Coventry University

DOCTOR OF PHILOSOPHY

Leakage forecasting with fuzzy evolving techniques

Birek, Lech

Award date:
2016

Awarding institution:
Coventry University

Leakage forecasting with fuzzy evolving techniques

By

Lech Birek

January 2016

A thesis submitted in partial fulfilment of the University's requirements for the Degree of Doctor of Philosophy

Declaration

I declare that the work described in this PhD thesis, unless otherwise stated in the text, is my own work and has not been previously submitted for any academic degree.

Lech Birek

January, 2016

Acknowledgements

I am deeply indebted to my Director of Studies, Prof. Dobrila Petrovic, and my supervisor, Prof. John Boylan, who have supported me constantly throughout my research and writing period, provided guidance, encouraged and motivated me while working on this Thesis. I would also like to thank the staff from Severn Trent Water, in particular Alan Tripp, who have given me the opportunity to conduct this research and helped me to understand the problems of leakage forecasting. I would also like to gratefully acknowledge Prof. Keith Burnham, the director of Control Theory and Applications Centre (CTAC), for his invaluable help during this research and his constant support. I also wish to thank my friends and colleagues at Coventry University for their advice, friendship, support and great working environment. Finally, I would like to acknowledge my girlfriend, Ania Moskalik, my friends and family for their support and encouragement during the research which has been of great help in order to stay focused and motivated.

Abstract

The amount of water lost in various leakage events in water networks in the UK not only generates economic loses for the water management companies, but more importantly, threatens supplies to households and businesses in times of severe drought. To address this problem, a novel forecasting approach is presented in this thesis that aims at predicting the leaks so that appropriate resources can be planned to handle them.

A modified Fuzzy Evolving Takagi-Sugeno (Mod eTS) algorithm has been developed and applied to the leakage forecasting problem. The algorithm recursively clusters the data samples as soon as they become available and builds the fuzzy structure based on the generated clusters. Each new data sample is compared to already gathered data by calculating its potential. Based on that, the new data sample can modify the structure of existing clusters or initiate a new one. The modification may not only shift the cluster centre, but can also change the area of influence of the cluster, by adjusting its radius. The clusters are used to generate fuzzy If-Then rules through Takagi-Sugeno inference. The modified version of Recursive Least Squares algorithm is used to estimate parameters of the resulting linear equations, by taking into account the firing strength of the Fuzzy If-Then rules. This way the system is allowed to evolve, by constantly learning and adapting to

changes in the data.

The algorithm has been applied to two sets of data: the leakage data from 8 regions of operation of one of the biggest water suppliers in the UK and to artificially generated time-series data using the Mackey-Glass process. The novel approach is evaluated and compared with a number of computational intelligence and widely accepted, statistical methods and consistently demonstrated highest accuracy for the leakage data sets (MASE of 1.263 as compared to 2.319 on average for other methods). It also preformed well on the Mackey-Glass time-series when both the accuracy and the complexity of the model were considered. Its performance demonstrated the potential to be further developed and applied in the industrial setting, not only in the water industry, but also in other areas.

Contents

List of Figures

List of Tables

"Prediction is very difficult, especially if it's about the future."

––Niels Bohr, Nobel laureate in Physics

Chapter 1

Introduction

In the last few decades, fuzzy logic has been successfully used to model complex, dynamic problems that would otherwise be difficult to accomplish using conventional mathematical approaches. The advantage of using fuzzy logic lays in its ability to express non-linear relations among variables, typically by combining several linear sub-models, expressed in the form of fuzzy If-Then rules. The rules are often generated based on an expert's knowledge. This, however, requires the presence of an experienced specialist, and it is usually time consuming and not feasible for large scale applications. One of the ways to overcome this problem is to use data clustering. The data points which share similar properties are automatically grouped into clusters which in turn are a base for the fuzzy If-Then rules. The recent advances in the field of fuzzy clustering (Lughofer 2011*b*, Angelov et al. 2010) have allowed for real-time generation and update of fuzzy If-Then rules. This is particularly helpful in situations where it is important to adapt the rule structure to changing conditions as well as being able to control the way the clusters are generated in the real time. The applications of this approach have

not yet been thoroughly explored in all prospective research areas, forecasting being one of them. As development and application of forecasting models to be used in real-world scenarios are often difficult tasks because of non-linear relationships between dependent and independent variables, measurement errors and incomplete datasets, the application of the fuzzy evolving methods in forecasting is a promising research direction for scientists working in those research areas.

1.1 Forecasting

Forecasting is the process of predicting future events based on data from the past and present. Forecasting has been widely researched by scientists around the world and it is integral to any business activity, especially in sales, finance, utilities and weather forecasting.

There are many forecasting methods, which can be divided into two groups: qualitative (or judgemental) and quantitative forecasting.

1.1.1 Qualitative forecasting

Judgemental methods involve subjective opinions and decisions, which are formed based on the knowledge from experts, group of experts or are formed as a result of opinion of a target group. The most commonly used methods are:

- Judgemental adjustments, where all or parts of the forecasts are adjusted by an expert in the field. This is a very common approach,

as in many cases statistical methods cannot incorporate the effects of some future events that may be known to the expert, or incorporate past experiences.

- Delphi method is a formalized way of obtaining a forecast from a group of experts in a structured, iterative manner, and was invented by Dalkey & Helmer (1963).

- Forecasting by analogy, where forecast is obtained based on the comparison with data from a similar scenario - "what forecast did we use last time when this event happened?"

- Scenario forecasting, where experts generate several outcomes of the forecast, usually, worst, middle and best cases scenarios. Forecasts based on scenarios allow for many possible forecasts to be generated in order to identify the likely extremes.

Some of those methods will be discussed in more detail in Section 2.2 of Chapter 2.

1.1.2 Quantitative forecasting

In quantitative forecasting the future data is mathematically predicted as a function of past and present data. Those methods are applicable when the data gathered so far indicates that the events of the future can be predicted based on that data (Hyndman & Athanasopoulos 2014). There are many quantitative methods, the most used and popular ones being:

- Simple regression methods, which assume that the outcome of the

forecast can be described by a linear relationship with one predictor variable.

- Multiple regression methods, which are very similar to simple regression methods, except that the forecast can be described by a linear relationship with more than one predictor variable.

- Exponential smoothing methods, which produce forecasts that are weighted averages of past observations, with the weights decaying exponentially as the observations get older. More details on those methods will be included in Section 2.3 of Chapter 2.

- ARIMA models, which use the similarity between observations as a function of the time lag between them (autocorrelation) in order to generate the forecast. More details on those will be provided in Section 2.4 of Chapter 2.

- Computational Intelligence methods, which try to automatically establish relationships (linear and non-linear) between the input variables and the output in order to generate a forecasting model. Those will be discussed in Section 2.5 of Chapter 2.

1.2 Fuzzy logic systems and fuzzy clustering

1.2.1 Fuzzy logic systems

The concept of fuzzy logic was introduced by Zadeh (1965) who proposed a methodology of processing data by allowing a partial set membership rather than a crisp set membership or non-membership. This allowed

the mathematical representation of vague, uncertain statements to create solutions with smoother transition between states.

The concept of fuzzy logic has been used to build fuzzy logic systems. Those systems use a combination of membership functions (an example of a membership function is presented below in Fig. 1.1) connected by fuzzy statements in order to model the behaviour of a system. This can be a physical system or, as in the case in this thesis, a system that describes the relationships between variables in order to generate a forecast.

Figure 1.1: Gaussian membership function.

An example of the fuzzy statement can be seen below. The predictor variables are modelled using fuzzy membership functions and the outcome is obtained through defuzzification of the system. The outcome can be represented by a fuzzy set (this is called Mamdani inference - Eq. 1.1) or may be represented as a linear function (similar to multivariate models) and is called Takagi-Sugeno

inference - Eq. 1.2.

$$\text{R1: } \textbf{IF } x_1 \textbf{ is } FREEZING \textbf{ AND } x_2 \textbf{ is } WINTER1 \textbf{ THEN } y = MANY \quad (1.1)$$

where x_1 and x_2 are predictor variables modelled by fuzzy sets through fuzzy membership functions (those could be similar to Fig. 1.1) described as $FREEZING$ and $WINTER1$ and the outcome y modelled by the fuzzy set $MANY$.

$$\begin{aligned} &\text{R1: } \textbf{IF } x_1 \textbf{ is } \textit{FREEZING } \textbf{AND } x_2 \textbf{ is } \textit{WINTER 1} \\ &\quad \textbf{THEN } y_1 = ax_1 + bx_2 + c \end{aligned} \quad (1.2)$$

where $y_1 = ax_1 + bx_2 + c$ is an output equation, where y_1 is the estimated output, a, b and c are equation parameters that are usually estimated using least squares method, and x_1 and x_2 are defuzzified values of fuzzy sets (in this case $FREEZING$ and $WINTER1$).

More detailed discussion and explanation of fuzzy systems will be presented in Section 3.3 of Chapter 3.

1.2.2 Fuzzy clustering

Clustering is a process where data is grouped based on observed similarities and patterns (Jain et al. 1999). An example of this can be seen in Fig. 1.2, where data is automatically grouped into 3 clusters based on the distance from the cluster centres.

Figure 1.2: k-means clustering algorithm.

Much like the fuzzy and boolean logic, clustering can be crisp or fuzzy. In crisp clustering each observed data point can only belong to one cluster. An example of the crisp clustering method is k-means. In fuzzy clustering a point belongs to a cluster with a degree of membership, which can depend on variety of factors, most often the distance to the cluster centre. An example of a fuzzy clustering can be c-means clustering method or subtractive clustering. Both approaches will be discussed in Section 3.4 of Chapter 3.

1.2.3 Automatic rule generation

One of the big issues of fuzzy systems is that fuzzy rules describing a system have to be built by an expert using domain knowledge. The developments in the area of fuzzy clustering methods allowed for automatic rule creation from data based on the position and spread of the cluster centres. More detailed

discussion on this approach and the algorithms will be presented in Section 3.5 of Chapter 3.

1.2.4 Evolving fuzzy systems

One of the drawbacks of the classic automatic rule generation is that rules could only be created once all, or a significant portion, of data has been collected. It was very difficult or almost impossible to easily extend the model once new data became available. Evolving fuzzy systems allow for recursive generation and update of both antecedent and consequent parts of fuzzy rules. This is achieved by using automatic on-line fuzzy rule creation based on the recursive clustering techniques. The model adapts itself based on the incoming data, without the need to manually build and estimate the parameters of the output functions. This approach has application to real-time systems or problems with non-linear, non-stationary data, since those can be approximated by a set of linear models (Wang & Mendel 1992). More details and the review of the existing algorithms will be presented in Chapter 4 of this thesis.

The approach is also applicable to problems in which relationships between different inputs and an output are not well understood and cannot be easily modelled, which is the case in the leakage forecasting problem described below.

1.3 Use case: water leakage

As demand for water increases every year together with an increase of population and overall level of life (Babovic et al. 2002), water distribution companies face the problem of maintaining and constantly improving their water networks. This issue is especially visible in countries such as the United Kingdom, which has many pipelines dating back to 1940's and experiences frequent structural failures, water leakages and interruption in service (Savic et al. 1997). In 1974 an extensive transformation started in the UK water industry when numerous small local water companies were merged to create regional authorities. Further changes were made when local authorities were privatized in 1989 and a regulatory body called OFWAT (Office of Water Services) was established. All these changes meant that water companies have to provide not only a certain level of service to their customers, but also satisfy requirements of the company's board and shareholders. One of the biggest problems which water companies face nowadays is to not exceed maximum agreed leakage levels.

Leakage detection and estimation of actual losses is a difficult task. Most households do not have meters installed to precisely measure the usage of water. The common approach is to measure the flow of water in different areas of the network during the night, when the demand for water is at its lowest level. The obtained measurements are compared during the course of a few days and the differences between the flow rates are investigated as possible leakage events. However, this approach is prone to errors as a sudden increase in demand, referred to as unaccounted night use, may lead to misreading the increase in usage of water as a leakage.

Leakage does not remain constant throughout the year and is subjected to the influence of seasonal factors, such as the temperature and rainfall, but also the investments made prior to the expected increase in leakage, water demand and materials used for pipes. All these factors have a big influence on the service levels and economic loss of water companies.

The difficulties in predicting the leakage which are caused by uncertainties in the data and relationships between the various explanatory variables are a promising area for exploration. These uncertainties are a result of measurement errors in water flows, unaccounted use of water (water theft, or simply lack of meters in households). They are also related to the relationships between the data. It is difficult to establish a simple relationship between the factors affecting the leakage, such as the effect of increase of resources in order to decrease the leakage.

As company operates across a considerable area, there are a number of factors which make different areas of operation unique. Those include structure of pipes, terrain or urbanisation level. This means that the forecasting model had to be tailored to the area of operation that it was applied to. Having an evolving, data focused, model that could be applied with minimal tuning would be beneficial and evolving fuzzy algorithms fulfil that need.

The company very often judgementally adjusts the forecasts, however those adjustments have not really been measured, and they can often be biased (Harvey & Harries 2004). Judgement should be incorporated in the forecasting support system and some researchers (Yager 1984) suggested using fuzzy subsets to provide a quantitative framework in which to represent linguistic forecasts.

The difficulties in predicting the leakage mentioned above are a promising area for exploration and work on the use of fuzzy methods in forecasting for this application. The fuzzy approach has been successfully used to model complex, dynamic problems, such as blood pressure monitoring (Omron Healthcare 2016) or control system for camera's auto-focus (Wilamowski & Irwin 2011), that would otherwise be difficult to accomplish using conventional mathematical approaches. The addition of real-time generation and data-based update of fuzzy If-Then rules through recursive clustering in evolving fuzzy systems can possibly mitigate issues related to difficulty in forming the rules manually.

1.4 Aims and objectives

The main aim of the research presented in this thesis is to evaluate the use of Mod eTS algorithm in forecasting the water leakage. This approach may prove particularly useful for this application as it is often difficult to determine the relationship between the dependent and independent variables (Makinde et al. 2014) and it is important to adapt the forecasting model to changing conditions. Having a number of regions of operation, it is also inconvenient to manually create forecasting models for each of these regions separately. An automatic method to generate those forecasts could be of great help to managers. The generalizability of the new method is also evaluated by using a benchmark data; the Mackey-Glass time series. With this in mind, the following objectives are considered in this thesis:

1. Investigate issues and difficulties that exist in forecasting water leakage and identify some possible areas for improvements.

2. Review of the existing statistical and computational intelligence forecasting methods.

3. Review the existing research on fuzzy evolving methods and the applications of those methods to forecasting.

4. Analyse the leakage data and choose appropriate predictor variables to be used in the forecasting process.

5. Develop the new fuzzy evolving algorithm and adapt it to work for the particular application of leakage forecasting.

6. Investigate the fine tuning of the parameters of the novel fuzzy evolving algorithm and recommend the choice of parameters for the leakage forecasting problem in order to obtain the best results.

7. Compare the performance of the novel algorithm to a number of well established statistical and computational intelligence methods, and also to other fuzzy evolving algorithms using the data of leakage problem.

8. Apply the novel algorithm to another, non-leakage, data set in order to evaluate its usefulness in other forecasting applications and analyse the differences in the obtained results (if any). Compare the performance on that new dataset to other forecasting algorithms.

1.5 Outline of the thesis

This thesis consists of 9 chapters (including this one). The brief outline of the thesis is presented below and the logical flow of the thesis is shown in Fig. 1.3.

Chapter 2: Forecasting: overview of basic forecasting methods. This chapter provides an overview of research on some of the more popular and established forecasting methods, such as judgemental forecasting, the exponential smoothing methods, and ARIMA models. A more detailed review on the use of other (not related to fuzzy clustering) Computational Intelligence methods in forecasting is also presented.

Chapter 3: General fuzzy methods and clustering. The aim of this chapter is to introduce the reader to the concept of fuzzy systems and clustering. The chapter discusses the fuzzy statements, fuzzy membership functions and fuzzy inference techniques. It explains what data clustering is and merges those two concepts to form a method for identification and automatic generation of fuzzy models, which is the foundation upon which the evolving fuzzy systems are built.

Chapter 4: Evolving fuzzy identification methods and their applications. In this chapter the concept of evolving fuzzy systems is presented. This approach allows for recursive generation and update of both antecedent and consequent parts of fuzzy rules, which is achieved by using automatic on-line fuzzy rule creation based on recursive clustering techniques. Three main families of evolving fuzzy algorithms are discussed: the eTS and its extensions, FLEXFIS and DENFIS. The advantages and disadvantages of those methods will be considered and some applications will be demonstrated.

Chapter 5: Use case - forecasting leakage in the water industry. The aim of this chapter is to outline the problem of water leakage forecasting and the factors which influence it. The current method of leakage forecasting is described and some issues and open questions are presented along with a possible approach to solve those.

Chapter 6: Modified Evolving Takagi-Sugeno (Mod eTS) algorithm for forecasting. In this chapter a novel algorithm, Mod eTS, is introduced. The algorithm was developed in order to increase the accuracy of prediction of the leakage forecasts, especially for periods and areas with high levels of leakage. The chapter provides a detailed description of how the algorithm can be utilized in forecasting and discusses the novel aspects that are introduced by the new algorithm.

