>
</tr>
<tr>
<td>UWCSS</td>
<td>unadjusted women’s clothing stores sales</td>
</tr>
<tr>
<td>WM</td>
<td>Wang-Mendel</td>
</tr>
</tbody>
</table>