Chapter 7: Application of the Mod eTS algorithm to leakage forecasting. This chapter presents the results and conclusions from the application of the Mod eTS algorithm to the leakage forecasting problem. It explains the process of choosing the explanatory variables and the pre-processing steps on the provided data sets. Further on, a sensitivity analysis on the configurable parameters is performed in order to establish their influence on the obtained results. Finally, the results of the application of the method to the leakage forecasting problem are compared with the results obtained from using other fuzzy forecasting and statistical methods.

Chapter 8: Evaluation of the Mod eTS algorithm on additional data set. In this chapter the Mod eTS algorithm is tested on the additional data set in order to establish if the conclusions and results from the previous chapter are still applicable to other data sets. The data used in this chapter is a well known non-linear time-series which is generated by the Mackey-Glass process. In addition to the Mod eTS three other algorithms are evaluated and the accuracy results are compared.

Chapter 9: Conclusions and Further Work. The last chapter summarises the outcomes of the research conducted in this thesis. The results obtained on both data sets are discussed and a number of directions for future research

in the area of forecasting and evolving fuzzy systems is suggested.

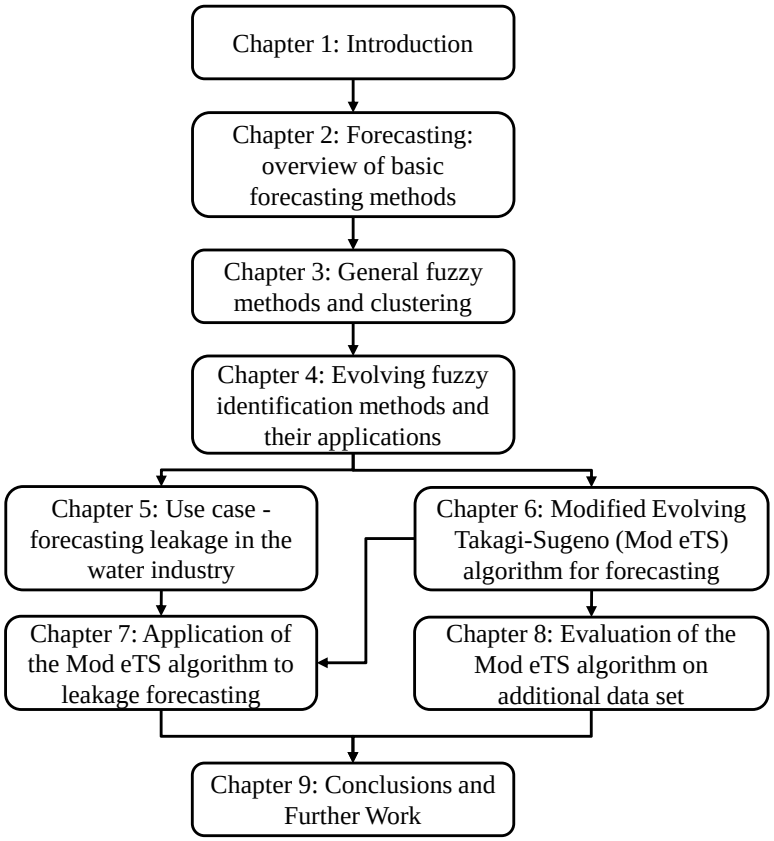

Figure 1.3: Logical flow of the thesis.

1.6 List of publications related to the thesis

1.6.1 Journal papers

Birek, L., Petrovic, D., & Boylan, J. (2014). Water leakage forecasting: the application of a modified fuzzy evolving algorithm. *Applied Soft Computing*, 14, 305–315.

1.6.2 Conference papers and proceedings

Birek, L., Petrovic, D., & Boylan, J. (2011). A fuzzy logic based approach to leakage forecasting in water industry. *In Proceedings of the 31st International Symposium of Forecasting* (pp. 1–12). Prague, Czech Republic: International Institute of Forecasters.

Birek, L., Petrovic, D., & Boylan, J. (2011). An application of fuzzy clustering techniques to leakage forecasting in water industry. *In Proceedings of the OR53 in Nottingham* (pp. 1–21). Nottingham, United Kingdom: OR Society.

Birek, L., Petrovic, D., & Boylan, J. (2012). Leakage forecasting based on a modified fuzzy evolving multivariate method. *In Proceedings of the 32nd International Symposium On Forecasting* (p. 78). Boston, USA: International Institute of Forecasters.

1.6.3 Awards

2nd Prize for a poster presentation entitled Fuzzy evolving techniques in leakage forecasting for water industry, 6th July 2012, *Faculty of Engineering and Computing Postgraduate Research Symposium 2012*, Coventry University, Coventry, United Kingdom

Best Poster as voted by Students Prize for a poster presentation entitled Fuzzy evolving techniques in leakage forecasting for water industry, 6th July 2012, *Faculty of Engineering and Computing Postgraduate Research Symposium 2012*, Coventry University, Coventry, United Kingdom

Chapter 2

Forecasting: overview of basic forecasting methods

2.1 Introduction

Even though this thesis will focus on the use of fuzzy evolving methods in forecasting, and particularly, their application to water leakage forecasting, it is important to introduce other methods that are used in that field and their applications, advantages and weaknesses. The results obtained using these methods will be compared with the results of the fuzzy evolving methods. In this chapter the overview of some of the well used and established forecasting methods will be presented. One of the most frequently used approaches is judgemental forecasting, which is a family of methods where all or parts of the forecasts are made or adjusted by an expert in the field. This is a very common approach, as in many cases statistical methods cannot incorporate the effects of some future events that may be known to the

expert, such as promotions, sales and holidays. In the further sections of this chapter two other families of methods are described: the exponential smoothing methods, and ARIMA models. These two statistical methods are very well established and are widely used in research as well as in practise. Finally a quick overview of the application of computational intelligence methods to forecasting problems is presented. Those methods try to automatically establish relationships (linear and non-linear) between the input variables and the output in order to generate a forecasting model. Computational intelligence methods have had a mixed success. They have often outperformed statistical methods, but no consistent conclusion has been drawn as to what factors contributed to the success of those methods in some applications. Those methods, and particularly clustering methods and their use in automatically and recursively generating fuzzy systems for forecasting will be in focus for the remainder of the thesis. Finally, the forecasting methods applied to water industry, and particularly to water demand and leakage forecasting are reviewed in the last section of this chapter.

2.2 Judgemental forecasting

Judgemental forecasts are frequently used to generate forecasts when no historical data is available (or when historical data is no longer relevant) or as a means to adjust the results obtained from statistical methods. In a dynamical environment statistical forecasting methods that are affected by past patterns can be perceived as being slow to react to change. Also it is not straightforward to include special events that are known to occur in the future (for example sales or promotions) into statistical methods

(Webby et al. 2005). On the other hand, the accuracy of judgemental adjustments is affected by unskilled personnel or by human behavioural factors. Direct judgement is very widely used in forecasts, but the human mind has limited information processing capacity and, often, heuristics are used to cope with the complexity of statistical models (Goodwin 2000). Integrating judgemental forecast with statistical methods has been a focus of a number of research investigations, since both have complementary strengths and weaknesses. Collopy & Armstrong (1992) examined the feasibility of rule-based forecasting. They proposed a procedure that applied forecasting expertise and domain knowledge to produce forecasts according to features of the data. Armstrong (1998) looked at how judgemental and statistical methods should be integrated for time-series forecasting. In Goodwin (2002) two methods of integration were discussed. The first one is voluntary integration, where the forecaster is supplied with information on a statistical forecast, and decides how to use it in forming the judgement. The second one is mechanical integration, where the average of independent judgemental and statistical forecasts is calculated. The conclusion from the investigation was that the forecasting support system, which allows and encouraged judges to interact with statistical methods appear to offer the most promising way forward. Harvey & Harries (2004), in their psychological study, investigated the practice of using multiple judgemental forecasts to adjust the statistical model. It was concluded that people put too much weight on their own opinion, whereas they should pay more attention to new information from more experienced or knowledgeable advisers.

Fildes et al. (2009) collected data of more than 60,000 forecasts in order to evaluate the effects of judgemental adjustments on accuracy of supply-chain planning. The analysis revealed that in 3/4 companies the adjustments

in general improved the accuracy. However more detailed analysis showed that small adjustments often damaged accuracy and, on top of that, they were made in wrong direction suggesting bias towards optimism. Trapero et al. (2013) showed that judgemental adjustments in the presence of promotions can enhance baseline forecasts, but not systematically. Transfer function models were then developed based on past promotional events that were able to achieve higher accuracy. Davydenko & Fildes (2013) addressed an issue in using many existing error measures to evaluate accuracy of judgemental adjustments. The authors suggested a method based on aggregating performance ratios across time series using the weighted geometric mean, which treats under and over forecasting evenly, has more symmetric distribution and is robust.

A range of research on Forecasting Support Systems (FSS) is also present. FSS is a set of procedures that enables users to combine relevant information, analytical models, and judgements, as well as visualizations, to produce forecasts and monitor their accuracy (Ord & Fildes 2013). The issue of trust in the results and methods of FSS were looked at by Goodwin et al. (2012). The conclusions were that trust was affected by level of noise in time series and whether the trend was present, but unaffected by presence or absence of point forecasts. When FSS labelled the results as best/worst case and provided an explanation, the level of trust increased. Fildes & Goodwin (2013) summarised that the key challenges for FSS relate to behavioural issues rather than the technical aspects of statistical forecasting. Spithourakis et al. (2015) evaluated the use of simple implementation of the FSS in undergraduate course to train students in use of support systems. A purpose-built questionnaire was used to determine the systems acceptance and perceived educational effects. The results were encouraging

and amplified the learning effect. The proposed improvement considered adding the aspects of Delphi method (Rowe & Wright 1999), which is a way of combining the forecast of several experts commonly used in business industry (Kauko & Palmroos 2014).

Several publications used computational methods in an attempt to aid or model the judgemental aspect of forecasting. Yager (1984) suggested using fuzzy subsets to provide a quantitative framework in which to represent linguistic forecasts. Lee & Yum (1998) used neural networks to model judgemental adjustments in time-series. In Ben Ghalia & Wang (2000) an intelligent system that used fuzzy rules and inference to support judgemental business forecasting in estimating hotel room demand was proposed.

The accuracy implication of judgemental adjustments in intermittent demand (demand that appears occasionally, with some periods of time with no demand at all) forecasts was a focus of investigation by Syntetos et al. (2009). The conclusions of the research was that although adjusted forecasts were more accurate than system forecasts, they did not improve over time. The follow up research by the same authors (Syntetos et al. 2010) focused on assessing the impact of judgemental adjustments on service and inventory levels in a manufacturing environment. The judgementally adjusted forecast had a big positive impact on service levels (the ability to deliver complete demand from the stock). The results obtained were much better compared to system forecasts. When adding lower inventory costs, the combined performance of the final forecast was even higher.

For more detailed review, readers should refer to Lawrence et al. (2006), where an overview of progress of research on judgemental forecasting is presented. Some of the more interesting conclusions from the review were

that small adjustments were not as useful as the big ones. The view on the future was that more investigation into the performance of experts in forecasting had to be carried out since different studies yielded contradictory conclusions. Also it is important to investigate how people acquire and use information when they make forecasts and the effects of differences in the availability of information. The research should focus on the influence of heuristics and biases on forecast accuracy and development of improved methods for supporting judgemental forecasters, particularly support in identifying when judgemental intervention is needed.

2.3 Exponential smoothing methods

Exponential smoothing is a family of forecasting methods where forecasts are weighted averages of past observations, with the weights decaying exponentially as the observations get older. The work on exponential smoothing methods was initiated in the late 1950s by Brown (1959), Holt (2004) (a re-print of the article from 1957) and Winters (1960). For many years exponential smoothing methods have been considered as the most popular forecasting methods used in business and industry (Hyndman et al. 2008). The different exponential models are presented and explained in the Appendix, in section A.1.

In recent years most research in the area of exponential smoothing has focused on state space models of exponential smoothing (Ord et al. 1997, Hyndman et al. 2002), which allow the production of not only point forecasts but also the confidence intervals around them. A comprehensive overview of the progress made in this area has been presented in the book by Hyndman

et al. (2008). Some other methods based on series decomposition also evolved, such as Theta model (Assimakopoulos & Nikolopoulos 2000). The proposed method decomposes the original time series into two or more different Theta-lines. These are extrapolated separately and the subsequent forecasts are combined. The method performed particularly well in the M3 forecasting competition (Makridakis & Hibon 2000).

Smoothing methods have been applied to virtually every area of forecasting, as they work particularly well in business context and are easy to implement. Syntetos et al. (2009) reviewed the progress of research on forecasting for inventory planning and pointed out that damped trend smoothing methods worked particularly well for those applications. However, in cases where external factors may affect the forecast, such as day of the week, time of the year, some known events, usually more complex multivariate methods work better. Fildes et al. (1997) described two use cases for short-term demand forecasting for the utilities, water and gas. In these two cases extrapolative methods based on the past data history alone were outperformed by more complex multivariate approaches that included information on the effects of weather.

For further reading both papers by Gardner (1985, 2006) provide comprehensive overviews of the research on exponential smoothing.

2.4 ARIMA models

Similarly to exponential smoothing, ARIMA models are one of the most widely-used approaches to time series forecasting. While exponential

smoothing models are based on a description of trend and seasonality in the data, ARIMA models aim to describe the autocorrelations in the data (Hyndman & Athanasopoulos 2014), where autocorrelation is the similarity between observations modelled as a function of the time lag between them.

There is a number of ARIMA models, a few of the most popular and used are defined in the Appendix, in section A.2.

The ARIMA models have been used extensively in the area of energy, financial and inventory (Syntetos et al. 2009) forecasting. Hagan & Behr (1987) reviewed the application of time series analysis methods to electricity load forecasting and shown that they are well suited to this application, with the drawback of the inability to accurately represent the nonlinear relationship between load and temperature. Pai & Lin (2005) presented a methodology that tries to overcome that weakness, as it exploits the hybrid approach where ARIMA and SVM (Support Vector Machine) models are used in forecasting stock prices. Kavasseri & Seetharaman (2009) examined the use of ARIMA models to forecast wind speeds on the day-ahead (24 h) and two-day-ahead (48 h) horizons with results indicating significant improvements compared to other methods.

2.5 Computational intelligence methods

A number of computational intelligence methods have been also applied to forecasting problems. Hand (2009) in his paper on data mining and forecasting highlights that use of empirical models might lead to effective prediction and forecasting, but often does not lead to an enhanced

understanding of the effect of the change in input data on the forecast. Stepnicka et al. (2012) compared several computational intelligence methods in their study which evaluated Artificial Neural Networks (ANNs), Support Vector Machines (SVMs) and combinations of those methods with fuzzy rules on several seasonal time-series. The results showed that those methods were comparative in performance and accuracy to forecasts using ARIMA models.

Zhang et al. (1998) reviewed the progress of using artificial neural networks for forecasting. He concluded that while ANNs provide a great deal of promise, they also embody much uncertainty as researchers to date are still not certain about the effect of key factors on forecasting performance of ANNs. Research indicated that the ensemble (combination) of networks proved to outperform the single "best" network. Kourentzes et al. (2014) proposed an enhanced mode ensemble operator. Results indicated that mode ensembles overcame issues of uncertainty associated with data sampling, the stochasticity of neural network training and the distribution of the forecasts. Crone et al. (2011) reported on results of the extension of M3 forecasting competition aimed at Computational Intelligence methods. The results highlighted the ability of Neural Networks to handle complex data, including short and seasonal time series. The authors concluded, that although results were promising, the sheer number of methods that have been used in the competition makes it very difficult to establish what actually makes them perform well, and therefore a more focused approach is needed.

Song & Chissom (1993*b*) introduced the idea of fuzzy time series, where the time series data is partitioned into a set of distinctive fuzzy sets. This allows for forecasting in fuzzy environments, in which the historical data are fuzzy sets or linguistic values. The method has been applied by the

same authors to forecasting university enrolments (Song & Chissom 1993*a*, 1994). The results indicated, that the application of this method resulted in lower forecasting errors (3.18% and 4.37% on average) than in other studies on the same dataset. A number of other applications and extensions were also studied. In Chen & Tanuwijaya (2011) multivariate fuzzy forecasting based on fuzzy time series and automatic clustering was applied to forecast the Taiwan Stock Exchange Capitalization Weighted Stock Index. Here the proposed approach outperformed a number of other fuzzy forecasting methods, but was not compared to results obtained from any statistical approaches. In Egrioglu et al. (2011) the fuzzy time series forecasting method based on Gustafson-Kessel fuzzy clustering was introduced and applied to the same problem of forecasting university enrolments. The results indicated lower MSE when compared to the results obtained from earlier research on this dataset. Duru (2012) used a fuzzy integrated logical forecasting method (M-FILF) and multiplicative time series clustering to model a time-varying volatility for dry cargo (cargo that is of solid, dry material) freight market. The proposed algorithm was superior to GARCH (Generalised Autoregressive Conditional Heteroscedasticity) method, which was applied to freight market problems in earlier research.

Use of fuzzy sets and clustering in forecasting has also been explored by research community in a number of papers. Different clustering methods were used to model the fuzzy rules which were later used to produce the forecasts. Cardoso & Gomide (2007) and Chen & Chang (2010) used various versions of c-means clustering. Hadavandi et al. (2011) used k-means clustering to categorize data into k clusters and feed into a genetic fuzzy system for sales forecasting.

Evolving methods, in which the cluster structure and fuzzy rules evolve as new data becomes available have also been investigated in financial applications. In Maciel et al. (2011) evolving fuzzy systems were used for pricing fixed income options and in Dovzan et al. (2012) for petrol sales prediction.

There has been a considerable amount of research that focused on using Computational Intelligence methods in forecasting for utilities (particularly water) industry. As an example, in Kelo & Dudul (2011) a time lagged recurrent neural network was used to predict short-term electrical power load and Alvisi & Franchini (2011) used fuzzy neural networks to forecast water level and discharge. Li & Huang (2009) used fuzzy-stochastic-based violation analysis method for planning water resources management systems. Mangalova & Agafonov (2014) used k-nearest neighbour (kNN) algorithm to predict the wind power plants' power output. Based on RMSE criterion, the method achieved 2nd place in the Global Energy Forecasting Competition in 2012. Weron (2014) reviewed forecasting methods for Electricity price forecasting. The author pointed that Computational Intelligence (Neural Networks, Fuzzy Neural Networks and Support Vector Machines) methods have the advantage over statistical methods at better handling of complexity and non-linearity. At the same time, this flexibility is also their weakness as this behaviour does not necessarily result in better point forecasts. The pool of available Computational Intelligence tools is so diverse and rich that it is hard to find an optimal solution. It is also hard to compare the different methods thoroughly, as conclusions can only be drawn about the performance of a given implementation of a method, with certain initial conditions (parameters) and for a certain calibration dataset.

2.6 Forecasting in water industry

A number of approaches have been used to improve forecasting in water industry. In general, two most important factors for water industry are being forecasted: the demand for water and the amount of leakage that may occur in the water network. The problem of water demand and leakage prediction is complex due to non-linearities, therefore this limits the number of forecasting models that can be applied. The non-linearities originate from the nature of the problem - nearly all the physical phenomena are obtained by principles of conservation and are expressed in terms of non-linear partial or ordinary differential equations expressing these principles (Makinde et al. 2014).

In Herrera et al. (2010), different predictive approaches have been described and compared. This, among others, included Artificial Neural Networks (ANN), random forest, support vector regression and a heuristic model which was built based on the empirical analysis of the time-series. The results showed that, in general, machine learning methods outperformed different variants of ANN, as well as showing poor performance of the heuristic method. In Nasseri et al. (2011) the use of Extended Kalman Filter (EKF) and Genetic Programming (GP) approaches have been tested. The GP has been used to generate candidate models, which were then used for filtering based on the EKF. Although authors claimed promising results, the dataset consisted of only water demand time-series and did not include any explanatory variables, which, as described in Herrera et al. (2010), have influence on the water demand. Bennett et al. (2013) included those components and built an ANN-based residential water end-use demand forecasting model, where ANNs were used to model relationships between micro-components, such as demography, water efficiency of household

appliances and the use of water. It is worth mentioning, that in addition to that, as highlighted by Obradović (2000), in order to properly predict the water demand in real-life systems, the water losses must also be taken into account and should also be modelled separately.

With regards to leakage forecasting, some research has been done in the way the various regions of the network should be prioritized for inspection in order to prevent the leaks from occurring. In Francisque et al. (2009) a fuzzy risk approach was employed, which helped to deal with vague and imprecise data, such as water age, structural integrity of the pipes, water quality parameters and consumer sensitivity to water-related illness. This approach helped the experts to identify the most significant inputs by conducting a sensitivity analysis. Another approach, which was also able to deal with unavoidable uncertainties, involved using a Bayesian probabilistic framework (Poulakis et al. 2003). It predicted the most probable leakage locations and identified the noise threshold beyond which reliable diagnosis is not possible. Similar research was conducted by Babovic et al. (2002) who used Bayesian methods to determine the risks of pipe bursts and improve leakage detection in water pipe networks.

2.7 Conclusions

In the next few chapters, the focus will be put on the use of computational intelligence methods in forecasting, in particular the use of clustering methods and fuzzy systems. The principles of clustering and fuzzy systems will be introduced and the recent extension to those methods, fuzzy evolving systems, which allow for automatic generation of fuzzy rules as the data

becomes available will be presented. It is important however, to have a bigger picture on what other forecasting methods are being used in research and in practice. It is worth remembering, that the methods described in this chapter are only a small subset of what researchers and professionals are using in forecasting. However those methods described here are one of the most frequently used and many new statistical forecasting methods take inspiration from them. Later on in the thesis, several methods described here will be compared with a novel fuzzy evolving forecasting algorithm (introduced Chapter 6) on a leakage forecasting problem and on an artificial time series generated by the Mackey-Glass process. The methods will be assessed in terms of accuracy and a number of clusters, where applicable.

Chapter 3

General fuzzy methods and clustering

3.1 Introduction

In order to understand the fuzzy evolving methods and their use in forecasting some basic knowledge of fuzzy systems and clustering is required. The chapter is split into 3 sections. In the first one the fuzzy statements, fuzzy membership functions and fuzzy inference techniques are explained. In the second section the idea of data clustering is shown and two methods: k-means, which is a crisp clustering method; and c-means, a fuzzy clustering method are described. In the last section those two concepts are merged to form a method for identification and automatic generation of fuzzy models. The principles of this method form the foundation upon which the evolving fuzzy systems are built.

3.2 Notation

	Common symbols
k	time instance
i	cluster, rule index
R	total number of clusters, rules
j	index of the input
h	total number of inputs
n	total number of data samples
m	total number of outputs (usually $m = 1$)
σ^2	variance, or width of the Gaussian membership function
σ	standard deviation
A_{ij}	an j^{th} fuzzy set of rule i
$x_k = \begin{bmatrix} x_{k1} & x_{k2} & \dots & x_{kh} \end{bmatrix}$	data point, data sample, data vector
$x_k^e = \begin{bmatrix} x_{k1} & x_{k2} & \dots & x_{kh} & 1 \end{bmatrix}$	extended data sample
$\tilde{x}_k$	normalized or standardized data sample
$\overline{x}_k$	mean of the k data samples
$\hat{y}$	estimated output
$c_i = \begin{bmatrix} c_{i1} & c_{i2} & \dots & c_{ih} \end{bmatrix}$	cluster centre
c_{win}	winning (for example closest to the data sample) cluster
$dist(a, b)$	distance measure between two points or vectors, usually Euclidean distance
r_α, r_β	cluster radii
$P(x_k)$	potential of data sample x_k
$P_k(x_k)$	potential of data sample x_k at time instance k
μ_i	activation/membership degree of rule i
τ_i	firing degree of rule i
	c-means clustering
ε	threshold for c-means clustering
	Parameter estimation
ψ_i	regressor vector of rule i

Ψ	data matrix of regressor vectors ψ
a_{ij}	parameter of j^{th} input for rule i
$\theta_i = \begin{bmatrix} a_{i1} & a_{i2} & \dots & a_{ih+1} \end{bmatrix}$	vector of parameters for rule i
Θ	matrix of parameter vectors θ
J	cost function of the Least Squares algorithm
	Mountain method
N_i	node in Mountain method
t	step index of the clustering process
$M(N_i)$	Mountain function
α, β	parameters used in Mountain function calculation and update
γ	threshold parameter
	Subtractive algorithm
α, β	parameter used in Subtractive clustering potential calculation and update
γ, γ_{up}, γ_{down}	threshold parameters

3.3 Fuzzy logic

3.3.1 Principles of Fuzzy Logic Systems

Physical systems and processes are most commonly modelled using statistical or mathematical models. However very often the knowledge of an expert, accumulated over a period of time, is added to include the events or states which are not easily modelled using classical approaches. The knowledge of an expert is usually collated and formulated using linguistic rules (for example If-else statements) and is formed into what is called an expert

system. Fuzzy logic systems form a subset of rule based expert systems in which empirical knowledge of the described process or a system is modelled using fuzzy logic approach (Lughofer 2011*a*).

The concept of fuzzy logic has been introduced by Zadeh (1965). In boolean logic the outcome of the logic statement is either $true$ or $false$. In Fuzzy Logic, the outcome values belong to a set between $[0, 1]$, also called a membership degree. This allows to mathematically represent vague, uncertain statements, such as *Vehicle is moving slow* or *Vehicle is moving fast*. Those statements would be very difficult to model using boolean logic as it would require determining from which speed exactly the vehicle stops moving *slow* and starts moving *fast*. With fuzzy logic the task becomes easier as you can assign different membership degrees of various values of speed to express the word *slow* and *fast*, so that the speed can belong to both sets with varying membership degree. This creates solutions with smoother transition between states. It can also capture some of the cases which would require additional logic statements if the boolean logic approach was used.

3.3.2 If-then statements

The expert systems are frequently modelled using If-then statements. If more inputs are used, they are connected with logic operators, such as AND, OR or NOT. An example of the simple expert system could be one which aim is to decided on number of engineers to be used on field to detect the number of leaks in the water network. The inputs to that system are x_1, the average temperature in deg. Celsius, and x_2, the season of the year, expressed in months, starting from January. The output y is the number of engineers

required to be send to the field to maintain the detection rate at an acceptable level. One of the rules of such an expert system could be represented as:

$$\textbf{IF } x_1 \textbf{ is } -5 \textbf{ AND } x_2 \textbf{ is } 12 \textbf{ THEN } y = 100 \tag{3.1}$$

meaning that if the temperature is -5 deg. and it's December, 100 engineers need to be send to the field.

If the system is complex, this approach would require generation of many rules and use of inequalities to accurately represent all of the possible states. It would also be prone to errors as it is necessary to cover the whole data space to make sure that all of the possible combinations of inputs are considered.

In fuzzy expert systems the structure of the rules stays the same but the crisp values of inputs are replaced with fuzzy sets. The fuzzy sets are used to model the inputs so that their values can be represented as linguistic equivalents, making it easier to build up the rule system. The equivalent of the crisp rule presented above can be represented as a fuzzy rule in the following manner:

$$\textbf{IF } x_1 \textbf{ is } FREEZING \textbf{ AND } x_2 \textbf{ is } WINTER \textbf{ THEN } y = PLENTY \tag{3.2}$$

where $FREEZING$, $WINTER$ and $PLENTY$ are fuzzy sets modelled to represent the numerical values of the inputs and outputs. The parts of the rule describing the interaction of inputs are called *antecedents*, while the result of the rule is called *consequent*. The fuzzy systems consist of a set of rules which usually cover all combinations of the inputs and outputs. The creation of fuzzy system is usually a time consuming process requiring knowledge of an expert in the particular field. Further in this chapter some of the methods to generate the

rules automatically from the data will be demonstrated.

3.3.3 Fuzzy Sets and fuzzy operators

Fuzzy sets are the main building blocks of a fuzzy system. The members of fuzzy sets belong to the set with various membership degrees in the interval $[0, 1]$. The way the membership value is calculated depends on the type of a fuzzy set used. The type is represented as a distribution function, such as triangle, trapezoid or Gaussian. The aim of using those functions is to model uncertainties which surround the inputs of the system. Three most popular types of fuzzy sets are described in Appendix in section A.3. The advantage of triangular and trapezoid fuzzy sets is that they have clearly defined ranges of maximum membership degrees and are easy to interpret. The disadvantage on the other hand is that they may not cover input space properly if the input data point falls outside of the set as the membership value would then equal to 0. This may lead to the situation when combination of inputs is not covered by the rules and will in turn fail to provide an output. The drawbacks of those fuzzy sets are mitigated in Gaussian fuzzy set as the Gaussian function will never provide 0 as a membership degree.

The antecedent part of fuzzy rule usually consists of more than one input. The inputs are connected using various fuzzy operators: unions, intersections and complements. The operators are described in the Appendix, in section A.4.

Having created a set of fuzzy If-then rules describing the system, the process is required to compute the output given a set of inputs. In fuzzy systems this process is called fuzzy inference. Two most popular inference systems in fuzzy logic are Mamdani inference system and Takagi-Sugeno inference system.

3.3.4 Mamdani inference

Mamdani inference was developed by Mamdani & Assilian (1975). In Mamdani inference, the consequent part of the fuzzy rule is expressed as a fuzzy set, which provides fully linguistic based system. The Mamdani inference process requires several steps:

1. Determine the set of fuzzy rules describing the system.
2. Fuzzify (obtain membership degrees) the inputs using fuzzy membership functions.
3. Combine the fuzzified inputs using the fuzzy operators to obtain the firing strength of each rule.
4. Find the consequence of each rule by combining the firing strength of the rule and the corresponding output membership function.
5. Combine the consequent value of each rule to get the output distribution.
6. Obtain the defuzzified output by using one of the defuzzifing methods.

Expanding on the leakage detection fuzzy system from previous section, an example of simple Mamdani fuzzy inference system is presented in section A.5 of the Appendix.

Mamdani systems are well understood due to the simple, linguistic expression of rules, but may struggle to give predictable outputs. Another type of fuzzy inference method, Takagi Sugeno, will be described in the next section and partially solves that issue.

3.3.5 Takagi-Sugeno inference

The fuzzy inference in which the consequent part is described by the mathematical equation rather than fuzzy set has been developed by Takagi & Sugeno (1985) and is also called Takagi-Sugeno (TS) inference system. In principle the antecedent part is described and evaluated in the same way as in Mamdani inference. The main difference lays in the consequent part in which a set of equations is used to describe all of the rules. The example rule from the previous section described using TS inference (Eq. 3.3) can be seen below.

$$\begin{aligned} &\text{R1: } \textbf{IF } x_1 \textbf{ is } \textit{FREEZING } \textbf{AND } x_2 \textbf{ is } \textit{WINTER 1} \\ &\quad \textbf{THEN } y_1 = ax_1 + bx_2 + c \end{aligned} \tag{3.3}$$

As in Mamdani inference, the firing strength τ_i of each rule i is calculated. To obtain defuzzified output y, the weighted average of each of the consequent functions y_i is calculated using the firing strengths of each rule (Eq. 3.4).

$$y = \frac{\sum_{i=1}^{R} \tau_i y_i}{\sum_{i=1}^{R} \tau_i} \tag{3.4}$$

The parameters of the consequent equations can be chosen empirically or, more often, are estimated using well established identification techniques, such as Least Squares method.

The TS method works really well as an interpolating supervisor of multiple linear models that differ depending on observed conditions of a dynamic non-linear system. It smoothly interpolates the linear gains that are applied across the input space. In turn, a TS system is well suited for modelling non-linear systems as they can be decomposed into a multi-model structure, where each subset of the data space is described by a fuzzy set and the generated corresponding linear equation.

3.4 Clustering

So far, all of the fuzzy models had to be built by an expert using domain knowledge. While this can work well when evaluating simple systems, growing complexities will deem that solution very difficult to implement. Increase in complexity of the systems led to growing popularity of data-driven (DD) approaches. In DD approach, the system or phenomena, is modelled automatically using the observed data. In order to do that, data is usually grouped first based on the observed similarities and patterns. This process is called clustering and will be briefly described in this section. Much like the fuzzy and boolean logic, clustering can be crisp or fuzzy. In crisp clustering each observed data point can only belong to one cluster. An example of the crisp clustering method is k-means. In fuzzy clustering a point belongs to a cluster with a degree of membership, which can depend on variety of factors, most often the distance to the cluster centre. An example of fuzzy clustering can be c-means clustering method or subtractive clustering. In addition, an example of the application of the c-means algorithm in forecasting will be presented below. Both approaches will be briefly discussed in the following sections.

3.4.1 k-means clustering

The k-means clustering method has been developed by Hartigan (1975). Later, faster and optimized version was introduced by the same author (Hartigan & Wong 1979). k-means is a crisp clustering method, meaning that each sample belongs to only one cluster. It partitions the data into k mutually exclusive clusters. Data points are partitioned so that they are as close to each other as possible within the cluster, and as far as possible from data points assigned to other clusters. Each cluster is characterized by its members (the data points belonging to the cluster) and the centroid (or centre). The centre is the point at which the sum of distances

to each member of the cluster is minimized. k-means uses a recursive algorithm that minimizes the sum of distances from each data point to its cluster centre, over all clusters. The algorithm moves data points between clusters until the sum cannot be decreased further. As a result a set of clusters are created that are as compact and well-separated as possible.

The algorithm can be described in 4 steps:

Algorithm 1 k-means clustering algorithm

1: Initialize k centroids by positioning them in the data space. The initialization process can be random or using an initialization method, such as Random Partition (Hamerly & Elkan 2002);
2: **while** centroids keep moving **do**
3: Assign each data point to the closest centroid;
4: Recalculate the position of each k centre by calculating the mean of all of the points assigned to that centre;
5: **end while**

An example of clustering of a random 200 data points with two input values x and y into $k = 3$ clusters can be seen below on Fig. 3.1.

Figure 3.1: k-means clustering algorithm.

3.4.2 c-means clustering

Fuzzy c-means algorithm is a clustering technique in which each data point belongs to a cluster with a certain degree of membership. This technique was originally introduced by Bezdek et al. (1984) as an improvement on earlier clustering methods. The fuzzy c-means clustering algorithm is very similar to k-means clustering. The difference is that rather than assigning the explicit cluster to the data sample it belongs to the cluster with a degree of membership depending on the distance to its centre.

The membership degree $\mu_i(x_k)$ is calculated at each iteration of the algorithm and is described by the following equation:

$$\mu_i(x_k) = \frac{1}{\sum_{l=1}^{R} \left(\frac{dist(c_i, x_k)}{dist(c_l, x_k)} \right)^{2/(m-1)}} \tag{3.5}$$

where x_k is the k^{th} data sample, R is the total number of clusters, $dist(c_i, x_k)$ is the distance between i^{th} cluster centre c_i and the data sample x_k and m is the fuzzifying factor. The larger the m the smaller the membership value resulting in fuzzier clusters. Assigning $m = 1$ causes c-means to become a crisp algorithm as the membership values will either be 1 or 0.

The centroid of the cluster c_i is calculated as a mean of all points weighted by their membership degree $\mu_i(x_k)$:

$$c_i = \frac{\sum_{j=1}^{k} \mu_i(x_j)^m x_j}{\sum_{j=1}^{k} \mu_i(x_j)^m} \tag{3.6}$$

The c-means clustering algorithm is presented below:

Algorithm 2 fuzzy c-means clustering algorithm

1: Choose a number of clusters R, and convergence threshold ε
2: Randomly assign R membership degrees to each data points
3: **while** change in μ_i between iterations $< \varepsilon$ **do**
4: Calculate the centre c_i of each cluster using Eq. 3.6;
5: For each point x_k calculate (Eq. 3.5) the new membership degree $\mu_i(x_k)$ based on the new cluster centres;
6: **end while**

Fig. 3.2 illustrates an example of the implementation of c-means algorithm. Here the points in black have not been assigned to any of the clusters because their maximum membership is not higher than 0.5. It can be seen that cluster centres are very similar to the ones obtained through k-means clustering.

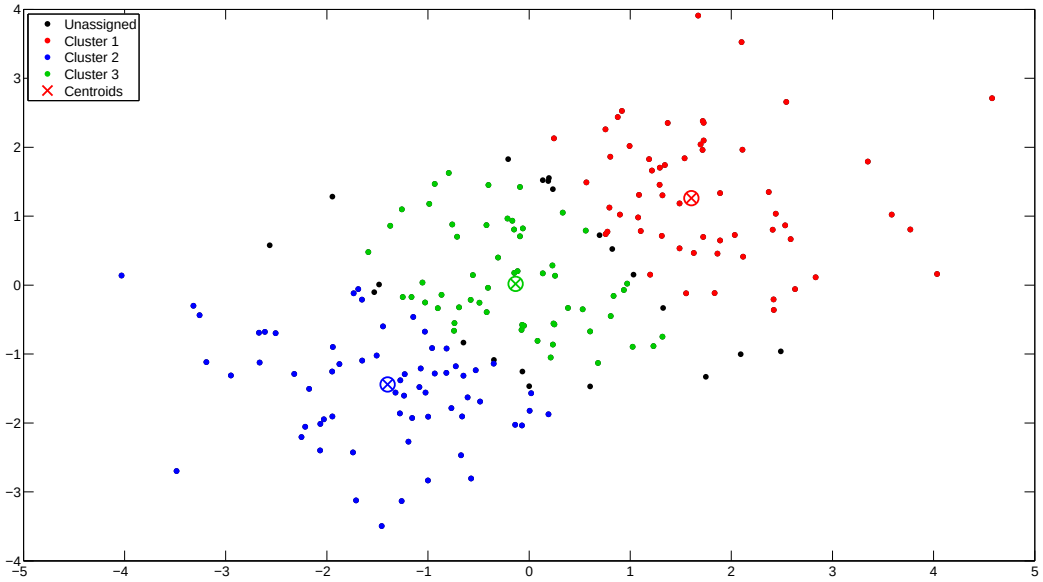

Figure 3.2: c-means clustering algorithm.

3.4.3 Fuzzy c-means algorithm for forecasting

The c-means clustering algorithm was modified and used in forecasting by Chen & Chang (2010), and later further assessed and evaluated in Birek et al. (2011).

The result of the c-means clustering algorithm is a set of cluster centres c_i and the membership degrees $\mu_i(X_k)$ of all of the k training samples X_k belonging to the clusters. This information can be used to construct the fuzzy If-then rules. Each cluster corresponds to a different fuzzy rule, therefore the total number of rules is equal to the number of clusters. The membership functions used in the fuzzy system are of triangle shape, due to the ease of implementation and low computational burden (Pedrycz 1994).

The characteristic values of triangular membership function are obtained based on the membership degrees of data vectors belonging to a cluster which forms the corresponding fuzzy rule. The centre of the triangular set c_{ij} (Eq. 3.7) is formed from the j^{th} input value x_{kj} of data vector X_k which has the highest membership degree $\mu_i(X_k)$. The lower a_{ij} and upper b_{ij} bounds are obtained from Eq. (3.8) and Eq. (3.9) for x_{kj} that are smaller and bigger then c_{ij} respectively.

$$c_{ij} = x_{kj} \text{ such that } \mu_i(X_k) = \max_{k=1,\ldots,h} \mu_i(X_k) \tag{3.7}$$

$$a_{ij} = \frac{\sum_{k=1}^{n} \mu_i(X_k)\, x_{kj}}{\sum_{k=1}^{n} \mu_i(X_k)} \text{ for } x_{kj} \leq c_{ij} \tag{3.8}$$

$$b_{ij} = \frac{\sum_{k=1}^{n} \mu_i(X_k)\, x_{kj}}{\sum_{k=1}^{n} \mu_i(X_k)} \text{ for } x_{kj} \geq c_{ij} \tag{3.9}$$

The fuzzy rules obtained from generated clusters allow generation of forecast based on the previous observation values which are organized into a data vector $X_k = [x_1\ x_2\ ...\ x_h]$. The observation values activate a rule, if all membership values of the triangular membership functions of this rule calculated by Eq. (A.7) are bigger than 0. If at least one rule is activated, the forecast is obtained using Eq. (3.10), where p is the total number of activated rules, $min\ A_i(x)$ is a minimum value of all of the membership values of i^{th} rule and dB_i is a defuzzified value of the consequent part of the activated rule.

$$F = \frac{\sum_{i=1}^{p} minA_i(x)\, dB_i}{\sum_{i=1}^{p} minA_i(x)} \tag{3.10}$$

If none of the rules are activated, the forecast is derived using a weighting ratio of all the rules with respect to the vector of the observation values X_k:

$$W_i = \frac{1}{\sum_{d=1}^{c} \left(\frac{\|X_k - D(dA_i)\|}{\|X_k - D(dA_d)\|} \right)^2} \tag{3.11}$$

where $\|X_k - D(dA)\|$ is the Euclidean distance between vector of the observed values X_k and a vector of defuzzified values $D(dA)$ of the fuzzy membership functions A of the antecedent part of the corresponding rule. The forecast is then obtained from (3.12) using the calculated weights and defuzzified values of the consequent parts of the corresponding fuzzy rules.

$$F = \sum_{i=1}^{c} W_i\, dB_i \tag{3.12}$$

3.4.4 Mountain method

The Mountain method was developed in order to facilitate the estimation of the initial number and position of the cluster centres for a fuzzy c-means clustering algorithm described in previous section. In the fuzzy c-means approach, coordinates of the cluster centres are obtained as a result of minimizing the cost function which takes into account the distance between cluster centres and data samples and the corresponding membership degrees. This, however, requires the number of clusters to be specified in advance.

The Mountain method divides the data space into a discrete grid. The rank of each grid point is calculated using the Mountain function (Eq. 3.13):

$$M(N_i) = \sum_{k=1}^{n} e^{-\alpha\, dist(x_k, N_i)} \tag{3.13}$$

The function takes into account the distance *dist* (calculated using the chosen distance measure, for example Euclidean distance) between the node N_i and all of the data points x_k, $k = 1, \ldots, n$ (α being a positive constant). Its value represents the potential $M(N_i)$ of the node N_i calculated as the density of the data points surrounding the node, which can be seen as the potential of that node to be a cluster centre. The grid point with the highest value of the potential is considered as the first cluster centre (Eq. 3.14):

$$M_1 = \max_i [M(N_i)] \tag{3.14}$$

At stage t the next cluster is found after eliminating the influence of the newly created centre at stage $t-1$. This is done through lowering the potential values of each node by the potential M_{t-1} of the newly formed cluster. The exponential function is used so that the nodes around the newly formed cluster have their mountain values reduced more than the ones further away. The function is also

bound by zero, so the result is always non-negative (Eq. 3.15):

$$M_t(N_i) = \max[M_{t-1}(N_i) - M_{t-1} \sum_{k=1,k\neq i}^{n} e^{-\beta\, dist(N_{k-1},N_i)}, 0] \tag{3.15}$$

New clusters are formed until the current level of maximum mountain value M_{t-1} becomes too low as compared to the original maximum M_1. This can be expressed as the ratio presented in (Eq. 3.16) with γ being a threshold parameter:

$$\frac{M_1}{M_{t-1}} < \gamma \tag{3.16}$$

The algorithm can be summarised in Algorithm 3:

Algorithm 3 Mountain method

1: Set parameters α, β and threshold γ;
2: Divide the data space and form a discrete grid with n nodes;
3: Calculate the mountain value for each of the N_i nodes (Eq. 3.13);
4: Form the first cluster in the node with the highest mountain value (Eq. 3.14);
5: **while** inequality Eq. 3.16 doesn't hold **do** {check the threshold value}
6: Modify the mountain values of other nodes and find another cluster centre (Eq. 3.15);
7: **end while**
8: Finish clustering;

Although the method proves to be simple and effective, its computational efficiency does not scale very well, as the complexity grows exponentially with dimension (number of variables). Higher dimension requires more nodes to be created, which has an impact on the performance as the mountain values need to be calculated for each node. This requires setting the resolution of the grid carefully so that both the quality of the solution and efficiency are considered.

3.5 Basic identification of Fuzzy Takagi-Sugeno models

As it has been described in the previous sections, Takagi-Sugeno models offer the transition between a conventional fuzzy rule-based model and a mathematical explanation of the system through a set of linear equations.

In this section the Subtractive clustering algorithm (Chiu 1994) will be described. It is a fuzzy model-based identification method which uses a modification of the Mountain method proposed by Yager & Filev (1994). The principles of the method are based on the identification of the fuzzy structure and estimation of parameters of the resulting linear equations. The structure identification is performed by the estimation of the focal points of the fuzzy rules through fuzzy clustering. The parameters of the consequent linear equations are obtained by applying a parameter estimation method based on least squares (LS).

3.5.1 Subtractive clustering fuzzy model identification method

The computational performance of the Mountain Method has been addressed in the Subtractive clustering algorithm (Chiu 1994), where the data points, and not the grid points, are considered for the cluster centre selection. This means that the number of points to be evaluated depends only on the amount of available data and is independent of the dimension of the data space. In addition, this eliminates the need to specify the grid resolution. The proposed method forms the basis for the fuzzy model identification algorithm. The cluster centres are automatically generated from the collected training data through calculation of the potential of data samples. The obtained cluster structure can then be used to build a fuzzy

model based on Takagi-Sugeno inference and a fuzzy weighted Least Squares (LS) estimation of the parameters of the consequent i.e., Then part of the rules.

For the data to be clustered correctly, the inputs need to be normalized so that the range of all the normalized data points $\tilde{x}_k \in [0, 1]$:

$$\tilde{x}_k = \frac{x_k - \min\limits_k[x_{k-1}]}{\max\limits_k[x_{k-1}] - \min\limits_k[x_{k-1}]} \tag{3.17}$$

$\tilde{x}_k$ and x_k may represent a vector if there is more than one input variable. The $\min[x_{k-1}]$ and $\max[x_{k-1}]$ denote the minimum and the maximum of all previously obtained data points over each dimension separately. The normalization is done to compensate for the differences in the range of the values in different dimensions (Dovzan et al. 2012), so that each dimension of the input data space is treated equally when computing the distance between inputs. For the sake of simplicity, the normalized input data will be described as $x_k = \tilde{x}_k$ in the further steps of the clustering part of the algorithm. However, when the parameters of the resulting linear equations are estimated, real data is used.

The way the potential of the data points is calculated is similar to the mountain function from the mountain method. The potential P of the training data point x_k is calculated as follows:

$$P(x_k) = \sum\nolimits_{l=1, l\neq k}^{n} e^{-\alpha\, dist(x_k, x_l)} \tag{3.18}$$

$$\alpha = \frac{4}{r_\alpha^2} \tag{3.19}$$

$$P_1 = \max_k[P(x_k)] \tag{3.20}$$

where n is the number of data points and r_α is a cluster radius, which defines the influence of the data samples on the potential. The choice of r_α can be based on the grid search over the set of values, typically in the range of $[0, 1]$, which yield

lowest training error. The literature review reports values between $[0.25, 0.5]$ as a good starting point (Chiu 1994). The data sample x_k with the highest potential P_1 is always chosen to be the first cluster centre $c_1 = x_k$.

After every step, the potential of the remaining data samples is updated to account for the new cluster centre. It is bound by 0, similar to Equation (3.15):

$$P(x_k) = \max[P(x_k) - P(c_i)e^{-\beta\, dist(x_k, c_i)}, 0] \tag{3.21}$$

$$\beta = \frac{4}{r_\beta^2} \tag{3.22}$$

where r_β is a radius defining the neighbourhood which will be impacted by the reduction of the potential. It is set to be greater than r_α (a good choice is $1.5r_\alpha$ according to Chiu (1994)) as this prevents closely placed data samples to be chosen as cluster centres. $P(c_i)$ is the potential of the newly obtained cluster centre c_i.

The clustering is considered to be finished when the ratio between the potential $P(x_k)$ of the currently considered data sample and the highest potential P_1 is lower than a certain threshold γ:

$$\frac{P(x_k)}{P_1} < \gamma \tag{3.23}$$

The value of γ affects the results considerably, as choosing it to be too small results in generating too many clusters, whereas if it is too large, not enough data samples will become cluster centres and consequently not enough fuzzy If-Then rules will be created (Yager & Filev 1994).

As it is difficult to establish a single value of γ that works well for all data patterns, Chiu (1994) propose alternative approach. Two values are used γ_{up} and γ_{down}. The γ_{up} specifies the threshold above which the data sample x_k will be accepted as a cluster centre. The value of γ_{down} specifies the threshold below which the sample will be rejected. Good starting values are $\gamma_{down} = 0.5$ and $\gamma_{down} = 0.15$ (Chiu 1994). The values in between are additionally checked if they provide a good

trade-off between their potential and the distance to the closest cluster centre. The assessment is done according to Algorithm 4.

Algorithm 4 Data sample assessment

if $\frac{P(x_k)}{P_1} > \gamma_{up}$ **then**
    Accepted x_k as a cluster and continue;
else if $\frac{P(x_k)}{P_1} < \gamma_{down}$ **then**
    Reject x_k and finish clustering;
else
    Find $dist_{min}$ - the shortest distance between x_k and all identified clusters;
    if $\frac{dist_{min}}{r_\alpha} + \frac{P(x_k)}{P_1} \geq 1$ **then**
        Accept x_k as a cluster and continue;
    else
        Reject x_k and set $P_k = 0$ and continue;
    end if
end if

The Subtractive clustering algorithm is summarized in Algorithm 5.

Algorithm 5 Subtractive clustering

Set the radius r_α, $r_\beta = 1.5r_\alpha$ and thresholds γ_{down} and γ_{up};
Calculate the potential of all available data points from (3.18 - 3.19);
Find the data point with the highest potential P_1 (3.20) and establish it as the first cluster centre;
Update the potential of the remaining data points from (3.21);
Choose new candidate cluster x_k by taking the remaining data point with the highest potential;
while not finished **do**
    Apply Algorithm 4;
    Update the potential of the remaining data points from (3.21);
    Choose new candidate cluster x_k by taking the remaining data point with the highest potential;
end while

When the clustering process is finished, the resulting structure can be used to generate fuzzy If-Then rules. Each cluster corresponds to one fuzzy rule with the cluster centre coordinates being the focal points of Gaussian membership functions of the antecedent, i.e. If part of the fuzzy rule. The consequent, i.e., Then part is

in the form of a linear function and Takagi-Sugeno inference is applied. Rule i has the following form:

$$\begin{aligned} &\textbf{IF } x_1 \textbf{ is } A_{i1} \textbf{ AND } x_2 \textbf{ is } A_{i2} \textbf{ AND } \ldots \textbf{ AND } x_h \textbf{ is } A_{ih} \\ &\textbf{THEN } y_i = a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{ih}x_h + a_{ih+1} \end{aligned} \tag{3.24}$$

where $x = \begin{bmatrix} x_1 & x_2 & \ldots & x_h \end{bmatrix}$ is a data sample, $j = 1, \ldots, h$ is an index of h input values and A_{ij} represents a fuzzy set with the Gaussian membership function:

$$f(x_j, c_{ij}, \sigma_{ij}^2) = e^{-\frac{dist(x_j, c_{ij})}{2\sigma_{ij}^2}} \tag{3.25}$$

with c_{ij} being the centre and σ_{ij}^2 a variance (controls the width of the distribution) of the Gaussian membership function.

The parameters of all linear functions of the consequent parts of fuzzy rules are estimated using the Least Squares algorithm. The activation degree μ_i of each rule is obtained as the membership degree of inputs of the considered data point x_k (note that x_k and c_i are vectors, therefore $dist$ represents the Euclidean distance) belonging to the corresponding Gaussian membership functions (3.25).

$$\mu_i(x_k) = e^{-\frac{dist(x_k, c_i)}{2\sigma_i^2}} \tag{3.26}$$

The firing degree τ_i of each rule, i.e. the degree to which the rule is used as compared to other fuzzy rules, is then obtained as follows:

$$\tau_i = \frac{\mu_i}{\sum_{l=1}^{R} \mu_l} \tag{3.27}$$

where R is the total number of fuzzy rules (clusters). The firing degrees are

combined with input values to create the data matrix Ψ used in the LS algorithm.

$$\Psi = \begin{bmatrix} \tau_1 x_1^e & \tau_2 x_1^e & \cdots & \tau_R x_1^e \\ \vdots & \vdots & \vdots & \vdots \\ \tau_1 x_n^e & \tau_2 x_n^e & \cdots & \tau_R x_n^e \end{bmatrix} \tag{3.28}$$

where x_k^e is an extended data vector with h input values and an additional dimension to account for a parameter from a θ vector:

$$x_k^e = \begin{bmatrix} x_{k1} & x_{k2} & \cdots & x_{kh} & 1 \end{bmatrix} \tag{3.29}$$

The $\Theta = \begin{bmatrix} \theta_1^T & \theta_2^T & \cdots & \theta_R^T \end{bmatrix}$ consists of vectors of estimated parameters of the consequent part of each rule where $\theta_1 = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1h} & a_{1h+1} \end{bmatrix}$, $\theta_2 = \begin{bmatrix} a_{21} & a_{22} & \cdots & a_{2h} & a_{2h+1} \end{bmatrix}$, …, $\theta_R = \begin{bmatrix} a_{R1} & a_{R2} & \cdots & a_{Rh} & a_{Rh+1} \end{bmatrix}$. The parameters are chosen so that the sum of squared errors between real output values Y and the outputs estimated by using fuzzy rules $\Psi^T\Theta$ is the lowest:

$$J = \left(Y - \Psi^T\Theta\right)^{-1}\left(Y - \Psi^T\Theta\right) \tag{3.30}$$

The parameters can be estimated by the pseudo inverse:

$$\Theta = (\Psi^T\Psi)^{-1}\Psi^T Y \tag{3.31}$$

Knowing the values of the dependent variables (inputs) of extended data vector x^e, the generated model can be used to predict the output based on the assessment of the firing degrees τ_i using the estimated parameters Θ. The output $\hat{y}$ is estimated as:

$$\hat{y} = \sum\nolimits_{i=1}^{R} \tau_i \theta_i x^e \tag{3.32}$$

The fuzzy model identification method with the use of Subtractive clustering

algorithm is presented in Algorithm 6:

Algorithm 6 Subtractive clustering fuzzy model identification method

1: Find all cluster centres using the Subtractive clustering Algorithm 5;
2: Establish each cluster centre as an antecedent part of a fuzzy rule as in (3.24);
3: Calculate the degree μ_i at which each of the rules is activated by the available data from (3.26);
4: Obtain the firing degree of each rule from (3.27);
5: Build a data matrix Ψ (3.28);
6: Obtain the parameters of the consequence linear equations by using Least Squares optimization (3.30 - 3.31);
7: The output of each of the linear functions can then be estimated based on the provided input values and the resulting firing degrees of the activated fuzzy rules (3.32);

Subtractive clustering can be used as an initialization method for other clustering algorithms, such as fuzzy c-means, which require the number and initial position of clusters (these are usually chosen randomly) to be set. It can also be used as an approximate clustering algorithm on its own. The presented combination of the Subtractive clustering algorithm and Least Squares regression allows for automatic generation of the Takagi-Sugeno fuzzy models. The algorithm is robust and can work with noisy data due to the nature of the fuzzy systems (through firing degree calculation from Equation 3.27).

3.6 Conclusions

In this chapter the idea of automatic generation of fuzzy models has been introduced. This could be achieved by combining the data clustering approach and fuzzy Takagi-Sugeno inference. The main drawback of the presented algorithm is that for the estimation to take place all data needs to be available. Therefore, the conditions under which the system operates need to be constant for the generated model to keep its validity. If the conditions change, and the behaviour has not

been captured by the data before, the whole model needs to be generated from scratch (Angelov 2004*a*). The need for an adaptive version of the presented algorithm arose, as some applications (e.g. forecasting or control engineering) include systems that are dynamically changing. In the next chapter the idea of evolving fuzzy systems will be introduced which overcomes that drawback by dynamically building and evaluating fuzzy systems as new data becomes available.

Chapter 4

Evolving fuzzy identification methods and their applications

4.1 Introduction

In the previous chapter the concept of fuzzy systems has been introduced. Initially the fuzzy rules were generated based on the engineering knowledge and required manual input. The developments in the area of fuzzy clustering methods allowed for automatic rule creation from the data based on the position and spread of the cluster centres. One of the drawbacks of that approach is that the rules could only be created once all, or significant portion, of the data has been collected. It was very difficult or almost impossible to easily extend the model once new data became available.

The aim of this chapter is to introduce the reader to the concept of evolving fuzzy systems which overcomes that drawback. Evolving fuzzy systems allow for recursive generation and update of both antecedent and consequent parts of fuzzy

rules. This is achieved by using automatic on-line fuzzy rule creation based on the recursive clustering techniques. The model adapts itself based on the incoming data, without the need to manually build and estimate the parameters of the output functions. This approach has application to real-time systems or problems with non-linear, non-stationary data, since those can be approximated by a set of linear models (Wang & Mendel 1992). The approach is also applicable to problems in which relationships between different inputs and an output is not well understood and can't be easily modelled.

Three main families of evolving fuzzy algorithms will be presented in this chapter: the eTS and its extensions, FLEXFIS and DENFIS. The advantages and disadvantages of those methods will be considered and some applications will be demonstrated.

4.2 Notation

As a lot of algorithms will be presented in this chapter, it seems viable to provide the reader with a notation which will be used to describe them. Some clarification will be provided for the situations when the same notation is used to describe different variables.

	Common symbols
k	time instance
i	cluster, rule index
R	total number of clusters, rules
j	index of the input
h	total number of inputs
n	total number of data samples
m	total number of outputs (usually $m = 1$)
σ^2	variance, or width of the Gaussian membership function
σ	standard deviation

A_{ij}	an j^{th} fuzzy set of rule i
$x_k = \begin{bmatrix} x_{k1} & x_{k2} & \dots & x_{kh} \end{bmatrix}$	data sample
$x_k^e = \begin{bmatrix} x_{k1} & x_{k2} & \dots & x_{kh} & 1 \end{bmatrix}$	extended data sample
$\tilde{x}_k$	normalized or standardized data sample
$\overline{x}_k$	mean of the k data samples
$\hat{y}$	estimated output
$c_i = \begin{bmatrix} c_{i1} & c_{i2} & \dots & c_{ih} \end{bmatrix}$	cluster centre
c_{win}	winning (for example closest to the data sample) cluster
$dist(a, b)$	distance measure between two points or vectors, usually Euclidean distance
$r_{k,ij}$	radius of cluster i, input j at time instance k
$P(x_k)$	potential of data sample x_k
$P_k(x_k)$	potential of data sample x_k at time instance k
μ_i	activation degree of rule i
τ_i	firing degree of rule i
	Parameter estimation
ψ_i	regressor vector of rule i
Ψ	data matrix of regressor vectors ψ
a_{ij}	parameter of j^{th} input for rule i
$\theta_i = \begin{bmatrix} a_{i1} & a_{i2} & \dots & a_{ih+1} \end{bmatrix}$	vector of parameters for rule i
Θ	matrix of parameter vectors θ
Q	high number used to initialise the covariance matrix
Cov_k	covariance matrix at time instance k
cov_{12}	for global learning, an element of global covariance matrix at row 1, column 2 (example)
cov_{ik}	for local learning, a local covariance matrix of rule i at time instance k
L	gain vector
$\hat{\varepsilon}$	one step ahead prediction error
	eTS algorithms

σ_k, ϑ_k, β_{kj}, ν_k	components of recursive potential calculation equation
$S_k(x_k)$	scatter of data sample x_k at time k
$N_k(c_i)$	population of cluster c_i at time k
$age_k(c_i)$	age of cluster c_i at time k
$At_k(c_i)$	accumulated time of arrival of cluster c_i at time k
I_l	time instance of the l^{th} data sample added to the cluster
$U_k(c_i)$	utility of the fuzzy rule i at time k
$L_k(c_i)$	local density in eTS+
γ	parameter used to weight the influence of the radius
ω_j	ratio of contribution of the input j
ϵ	threshold for ignoring the rules based on the population
	FLEXFIS algorithms
ρ	parameter used for cluster assessment
k_{c_i}	time index of creation of the new cluster c_i
ϵ	parameter used to initiate the range of the clusters
η	learning gain
	DENFIS algorithms
D_{thr}	clustering threshold
λ	forgetting factor

4.3 Evolving Takagi-Sugeno (eTS) family of algorithms

The eTS family of algorithms is based on the Subtractive clustering algorithm described in the previous chapter. The modification to the standard off-line algorithm allows for recursive clustering and updating of parameters (fuzzy

evolving model identification) as new data becomes available. Various modification and extensions of the standard Evolving Takagi-Sugeno fuzzy identification method were introduced over the years. They include better rule management, different ways of regulating the cluster structure and modifications to the Recursive Least Squares method for parameter estimation. The more important extensions to the standard eTS algorithm will be described and the methods will be compared at the end of the section together with some examples of the various applications to real world problems which will be highlighted at the end of this chapter.

4.3.1 Evolving Takagi-Sugeno (eTS) algorithm

The Evolving Takagi-Sugeno (eTS) algorithm (Angelov & Filev 2004, Angelov 2004*b*) is based on the Subtractive clustering algorithm described in the previous chapter, with modifications which allow the gradual update of the antecedent part of the fuzzy If-Then rules, as well as the consequent parameters via a modified Recursive Least Squares (RLS) algorithm. The algorithm can be initiated with the already generated rules (e.g. through off-line identification of a fuzzy model based on Subtractive clustering) or starting from the first data sample. The high level diagram of the algorithm can be seen below:

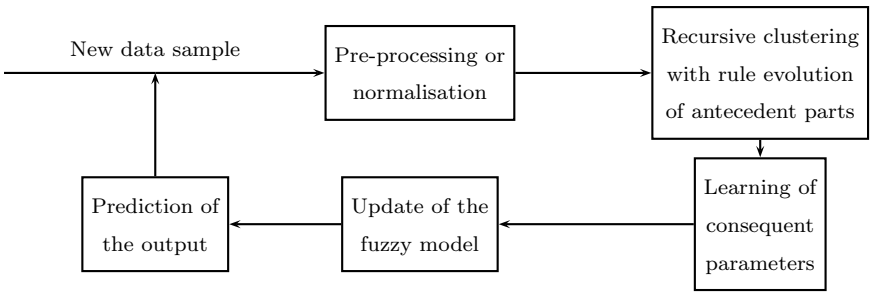

Figure 4.1: High level diagram of fuzzy evolving algorithm.

4.3.1.1 Data normalization in eTS

Similarly to the Subtractive algorithm (3.17), the data needs to be normalized in advance, before the clustering can begin. When the data becomes available gradually, the information of the maximum and minimum values may not be available immediately. It is however often possible to estimate the maximum and minimum values from some subset (training set) of available data or those values can be chosen judgementally based on the expertise and knowledge about the system. The incoming data is therefore normalized in the same manner as in the Subtractive clustering algorithm using Equation 3.17.

4.3.1.2 Recursive potential calculation

For the algorithm to be viable in on-line applications, the potential Equation (3.18) used in the Subtractive clustering algorithm needs to be modified so that it can be updated recursively. The potential of the data samples in eTS is measured using a Cauchy type function, which is an approximation of the Gaussian kernel. It has similar characteristics, in that it is monotonic and inversely proportional to the distance, and after modification allows for the recursive calculation. The formula for the potential $P_k(x_k)$ in the step k is given by the Cauchy function:

$$P_k(x_k) = \frac{k-1}{(k-1)(\vartheta_k + 1) + \sigma_k - 2\nu_k} \tag{4.1}$$

where:

$$\begin{aligned} \vartheta_k &= \sum_{j=1}^{h+1} (x_{kj}^e)^2 \\ \sigma_k &= \sum_{l=1}^{k-1} \sum_{j=1}^{h+1} (x_{lj}^e)^2 = \sigma_{k-1} + \vartheta_{k-1} \\ \nu_k &= \sum_{j=1}^{h+1} x_{kj}^e \beta_{kj} \\ \beta_{kj} &= \sum_{l=1}^{k-1} x_{lj}^e = \beta_{k-1j} + x_{k-1j}^e \end{aligned} \tag{4.2}$$

The explanation of the above equation and the derivation can be found in Angelov & Filev (2004).
If the algorithm starts from an empty rule base and the first data sample x_1 is considered, then the following initialization is performed:

1. The first data sample becomes the first cluster centre $x_1 : c_1 = x_1$;
2. The potential $P_1(c_1)$ of the first cluster is set to 1: $P_1(c_1) = 1$;
3. The values of the parameters (4.2) used to calculate the potential (4.1) are initialized as follows:
   (a) $\vartheta_1 = 0$;
   (b) $\sigma_1 = 0$;
   (c) $\nu_1 = 0$;
   (d) $\beta_1 = [\beta_{11} \quad \beta_{12} \quad \cdots \quad \beta_{1j} \quad \cdots \quad \beta_{1h+1}] = [0 \quad 0 \quad \cdots \quad 0]$;

Similarly to Subtractive clustering (Eq. 3.21), the potential has to be updated for the already generated cluster centres after each step, as it depends on the distances to all currently available data samples. The update is performed in a recursive way:

$$P_k(c_i) = \frac{(k-1) P_{k-1}(c_i)}{k - 2 + P_{k-1}(c_i) + P_{k-1}(c_i) \sum_{j=1}^{h+1} (c_{ij} - x_{kj})^2} \tag{4.3}$$

where $P_k(c_i)$ is the potential of the k recursive step of the i^{th} cluster centre c_i. The derivation of the above equations can be obtained from Angelov & Filev (2004).

4.3.1.3 Recursive cluster assessment

Due to the fact that the data becomes available gradually, there is a need for the process of deciding when to add a new cluster, change the already existing one or when to leave the cluster structure unchanged. This process is based on the

values of the potentials, distance between cluster centres and new data sample and cluster radius r. The potential of the candidate cluster $P_k(x_k)$ is compared with the updated potentials of all previously selected clusters using the Algorithm 7:

Algorithm 7 Decision algorithm for recursive cluster assessment in the eTS

1: **if** $P_k(x_k) > \max_i[P_k(c_i)]$ **then**
2: calculate $dist_{\min} = \min_i \left[\|c_i - x_k\|^2\right]$ - the minimum distance between the data sample x_k and the cluster centre c_i;
3: **if** $\frac{P_k(x_k)}{\max_i[P_k(c_i)]} - \frac{dist_{\min}}{r} \geq 1$ **then**
4: The closest cluster centre c_i is replaced by the current data sample x_k;
5: The potential of the changed cluster is replaced by the potential $P_k(x_k)$ of the data sample x_k;
6: **else**
7: New cluster $c_{(R+1)}$ is added with the coordinates of data sample x_k and the potential $P_k(x_k)$;
8: **end if**
9: **else**
10: Ignore the data sample x_k and proceed further;
11: **end if**

The meaning of the condition in step 3 of the above algorithm is similar to the one in Algorithm 4 at step 7. The trade-off between the potential $\frac{P_k(x_k)}{\max_i[P_k(c_i)]}$ and the distance $\frac{dist_{\min}}{r}$ to the closest cluster centre is checked to assess if the data sample should replace the closest cluster or if the new cluster centre should be created in place of x_k.

4.3.1.4 Recursive rule generation - antecedent parts

When the new cluster c_i is added, it automatically generates a Takagi-Sugeno fuzzy If-Then rule as presented in (3.24) with antecedent part described by the Gaussian membership function obtained from (3.25). The consequence part of the rule is obtained via a modified RLS algorithm which will be described in the next subsection. When the clusters are replaced (see Algorithm 7) the antecedent part of the rule is also replaced and the new membership function is defined by the new coordinates from x_k.

4.3.1.5 Recursive rule generation - consequence linear equation parameter estimation

The recursive process also involves the on-line estimation of the parameters of the consequence linear equations. This is done through the use of the Recursive Least Squares (RLS) algorithm. The estimation can be performed on the global and local level. In the global learning one covariance matrix for all rules is considered, whereas in the local learning each rule is described by a different locally optimized covariance matrix. This introduces a variation in the algorithm which will be presented below. In general, the local learning proves to have more locally interpretable rules and sometimes may produce better results (Angelov et al. 2010). The algorithm is initialized in the following manner:

1. The parameter vector θ_1 of the consequent linear equation of the first rule is set to: $\theta_1 = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^{h+1}$, where $h+1$ indicates the number of columns in the vector (h being the number of inputs). In case of global learning, the parameter matrix Θ of the consequent linear equations is set to θ_1, $\Theta = \theta_1$. The local learning operates on the set of locally optimized parameters, therefore there is no need to use the global matrix.

2. The fuzzy weight τ_1 of the first rule is set to 1.

3. The covariance matrix Cov_1 (or cov_1 for local learning, although at this stage they are the same) is initialized with a high number Q multiplied by the $(h+1) \times (h+1)$ identity matrix:

$$Cov_1 = cov_1 = \begin{bmatrix} Q & 0 & \cdots & 0 \\ 0 & Q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Q \end{bmatrix}^{(h+1)\times(h+1)} \tag{4.4}$$

4. The regressor vector ψ_1 which will store weighted input values for the parameter estimation purpose is initialized with: $\psi_1 = \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix}^{h+1}$, $\Psi_1 = \psi_1$ for global learning.

The size of the covariance matrix will depend on the number of extended inputs $h+1$ (h being number of inputs) and, for global covariance matrix, on the number of existing rules R. Therefore, the old global covariance Cov_{k-1} will have a size of $R(h+1) \times R(h+1)$. When a new rule is added, the global covariance matrix needs to be resized and reset as the addition affects the already existing rule structure:

$$Cov_k = \begin{bmatrix} \rho cov_{11} & \cdots & \rho cov_{1R(h+1)} & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & 0 & \cdots & 0 \\ \rho cov_{R(h+1)1} & \cdots & \rho cov_{R(h+1)R(h+1)} & 0 & \cdots & 0 \\ 0 & 0 & 0 & Q & \cdots & 0 \\ \cdots & \cdots & \cdots & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & Q \end{bmatrix} \tag{4.5}$$

where R is the number of rules before addition, cov are elements of the old covariance matrix, and $\rho = (R^2+1)/R^2$ is the resetting factor.

In the local learning the parameters of the consequence part of each rule are described by separate, local covariance matrices cov_i. When the new rule is added, the new covariance matrix is initialized in the same way as in (4.4).

Adding a new rule also requires setting the initial parameter vector $\theta_{(R+1)}$ for the consequence linear equation. This is done (for both global and local learning) by weighted averaging parameters of all already generated rules with a weight set by the firing degree τ_i (3.27) of each of the rules:

$$\theta_{(R+1)} = \sum_{i=1}^{R} \tau_i \theta_i \tag{4.6}$$

When the rule is updated (replaced by the new cluster centre) the parameters and the covariance matrix are inherited by the newly generated cluster for both local and global learning.

After that, the regressor vector ψ_i is calculated for each rule using the firing degrees $\tau_1, \tau_2, \ldots, \tau_R$ obtained from Eq. 3.27 and the extended input data sample x_k^e (Eq. 3.29):

$$\psi_i = \begin{bmatrix} \tau_i x_{k1} & \tau_i x_{k2} & \cdots \tau_i x_{kh} & \tau_i \end{bmatrix} \tag{4.7}$$

where $i = 1, \ldots, R$.

In the case of global learning, the regressor vectors of all rules are additionally combined into a data matrix Ψ_k:

$$\Psi_k = \begin{bmatrix} \psi_1^T & \psi_2^T & \ldots & \psi_R^T \end{bmatrix}^T \tag{4.8}$$

The recursive update of the parameters differs between global and local learning. For global learning, the RLS estimation of the parameters of the linear consequent

part of the fuzzy If-Then rules is applied as follows:

$$\begin{aligned} L &= \frac{Cov_k \Psi_k}{1 + \Psi_k^T Cov_k \Psi_k} \\ \hat{\varepsilon} &= y_k - \Psi_k^T \Theta_{(k-1)} \\ \Theta_k &= \Theta_{(k-1)} + L\hat{\varepsilon} \\ Cov_{(k+1)} &= Cov_k - L\Psi_k^T Cov_k \end{aligned} \tag{4.9}$$

where Cov_k is a global covariance matrix, Ψ_k is a global data matrix and $\Theta_{(k-1)}$ is a global matrix of previously estimated parameters at step $k-1$ $\Theta_{(k-1)} = \begin{bmatrix} \theta_1^T & \theta_2^T & \cdots & \theta_R^T \end{bmatrix}$ and k is the current time step. L is referred to as gain vector and $\hat{\varepsilon}$ as one-step ahead prediction error. Detailed derivation of the RLS algorithm can be found in the book by Wellstead & Zarrop (1991).

In the local learning the following algorithm is performed for each rule:

$$\begin{aligned} L &= \frac{cov_{k,i} x_k^{eT} \tau_i}{1 + \psi_{k,i}^T cov_{k,i} x_k^{eT}} \\ \hat{\varepsilon} &= y_k - x_k^{eT} \theta_{(k-1),i} \\ \theta_{k,i} &= \theta_{(k-1),i} + L\hat{\varepsilon} \\ cov_{(k+1),i} &= cov_{k,i} - L x_k^{eT} cov_{k,i} \end{aligned} \tag{4.10}$$

The estimated output for the next period based on the input values and obtained parameter estimates is given by Equation (3.32).

The whole algorithm is summarized below:

Algorithm 8 Evolving Takagi-Sugeno algorithm

if First data sample **then**
    Initialize the first cluster and the algorithm parameters;
    Initialize the parameters θ_1 of the consequent linear equation, the covariance matrix Cov_1 (cov_1 for local learning) and the regressor vector ψ_1;
    Create the first fuzzy rule, where the antecedent If-Else part is based on the cluster centre c_1 and the parameters of the consequence linear equation are set to θ_1, set the weight τ_1 to 1;
else
    Calculate the potential of the new data sample x_k through (4.1);
    Update the potential of already existing clusters through (4.3);
    Using Algorithm 7 decide if the new data should become a new cluster, replace already existing one or if it should be ignored;
    if Global learning **then**
        if New rule **then**
            Resize and reset the global covariance matrix Cov_k (4.5);
            Set the initial parameters of the new rule $\theta_{(R+1)}$ (4.6);
        else if Replace the rule **then**
            Inherit the covariance matrix and the parameters of the consequence equation from the rule which is being replaced;
        end if
        Calculate the regressor vector ψ_i for each rule and combine into a data matrix Ψ_k;
        Estimate the global parameter values Θ_k for all rules through (4.9);
    else if Local learning **then**
        if New rule **then**
            Initialize the local covariance matrix $cov_{k,(R+1)}$ (4.4);
            Set the initial parameters of the new rule $\theta_{(R+1)}$ (4.6);
        else if Replace the rule **then**
            Inherit the covariance matrix and the parameters of the consequence equation from the rule which is being replaced;
        end if
        Calculate the regressor vector ψ_i for each rule;
        Estimate the local parameter values $\theta_{k,i}$ for each rule separately (4.10);
    end if
    Obtain the output estimate (3.32);
end if
Read next data sample

The block diagram of the algorithm is shown in Fig. 4.2

Figure 4.2: Block diagram of the Evolving Takagi-Sugeno algorithm.

4.3.1.6 Summary

The eTS algorithm brings a significant improvement to a Subtractive clustering algorithm in terms of the usability and applicability to dynamic systems. The research over the past few years resulted in numerous extensions to the standard eTS algorithm, such as simplification of the potential calculation, dynamic cluster radii, better rule validation and control or improvements to recursive data

normalization to mention the few. Some of those improvements will be presented below.

4.3.2 Simplified Evolving Takagi-Sugeno (Simpl_eTS) algorithm

The simpl_eTS method was introduced by Angelov & Filev (2005) in order to improve the computational speed of the eTS learning algorithm. The improvement was achieved by implementing the scatter calculation instead of the potential. The scatter is somewhat similar to the way potential is calculated in eTS, but it improved the computational efficiency. Additionally, the Gaussian membership functions have been replaced by the Cauchy type function and the concept of age of the rules have been introduced in order to improve the readability of the rule structure.

4.3.2.1 Recursive data normalization

The differences between the data samples are calculated through the Euclidean distance; therefore it is important to normalize the data before the clustering step. This is easily achieved in batch algorithms, where all of the data is available and can be normalized based on the mean and the variance, but in the on-line case some additional effort is required.

In simpl_eTS the data is standardized based on the recursive calculation of mean and standard deviation, with the initial values of mean $\overline{x}_1$ and variance σ_1^2 equal to 0. The recursive formulae for mean and variance are presented below:

$$\begin{aligned} \overline{x}_k &= \frac{k-1}{k}\overline{x}_{(k-1)} + \frac{1}{k}x_k \\ \sigma_k^2 &= \frac{k-1}{k}\sigma_{(k-1)}^2 + \frac{1}{k-1}\left(x_k - \overline{x}_k\right)^2 \end{aligned} \tag{4.11}$$

Having those values, we can now obtain the standardized value $\tilde{x}_k$ of the data sample x_k:

$$\tilde{x}_k = \frac{x_k - \overline{x}_k}{\sigma_k} \tag{4.12}$$

where $\overline{x}_k$ is previously calculated mean of all obtained data samples and σ_k is a standard deviation obtained by taking a square root of calculated variance σ_k^2. In the following calculations it is assumed that the data sample x_k has been already normalized to simplify the notation.

4.3.2.2 Use of Cauchy type function

The Cauchy type function, introduced in Subsection 4.3.1.2 replaces the Gaussian membership function for the antecedent parts of the rule. The motivation to use this function comes from the fact that it is an approximation of the Gaussian kernel and is faster to calculate, due to lack of exponential function. This significantly improves the computational time (Angelov & Filev 2005), which is especially important for real-time applications:

$$\mu_i(x_k) = \frac{1}{\prod_{j=1}^{n}\left(1 + \left(\frac{2(x_{k,j} - c_{ij})}{r}\right)^2\right)} \tag{4.13}$$

where j denotes the input index, n is the total number of inputs, $x_{k,j}$ is the j^{th} input of the k^{th} data sample, c_{ij} is the j^{th} input of i^{th} cluster centre and r is radius of the cluster defining area of its influence. In general, the smaller the radius, the lower the activation degree of the membership function (but it also heavily depends on the distance of the data sample from the cluster centre).

4.3.2.3 Use of Scatter

In simple_eTS the importance of the incoming data samples is described by their scatter rather than a potential. This change is also justified by the decrease in the computation time while maintaining similarity to the potential (Eq. 3.18) in the subtractive algorithm. In the formula below, n is the number of input values and 1 accounts for one output. Although it is possible to have m number of outputs, multi-output systems are not in the scope of this thesis. The off-line scatter at time k is given by:

$$S(x_k) = \frac{1}{k(n+1)} \sum_{l=1}^{k} \sum_{j=1}^{n+1} (x_{l,j} - x_{k,j})^2 \tag{4.14}$$

The scatter $S(x_k)$ represents the average distance of the data sample x_k to all other data samples.

To be able to use it in the evolving algorithm, the notion of scatter has to be calculated in a recursive manner. The formula used in the simpl_eTS to calculate the importance of the new upcoming data recursively, similar to Eq. (4.1), is given by:

$$S_k(x_k) = \frac{1}{(k-1)(n+1)} \left((k-1) \sum_{j=1}^{n+1} x_{k,j}^2 - 2 \sum_{j=1}^{n+1} x_{k,j} \beta_{k,j} + \vartheta_k \right) \tag{4.15}$$

where:

$$\begin{aligned} \beta_{k,j} &= \beta_{(k-1),j} + x_{(k-1),j}, \\ \vartheta_k &= \vartheta_{(k-1)} + \sum_{j=1}^{n+1} x_{k,j}^2 \end{aligned} \tag{4.16}$$

The scatter for all existing cluster centres is also updated after each new data

sample becomes available through the Eq. (4.17):

$$S_k(c_i) = \frac{k-2}{k-1} S_{k-1}(c_i) + \sum_{j=1}^{n+1} \left(x_{k,j} - x_{(k-1),j}\right)^2 \tag{4.17}$$

It is worth noting that for all of the equations involving scatter calculation, the division is always performed on the simple integer values which leads to simplification in computational calculations as compared to original eTS.

4.3.2.4 Recursive cluster assessment in simpl_eTS and population of the clusters

The scatter calculation is not the only characteristic which describes the clusters. Additionally, each cluster c_i maintains the information on how many data samples belong to it in step k, as well as store the information about their age. The population of the cluster $N_k(c_i)$ is increased by one if the data sample is close enough to the cluster i:

$$N_k(c_i) = N_{(k-1)}(c_i) + 1; \tag{4.18}$$

The age of the cluster $age_k(c_i)$ describes the accumulated time of arrival of the data sample. If the new data sample is in close proximity to an already existing centre, the value of the $age_k(c_i)$ is going to increase by k:

$$age_k(c_i) = age_{(k-1)}(c_i) + k \tag{4.19}$$

Similar to standard eTS, after the scatter calculation the new data sample can become a new cluster, replace the already existing cluster centre or be assigned to the nearest cluster (this step is new, in eTS the data sample was ignored). The following algorithm presents the decision making process:

Algorithm 9 Decision algorithm for recursive cluster assessment in the simpl_eTS

if $S_k(x_k) < \min_i[S_k(c_i)]$ **OR** $S_k(x_k) > \max_i[S_k(c_i)]$ **then**
  calculate $dist_{\min} = \min_i \left[\|c_i - x_k\|^2\right]$ - the minimum Euclidean distance $\|\cdot\|^2$ between the data sample x_k and the cluster centre c_i;
  if $dist_{\min} < 0.5r$ **then**
    The closest cluster centre c_i is replaced by the current data sample x_k;
    The scatter of the changed cluster is replaced by the scatter $S_k(x_k)$ of the data sample x_k;
    Increase the age of the cluster (4.19);
    Increase the population of the cluster (4.18);
  else
    New cluster c_R is added with the coordinates of data sample x_k and the scatter $S_k(x_k)$;
    The age and the population of the new cluster is initiated: $age_k(c_R) = k$, $N_k(c_R) = 1$;
  end if
else
  The data sample x_k is assigned to the nearest cluster i, $i : \min_{i=1}^{R} \|x_k - c_i\|^2$;
  Increase the age of the cluster (4.19);
  Increase the population of the cluster (4.18);
end if

Additionally the population of the clusters is monitored, and when no more new samples are added to a certain cluster over a certain period of time, i.e. when the population N_k of the cluster c_i drops below 1% of the total number of N data samples the cluster c_i is ignored by setting the firing rule τ_i to 0:

$$\textbf{IF } \frac{N_k(c_i)}{N} < 0.01 \textbf{ THEN } \tau_i = 0 \tag{4.20}$$

The estimates of the parameters of the consequence linear equations are obtained in the same fashion as in the standard eTS algorithm, including the possibility of using global and local learning.

The algorithm is summarized below:

Algorithm 10 Simpl_eTS algorithm

Read x_k;
Normalize x_k using (4.11) - (4.12)
if $k == 1$ **then** {first data sample, initialize the structure}
    $S_1(x_1) = 0$, $\beta_1 = \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix}$, $\vartheta_1 = 0$;
    $c_1 = x_1$, $N_1(c_1) = 1$, $age_1(c_1) = 1$;
    $R = 1$; {Initialize first fuzzy rule}
    Initialize parameters of consequence equation (same as in eTS);
else {additional data sample, evolve the structure}
    Calculate $S_k(x_k)$ using (4.15);
    Update $S_k(c_i)$ of each centre using (4.17);
    Apply Algorithm 9 to update the cluster structure accordingly;
    Generate rule base from the cluster structure (same as eTS);
    Apply rule (4.20) to reduce the rule base if necessary;
    Estimate the parameters of consequent linear equations (local or global learning). Same procedure as in eTS (4.4) - (4.10);
end if

4.3.3 Extended Evolving Takagi-Sugeno (exTS) algorithm

The Extended Evolving Takagi-Sugeno algorithm (exTS) has been introduced by Angelov & Zhou (2006) and included further improvements to the evolving fuzzy identification method. The data samples were again assessed based on the potential (like in eTS algorithm). The new addition was the introduction of the recursively updated cluster radii, which adapted to the data characteristics of each cluster. Additionally new condition which replaced the contradictory rules and additional quality measures of the clusters (such as support) were introduced.

4.3.3.1 Measures of quality of the generated clusters

In the on-line applications only the cluster centres are kept in the memory, while all other data samples are discarded. The issue arises as to how well those retained data samples represent all of the ones which were discarded. One way is to monitor

certain characteristics of the data which belong to a cluster. In the exTS algorithm this includes the support (population), age and radius of the cluster.

The population of the cluster describes how many data samples are in close proximity to the cluster centre, and is calculated in the same way as in Simpl_eTS algorithm (4.18).

The age of the cluster describes how recent was the last addition of the data sample to a certain cluster. This is calculated in a different way as compared to Simpl_eTS algorithm (4.19). The age in exTS depends on the number of the data samples which belong to that cluster and the average sum of the time indices of that data:

$$age_k(c_i) = k - \frac{2At_k(c_i)}{k+1} \tag{4.21}$$

At represents the accumulated time of arrival:

$$At_k(c_i) = \sum_{l=1}^{N_k(c_i)} I_l \tag{4.22}$$

I is the time stamp at which the data sample was added to the cluster. If the data sample x_k has been added at time instance k, and it's the l^{th} data sample added to the cluster c_i, I_l will equal to:

$$I_l = k,\ l = [1, N_k(c_i)] \tag{4.23}$$

The values of age are interpreted in a following way: the ones close to 0 mean that a recent data sample was included in that cluster, while higher values indicate that no recent data was added to a cluster (the cluster is *old*).

In real world scenarios, the distribution of the data is hard to estimate and is time-varying, making it difficult to capture with a pre-defined value of the cluster radius. In exTS the varying, adaptive cluster radius is introduced which takes into

account the variance of the data (local scatter):

$$r_{k,ij} = \gamma r_{(k-1),ij} + (1-\gamma) S_k(c_{ij}) \tag{4.24}$$

where γ is a weight which determines the importance of the new data as compared to the one already included in the cluster. If $\gamma = 0.5$ the new information and the previous value of the radius are treated equally. The $S_k(c_{ij})$ represents the local scatter of the j^{th} element of the cluster vector and is obtained from Eq. (4.25):

$$S_k(c_{ij}) = \sqrt{\frac{1}{N_k(c_i)} \sum_{l=1}^{N_k(c_i)} (c_{ij} - x_{l,j})^2} \tag{4.25}$$

The scatter approximates the variance of the data samples and is based on the distance of the data belonging to the cluster to its centre c_i. High values of the scatter mean that the data samples are spread around the centre with various distances, whereas small value indicate more even distribution. It is worth noting that concept of scatter has already been introduced in the previous section for Simpl_eTS (4.14). In this case we talk about local scatter which is limited to the cluster centre and not the whole population of the data.

If the new cluster $c_{(R+1)}$ is added, new local scatter $S_k(c_{(R+1)j})$ for each j variable is initialized as an average of local scatters of all existing clusters:

$$S_k(c_{(R+1)j}) = \frac{1}{R} \sum_{i=1}^{R} S_k(c_{ij}) \tag{4.26}$$

The radii of the clusters are different for each of the input's dimension, which makes it possible for them to adapt to the new information brought by the data samples.

4.3.3.2 Membership function calculation in exTS

A new way of calculating cluster radii influences the equation for the activation degree the rule. This is due to the fact that now the radii are different for each of the input values, therefore the activation degree will be different for each of the inputs:

$$\mu_{ij} = e^{\frac{-\|c_{ij} - x_{k,j}\|^2}{2\left(r_{k,ij}\right)^2}} \tag{4.27}$$

4.3.3.3 Recursive cluster assessment in exTS

The potential is calculated and updated in the same way as in original eTS (Eq. 4.1 and 4.3). The algorithm for accepting, replacing and rejecting the data sample in exTS as new cluster is presented below:

Algorithm 11 Decision algorithm for recursive cluster assessment in the exTS algorithm

if $P_k(x_k) < \min_i[P_k(c_i)]$ **OR** $P_k(x_k) > \max_i[P_k(c_i)]$ **then**
  if $\mu_{ij} > 1/3$ **then**
    The closest cluster centre c_i is replaced by the current data sample x_k;
    Population $N_k(c_i)$ increased (4.18);
    Age $age_k(c_i)$ modified (4.21);
    Local scatter $S_k(c_{ij})$ and radius $r_{k,ij}$ updated: (4.25) and (4.24);
  else
    New cluster $c_{(R+1)}$ is added;
    $c_{(R+1)} = x_k$, $P_k\left(c_{(R+1)}\right) = P_k\left(x_k\right)$;
    Population initialized: $N_k(c_{(R+1)}) = 1$;
    Age $age_k(c_{(R+1)})$ calculated (4.21);
    Local scatter $S_k(c_{(R+1)j})$ initialized (4.26);
    Radius $r_{k,(R+1)j}$ calculated (4.24);
  end if
else
  The data sample x_k is assigned to the nearest cluster i, $i : \min_{i=1}^{R}\|x_k - c_i\|^2$;
  Population $N_k(c_i)$ increased (4.18);
  Age $age_k(c_i)$ modified (4.21);
  Local scatter $S_k(c_{ij})$ and radius $r_{k,ij}$ updated: (4.25) and (4.24);
end if

Additional rules are introduced after that step to further refine the cluster structure which will affect future rules.

If the cluster has a very low population $N_k(c_i)$ (regulated by the pre-set threshold ϵ), the rule which is constructed from it can be removed (or rather ignored) by

allowing it's firing degree τ_i to be equal to 0:

$$\textbf{IF } \frac{N_k(c_i)}{k} < \epsilon \textbf{ THEN } \tau_i = 0 \tag{4.28}$$

Additionally, the age of the clusters can be used as a decision value to allow for a replacement of the old rule by other candidate data samples with high potential value.

The parameters of the consequence linear equations are learned through global optimization described in Section 4.3.1.

The overall structure of the exTS algorithm is presented below:

Algorithm 12 exTS algorithm

Read x_k;
Normalize x_k recursively
if $k == 1$ **then** {first data sample, initialize the structure}
    $P_1(x_1) = 1$, $\vartheta_1 = 0$, $\alpha_1 = 0$, $\nu_1 = 0$, $\beta_1 = \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix}$;
    $c_1 = x_1$, $N_1(c_1) = 1$, $age_1(c_1) = 1$, $At_1(c_1) = 1$;
    $r_{1,1j} = 1$, $S_1(c_{1j}) = 1$;
    $R = 1$; {Initialize first fuzzy rule}
    Initialize parameters of consequence equation (same as in eTS);
else {additional data sample, evolve the structure}
    Calculate $P_k(x_k)$ using (4.1);
    Update $P_k(c_i)$ of each centre using (4.3);
    Apply Algorithm 11 to update the cluster structure accordingly;
    Generate rule base from the cluster structure (same as eTS);
    Apply rule (4.28) to reduce the rule base if necessary;
    Estimate the parameters of consequent linear equations using global learning. Same procedure as in eTS (4.4) - (4.9), except (4.27);
end if

4.3.4 Enhanced version of Evolving Takagi-Sugeno (eTS+) algorithm

The Enhanced version of Evolving Takagi-Sugeno (eTS+) algorithm (Angelov et al. 2010) is the most recent extension to the eTS method which includes most of the improvements from simpl_eTS and exTS and adds some additional features:

- Additional quality monitoring features: utility and local density of the clusters.
- On-line input variable selection.

Similarly to other eTS algorithms, the eTS+ can be used with both global and local learning of the consequent parameters. More in depth review of both approaches has been conducted in (Angelov et al. 2010) for eTS+ algorithm.

4.3.4.1 Cluster quality monitoring features in eTS+

The features introduced in this algorithm aim to constantly monitor the quality of the existing cluster structure. This is achieved by looking at the different aspects of the population of the clusters, such as the number of samples associated, how recent was the last addition of the data sample and the importance of the cluster (it's accumulated firing level of the associated fuzzy rule).

The population $N_k(c_i)$ of the cluster c_i is calculated in the same way as in exTS and simpl_eTS (4.18). It gives indication of how much information is carried in the cluster.

Local density calculation provides a way of measuring the spread among the data samples belonging to the cluster (4.29) and gives information about the

distribution of the data:

$$L_k(c_i) = \frac{1}{1 + \sum_{j=1}^{n+1}(S_k(c_{ij}))^2} \tag{4.29}$$

where $S_k(c_{ij})$ is a local scatter of cluster c_i over input at index j during time k calculated using (4.25).

Both population and local density provide information about the generalization power of the clusters.

The age of the cluster represents the accumulated time index of the data samples belonging to that cluster (4.30):

$$age_k(c_i) = k - \frac{\sum_{l=1}^{N_k(c_i)} I_l}{N_k(c_i)} \tag{4.30}$$

where $age_k(c_i)$ is the age of the cluster i at the time instance k, I_l is the time instance of the associated l data sample belonging to the cluster and $N_k(c_i)$ is the population of the cluster. The fuzzy rule structure can be simplified by looking at the values of age for each corresponding cluster. The clusters with high age values (meaning that no new data samples have been in their range of influence for a while) can be removed. By looking at the age dynamics (by measuring the change in the age), it may also be possible to discover drifts in the incoming data (Lughofer & Angelov 2011).

The utility represents a measure of how often the fuzzy rule associated with the cluster was used (4.31):

$$U_k(c_i) = \frac{\sum_{l=1}^{k} \tau_{l,i}}{k - I_i} \tag{4.31}$$

the $U_k(c_i)$ is the utility of the cluster i at the time instance k, $\tau_{l,i}$ is the firing level of the fuzzy rule associated with the cluster i at time l. I_i is the time index at which the cluster i has been created. The utility measures how often the fuzzy rule

has been used. High variability in the utility values indicate that different rules are activated over time with varying firing levels. The extend to which the rule is used over time can give some insight into the non-stationarity of the problem; if the utility value changes often it may indicate that data is non-stationary.

The aforementioned features are used to constantly assess the quality of the clusters during the clustering process. Therefore several conditions have been introduced (Angelov et al. 2010) in the eTS+ algorithm to make use of those features:

$$\begin{aligned}
&\textbf{IF } (U_k(c_i) < \overline{U}_k - \hat{U}_k) \textbf{ THEN } \text{Disable rule } i \\
&\textbf{IF } (age_k(c_i) > \overline{age}_k + a\hat{g}e_t) \textbf{ THEN } \text{Disable rule } i \\
&\textbf{IF } (N_k(c_i) < 3) \textbf{ AND } (t \geq I_i + 10) \textbf{ THEN } \text{Disable rule } i
\end{aligned} \tag{4.32}$$

where $\overline{U}_k$ and $\hat{U}_k$ are mean and standard deviation of the utility, $\overline{age}_k$ and $a\hat{g}e_k$ are mean and standard deviation of the age, $N_k(c_i)$ is the population and I_i is the time index at which the cluster i has been created. The rules are disabled by manually setting their firing levels τ_i to 0.

The cluster radii are updated in the same way as in exTS algorithm (4.24).

The algorithm for recursive cluster assessment for eTS+ is presented below (Algorithm 13).

Algorithm 13 Decision algorithm for recursive cluster assessment in the eTS+ algorithm

if $P_k(x_k) < \min_i[P_k(c_i)]$ **OR** $P_k(x_k) > \max_i[P_k(c_i)]$ **then**
  if $\mu_{ij} > e^{-1}$ **then**
    The closest cluster centre c_i is replaced by the current data sample x_k;
    Population $N_k(c_i)$ is increased (4.18);
    Local scatter $S_k(c_{ij})$ updated (4.25);
    Local density $L_k(c_i)$ modified (4.29);
    Radius $r_{k,ij}$ updated (4.24);
  else
    New cluster $c_{(R+1)}$ is added;
    $c_{(R+1)} = x_k$, $P_k\left(c_{(R+1)}\right) = P_k\left(x_k\right)$;
    Population initialized: $N_k(c_{(R+1)}) = 1$;
    Local scatter $S_k(c_{(R+1)j})$ initialized (4.26);
    Local density $L_k(c_i)$ modified (4.29);
    Radius $r_{k,(R+1)j}$ calculated (4.24);
  end if
else
  The data sample x_k is assigned to the nearest cluster i, $i : \min_{i=1}^{R}\|x_k - c_i\|^2$;
  Population $N_k(c_i)$ is increased (4.18);
  Local scatter $S_k(c_{ij})$ and radius $r_{k,ij}$ updated: (4.25) and (4.24);
  Local density $L_k(c_i)$ modified (4.29);
end if
Age $age_k(c_i)$ modified (4.30) for all clusters;
Utility $U_k(c_i)$ calculated (4.31) for all clusters;

4.3.4.2 Local and global learning in eTS+

The differences between global and local learning have already been explained in this chapter. Both methods vary in terms of computational time and give slightly different results. In general, local learning may produce more locally interpretable rule structure, whereas global learning, in theory, should give results with smaller errors due to the compensation effect from assessing the rules globally. Global learning is more demanding in terms of computational power due to calculations over one global covariance matrix and the use of global parameter vector. In local learning the calculation is done for each rule separately, which means that both parameter vector and covariance matrix are smaller and easier to process. The reset process, described earlier in the chapter, is more complex for global learning, with the need to expand the covariance matrix. The comparison of both learning methods in Angelov et al. (2010) showed that for several problems the local learning achieved lower RMSE error. The computational time was also significantly faster. The complexity introduced by the rule reduction in the eTS+ is reduced when local learning is used, due to the local assessment of the rules. Although the number of rules was still the same, the local variant was a preferable choice due to the lower error values, higher speed of computation and simpler implementation. This will be analysed and compared as well in the following chapters when the implementation and application of the modified algorithm will be shown.

4.3.4.3 On-line input selection

Very often the number of inputs is known in advance, due to existing knowledge of the factors influencing the outputs or through data pre-processing steps. This steps, in classification literature, are referred to as feature selection (Liu & Motoda 1998) and can include Principal Component Analysis (PCA), Genetic Algorithms (GA) or sensitivity analysis. For dynamic or non-stationary systems it may be

difficult to obtain a best subset of input variables for all conditions, therefore a method for on-line input selection has been suggested by Angelov et al. (2010).

The idea is based on the analysis of the value of input parameters. If a parameter of the input j is consistently small for a range of observed outputs across all of the rules, that input is not contributing significantly to the value of the output and can be seen as the candidate for removal. This can be represented as the accumulated sum of the parameters for the particular input in respect to all existing inputs:

$$\omega_j = \frac{\sum_{i=1}^{R} \theta_{ij}}{\sum_{j=1}^{h} \sum_{i=1}^{R} \theta_{ij}} \tag{4.33}$$

where θ_{ij} is a parameter of j^{th} input of i^{th} rule and ω_j represents the ratio of contribution of the input j.

Based on the value of that ratio, a condition can be implemented which compares the value of ω_j to the contribution of all parameters $\sum_{j=1}^{h} \sum_{i=1}^{R} \theta_{ij}$ or to the contribution of the most influential parameter of input j with highest $\sum_{i=1}^{R} \theta_{ij}$, and as a consequence remove that input.

It is worth noting that the removal of an input has an impact on the vector of data samples x, vector of cluster centres c and it affects the size of covariance matrix and parameter vector. This leads to the increase in complexity of the algorithm, as those changes need to be addressed by means of, for example, a reset of covariance matrix.

The feature of automatic input selection is very important in the domain of intelligent sensors (Angelov, Kordon & Zhou 2008). As the application to sensors is not in the scope of this research, this feature will not be discussed in more details in this thesis, however more information can be found in the papers by Angelov et al. (2010).

The steps of the eTS+ algorithm are described below:

Algorithm 14 eTS+ algorithm

Read x_k;
Normalize x_k recursively
if $k == 1$ **then** {first data sample, initialize the structure}
    $P_1(x_1) = 1$, $\vartheta_1 = 0$, $\alpha_1 = 0$, $\nu_1 = 0$, $\beta_1 = \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix}$;
    $c_1 = x_1$, $N_1(c_1) = 1$, $age_1(c_1) = 1$, $U_1(c_1) = 1$, $L_1(c_1) = 1$, $\omega_1 = 1$;
    $r_{1,1j} = 1$, $S_1(c_{1j}) = 1$;
    $R = 1$; {Initialize first fuzzy rule}
    Initialize parameters of consequence equation (same as in eTS);
else {additional data sample, evolve the structure}
    Calculate $P_k(x_k)$ using (4.1);
    Update $P_k(c_i)$ of each centre using (4.3);
    Apply Algorithm 13 to update the cluster structure accordingly;
    Generate rule base from the cluster structure (same as eTS);
    Apply rules from (4.32) to reduce the rule base if necessary;
    Calculate the activation degree for each input (4.27);
    if Global learning **then**
        Reset the global covariance matrix if necessary (4.5);
        Estimate the parameters of consequent linear equations (4.9);
    else if Local learning **then**
        Initialize new covariance matrix if necessary (4.4);
        Estimate the parameters of consequent linear equations (4.10);
    end if
    Optionally use on-line input selection (4.33);
end if

4.3.5 Comparison of different Evolving Takagi Sugeno approaches

Although the principles of all of the eTS algorithms are the same, a number of improvements have been introduced which improve the quality of the fuzzy models. Those improvements focused around the adaptivity of the cluster structure by introducing ways of recursive monitoring of the components of the clusters. These included features such as adaptive radius, age or population of the clusters among the few. The introduced features and the development of the algorithms can be seen in Table 4.2:

Table 4.2: Comparison of different features of eTS algorithms.

Feature	eTS	Simpl_eTS	exTS	eTS+
radius	r constant	r constant	r adaptive (4.24)	r adaptive (4.24)
potential	P_k - Gaussian potential (4.1)	S_k - Cauchy scatter (4.15)	P_k - Gaussian potential (4.1)	P_k - Gaussian potential (4.1)
cluster population	-	$N_k(c_i)$ (4.18)	$N_k(c_i)$ (4.18)	$N_k(c_i)$ (4.18)
cluster age	-	$age_k(c_i)$ (4.19)	$age_k(c_i)$ (4.21)	$age_k(c_i)$ (4.30)
cluster density	-	-	-	$L_k(c_i)$ (4.29)
cluster utility	-	-	-	$U_k(c_i)$ (4.31)
rule weighting	-	1% samples (4.20)	threshold ϵ (4.28)	conditions (4.32)
input selection	-	-	-	on-line (4.33)

It becomes obvious that introduction of additional features will have an impact on the speed of execution of the algorithms. On the other hand, some of the improvements aim at decreasing the computational burden by disabling some of the less influential rules. The on-line input selection process can improve the performance as well, if the initial input choice is too wide. Those approaches have been applied to a number of physical systems with the aim to predict their outputs and their performance has been compared, both in terms of accuracy and efficiency. These will be briefly described at the end of this chapter together with other on-line fuzzy clustering approaches such as FLEXFIS and DENFIS.

4.4 FLEXFIS algorithm

4.4.1 Introduction

The FLEXFIS (FLEXible Fuzzy Inference System) family of algorithms has been developed by Lughofer (2008). It consists of FLEXFIS algorithm for regression and FLEXFIS-Class algorithm for classification. The principles of the regression algorithm are the same as for eTS described in the previous section (see Fig. 4.1). The main difference lays in the way the clusters are generated and updated and how this relates to the learning part of the consequence parameters of the obtained rules. This section will briefly describe the way the algorithm is implemented, outline some of the modifications which were introduced afterwards and compare it to the eTS family of algorithms. Some new parameters will be introduced, for which the description can be found in the notation section at the beginning of this chapter.

4.4.2 Algorithm description

The algorithm has one parameter ρ, which needs to be set in advance. It is used to assess if the normalized data sample $\tilde{x}_k$ is close to the normalized cluster centre $\tilde{c}_i$, and as a result, if the new cluster centre should be created or if the existing centre should be updated.

Depending on the problem, the algorithm can be applied to a fully or partially trained Takagi-Sugeno model or can start from the first sample. If previous model is used, the parameter ρ can be set to the optimal value through parameter grid search, and the ranges (min and max values) of the variables can be estimated from the available data. If the algorithm is evolving from scratch, ρ can be set according to the past experience in the range of $[0.2\ 0.3]\frac{\sqrt{n+1}}{\sqrt{2}}$ (n being the number

of variables) (Lughofer 2011*a*).

4.4.2.1 Initialization of the algorithm

If the algorithm starts evolving from the first data sample x_1, the cluster structure is initiated with that sample ($c_1 = x_1$, $k_{c_1} = 1$, $R = 1$) and the width of the cluster centre is initiated with 0, $\sigma_1 = 0$.
The first fuzzy rule $R = 1$ is also created from the new cluster:

$$c_{Rj} \text{ (set)} = c_{Rj} \text{ (cluster)} \tag{4.34}$$

$$\sigma_{Rj} \text{ (set)} = \epsilon \text{ range}(x_j) \tag{4.35}$$

where c_{Rj}(set) is the focal point of j^{th} fuzzy set of rule R. σ_{Rj} is the width of the j^{th} fuzzy set and is initiated with the range $\max_k[x_{kj}] - \min_k[x_{kj}]$. A parameter ϵ is used to initiate the range with a small number to avoid instabilities (Lughofer 2008).
The consequence vector of parameters θ_1 is initiated with zeros, and local covariance matrix cov_1 with the constant identity matrix QI, Q being a large number.
Last step is to apply the local learning of parameters, same as in eTS family of algorithms (4.10), and update the ranges of the variables accordingly.

4.4.2.2 Cluster assessment

When reading the next data sample or when algorithm starts from the model which has already been developed, the data needs to be normalized. For new data sample x_k, this is done the same way as for the Subtractive clustering algorithm (3.17). For the identified cluster centres c_i, and the variance (widths) of Gaussian membership functions σ_i of the cluster centres, the normalization is done in the

similar fashion:

$$\begin{aligned} \tilde{c}_i &= \frac{c_i - \min\limits_k[x_{k-1}]}{\max\limits_k[x_{k-1}] - \min\limits_k[x_{k-1}]} \\ \tilde{\sigma}_i &= \frac{\sigma_i}{\max\limits_k[x_{k-1}] - \min\limits_k[x_{k-1}]} \end{aligned} \tag{4.36}$$

The distance from the data sample $\tilde{x}_k$ to all the cluster centres $\tilde{c}_i$ is then calculated using predefined measure (for example Euclidean distance) and the cluster centre closest to the data sample is marked as $\tilde{c}_{win}$.

4.4.2.3 Adding new cluster

If the shortest distance is higher or equal the parameter ρ then new cluster is created. Adding new cluster increases the number of fuzzy rules: $R = R + 1$. The parameter k_c is set $k_c = 1$. New cluster is created at the location of the current data sample $\tilde{c}_R = \tilde{x}_k$. The normalized variance $\tilde{\sigma}_R$ of the new cluster is set to $\tilde{\sigma}_R = 0$ in all dimensions. After that, the centres and variances of all clusters are transformed back to the original values:

$$c_i = \tilde{c}_i \left(\max_k[x_{k-1}] - \min_k[x_{k-1}] \right) + \max_k[x_{k-1}] \tag{4.37}$$

$$\sigma_i = \tilde{\sigma}_i \left(\max_k[x_{k-1}] - \min_k[x_{k-1}] \right) \tag{4.38}$$

The centres, widths of the new fuzzy sets, the elements of the consequence parameter vector θ_R and the covariance matrix cov_R of the new rule R are initiated in the same way as for the first fuzzy rule described in **Initialization of the algorithm** section.

4.4.2.4 Updating winning cluster

If the shortest distance is smaller than the parameter ρ then the winning cluster $\tilde{c}_{win}$ is updated by shifting its position towards the data sample $\tilde{x}_k$ using the learning gain η_{win}:

$$\tilde{c}_{win} = \tilde{c}_{win} + \eta_{win}(\tilde{x}_k - \tilde{c}_{win}) \tag{4.39}$$

$$\eta_{win} = \frac{\eta_{init}}{k_{c_{win}}} \tag{4.40}$$

where η_{init} is initial learning gain and $k_{c_{win}}$ is a number of samples belonging to the winning cluster.

The components of the variance $\tilde{\sigma}_{win}$ of the winning cluster need to be updated as well:

$$\tilde{\sigma}_{ij}^2 = \frac{k_{c_i}\tilde{\sigma}_{ij}^2 + (k_{c_i}+1)\Delta\tilde{c}_{ij}^2 + (\tilde{c}_{ij} - \tilde{x}_{ij})^2}{k_{c_i}+1} \tag{4.41}$$

where i is the cluster index, j is the data dimension index and $\Delta\tilde{c}_{ij}^2$ is a distance between the old and new position of the cluster. After those steps, the centres and variances (widths) of all clusters are transferred back to the original range as in (4.38). The new cluster centre and the widths are used to update the antecedent part of the corresponding fuzzy rule. The consequent parameters and the covariance matrix remain the same as before the update.

4.4.2.5 Update of consequent parameters

After the new data sample has been assessed, the parameters of the consequence linear equations of all the fuzzy rules are updated using RLS algorithm (Local Learning from Equation 4.10). The $\min_k[x_{kj}]$ and $\max_k[x_{kj}]$ values of the observed data are updated accordingly, and if new data is still available, the algorithm starts from **Cluster assessment** section.

The steps of the FLEXFIS algorithm are described below:

Algorithm 15 FLEXFIS algorithm (assume evolution from scratch)

$k == 1$, $\rho = [0.2\ 0.3]\frac{\sqrt{n+1}}{\sqrt{2}}$;
while Data sample x_k available **do**
  if $k == 1$ **then** {first data sample, initialize the structure}
    $c_1 = x_1$, $k_{c_1} = 1$, $\sigma_1 = 0$;
    Create new fuzzy rule $R = 1$ (4.35);
    Initiate θ_1 with zeros, set $cov_1 = QI$;
  else {new data sample, evolve the structure}
    Normalize x_k using (3.17), cluster centres and their widths using (4.36);
    Calculate the distance from $\tilde{x}_k$ to all of the cluster centres $\tilde{c}_i$;
    Mark closest cluster centre as $\tilde{c}_{win}$;
    if $\|\tilde{x}_k - \tilde{c}_{win}\| \geq \rho$ **then** {create new cluster}
      $R = R + 1$, $\tilde{c}_R = \tilde{x}_k$, $k_{c_R} = 1$, $\tilde{\sigma}_R = 0$;
      Transform centres and variances of all clusters back to the original values (4.38);
      Create new fuzzy rule R (4.35);
      Initiate θ_R with zeros, set $cov_R = QI$;
    else {update exisitng cluster}
      Update winning cluster $\tilde{c}_{win}$ using (4.40);
      Update the components of the variance $\tilde{\sigma}_{win}$ using (4.41);
      Transform all $\tilde{c}_k$ and $\tilde{\sigma}_k$ back to the original values (4.38);
      Update the antecedent part of the fuzzy rule with c_{win} and σ_{win};
      θ_{win} and cov_{win} remain unchanged;
    end if
  end if
  Estimate the parameters using the Local learning algorithm from (4.10);
  Update the $\min_k[x_{kj}]$ and $\max_k[x_{kj}]$;
end while

4.4.3 Extension of FLEXFIS and modifications to the algorithm

There are number of modifications to the existing FLEXFIS algorithm. In Lughofer et al. (2011) the focus was put to reduce the complexity of the system by decreasing the number of rules during training (evolution) of the cluster/rule structure. This was achieved by adding a step which calculated the similarity

degree between the rule which has just been updated and the existing rule structure. The similarity was used to merge rules which resulted in simplified cluster structure, without sacrificing the accuracy of the prediction. The drawback of this additional step was that it increased the computational time of the algorithm, which has an impact when fast, real time updates are required.

Lughofer & Angelov (2011) focused on the problem of handling drifts and shifts in the data space. An example of data drift could be when the incoming data starts pointing to a consistently different, new output value or when there is a change of range of the observed inputs. The drifts were detected by analysing the change of the *age* (4.30) of the clusters (the gradient of the *ageing* curve and the second derivative of *age*). When detected, the data drift affects the way the antecedent and consequent parts of the fuzzy rules are calculated. In antecedent part the learning gain η (4.40) calculation is modified by resetting the number of samples belonging to the cluster by using the forgetting factor λ. In the consequent part, the handling of data drifts is included by adding the forgetting factor λ to the local learning (4.10). The modified method was used to improve a prediction model of a steel rolling mill and also applied to a process of detecting errors in CD imprints. In both cases, when data shifts occurred, the approach improved the accuracy of the prediction. It is important to note, that the choice of the λ has a big influence on the results as choosing the value to be too high results in not enough flexibility (slow forgetting), while having the value too low may destabilize the model and create local optima (danger of over-fitting).

Finally, Lughofer (2011*b*) investigated the possibility of including recursive feature weighting in the classification algorithms based on the FLEXFIS approach. Features were weighted according to their importance / current usability in order to decrease the over-fitting of the model. This was achieved by integrating an incremental feature weighting algorithm (using Dy-Brodley's separability criterion) into the FLEXFIS Classifier. Two approaches were considered, one based on

leave-one-feature-out approach and another on feature-wise approach (Lughofer 2011*b*). Although inclusion of the feature weighting algorithm had an impact on the speed of the computation of the algorithm, it proved effective as it was able to eliminate redundant features and increased overall accuracy of the classification.

4.4.4 Comparison with eTS

The principles of FLEXFIS and eTS are generally the same, and similar modifications can be applied to both algorithms. However there are some important differences to consider, particularly the concept of radius in eTS (FLEXFIS has a radius included in widths of the fuzzy rules). Few more important difference from my perspective will be briefly described here.

Both approaches explore the RLS algorithm and its variation in case of local learning. Both algorithms can also recursively evolve the clusters (fuzzy rules) as new data becomes available. The main difference lays in learning from scratch, i.e. when no data is available to pre-train the model. In eTS it's possible to do so due to the way the potential of clusters can be updated and that no normalisation of data samples is necessary (although beneficial) as high potential values represent high data-dense regions and are not dependent on the scale of data. In FLEXFIS some data is required to establish the ranges for each feature (input). If some knowledge of data is present, this can be avoided by presetting the ranges based on that knowledge.

Another difference lays in the choice and role of the parameters. In eTS the initial value of radius r needs to be pre-set as it controls the area of influence of the cluster and decides when to replace (evolve) the cluster centre with new data. In FLEXFIS the parameter ρ controls the way the algorithm adapts cluster centre to new data. The parameter is selected through parameter grid search on the available data. Due to the way those two parameters are implemented, there is a

fundamental difference on how the cluster centres are allowed to change in those two algorithms. In eTS the cluster centre can shift to another data rich region, but the position of the centre is limited to the considered data samples. In FLEXFIS the centres can evolve more smoothly by slowly shifting to the more promising region.

As in FLEXFIS the clusters evolve whenever a new data sample lays further than a pre-defined parameter ρ, the algorithm will generate new cluster even though the considered data sample may be an outlier. In eTS, the potential of the region around that data sample has to be high (more data samples need to exist in proximity defined by radius r) for the cluster to be created, making it more robust to outliers. Additional modifications to the FLEXFIS algorithm are required in order to improve it's robustness.

For more detailed comparison, please refer to (Angelov & Lughofer 2008) which in detail discusses the difference between those two approaches.

4.5 Dynamic evolving neuro-fuzzy inference system (DENFIS)

4.5.1 Introduction

Dynamic evolving neuro-fuzzy inference system (DENFIS) was introduced by Kasabov & Song (2002). The principles of this method are very similar to the approach used in eTS. Takagi-Sugeno models are identified on-line by partitioning data space using recursive clustering algorithm called Evolving Clustering Method (ECM). The fuzzy rules are generated and updated constantly from the clusters and the output is calculated on-line based on the number of most activated fuzzy

functions.

The main differences lay in the way the clusters are generated (using ECM), the use of triangular membership functions instead of Gaussian and the use of forgetting factor in the estimation of parameters of the linear output equations.

4.5.2 Evolving Clustering Method

ECM can be used both in real time applications and when all of the data is available. In the off-line mode the ECM is applied to the part of the dataset to establish initial cluster structure. The clusters are then further optimised with remaining data using the objective function (Kasabov & Song 2002).

The main differences between ECM and the eTS clustering lay in the way the cluster radii are calculated and updated as well as the way the cluster centres are moved around the data space. More detailed description is presented below in the Algorithm 16:

Algorithm 16 ECM clustering in DENFIS

```
1: k == 1, set max radius value D_thr;
2: while data sample x_k available do
3:     if k == 1 then        {first data sample, initialize the cluster structure}
4:         c_1 = x_1, r_1 = 0;
5:     else        {new data sample, evolve the structure}
6:         Calculate the distance from x_k to all of the cluster centres c_i ||x_k − c_i||;
7:         Mark closest cluster centre as c_min;
8:         if ||x_k − c_min|| ≤ r_win then
9:             Sample x_k belongs to c_min;
10:        else
11:            Calculate dist_min = min (||x_k − c_i|| + r_i);
12:            if dist_min > 2D_thr then        {create new cluster}
13:                c_k = x_k, r_k = 0;
14:            else if dist_min ≤ 2D_thr then        {cluster is updated}
15:                r_min = dist_min/2;
16:                Updated cluster centre c_new is positioned on the line between the old centre
                   c_min and the data sample x_k;
17:                Distance to the data sample x_k equals to the new radius r_win;
18:            end if
19:        end if
20:    end if
21: end while
```

4.5.3 DENFIS inference system

In DENFIS first order Takagi-Sugeno models are used, with linear functions in the consequence part. The parameters of those functions are established using weighted RLS algorithm with a forgetting factor. The forgetting factor is used in order to speed up the adjustment of the estimated parameters to changing inputs.

Similarly to other algorithms, new rules are created and updated together with the cluster updates. They inherit the initial parameter values from the fuzzy rule which are reflected by the closest cluster. The membership functions of the fuzzy rules in DENFIS are triangular and are described by 3 parameters directly related

to the properties of the cluster:

$$\mu(x_j) = mf(x_j, a, b, c) = \begin{cases} 0, & x \leq a \\ \frac{x_j - a}{b - a}, & a \leq x \leq b \\ \frac{c - x_j}{c - b}, & b \leq x \leq c \\ 0. & c \leq x \end{cases} \tag{4.42}$$

where b is the i^{th} cluster centre c_{ij} on the j^{th} input dimension, $a = b - d \cdot D_{thr}$, $c = b + d \cdot D_{thr}$, constant $d = 1.2 - 2$, D_{thr} is a clustering threshold.

Rules are updated if the distance of the reflecting clusters to the data sample is higher than 2 times the threshold D_{thr}. If the position of the centre changes, the rules are updated accordingly with the triangular membership functions. The steps of the DENFIS algorithm can be seen below:

Algorithm 17 DENFIS algorithm

1: $k == 1$, set max radius value D_{thr};
2: **while** data sample x_k available **do**
3: Read data sample x_k;
4: Apply ECM clustering algorithm (Alg. 16);
5: Generate/update antecedent parts of fuzzy rules using the triangular membership functions (4.42);
6: Estimate the parameters of the consequent linear equations using wRLS with forgetting factor λ;
7: **end while**

4.6 Applications of Fuzzy Evolving methods

The original eTS algorithm has been tested on a variety of data sets. It has also been frequently used as a baseline of performance when the results from more advanced or modified algorithms have been presented. In Angelov (2004*b*) the lactose concentration was modelled on-line with eTS by using data for the cell mass concentration and the dissolved oxygen concentration. Satisfactory RMSE values

were reported, but the results have not been compared with any other approach in this paper. In Angelov & Filev (2004) more thorough testing of eTS has been carried out with the modelling of the fan-coil sub-system of an air conditioning unit and an artificial time-series generated by the Mackey-Glass (MG) process. The results obtained from the MG time-series have been compared with DENFIS and a number of off-line algorithms such as Artificial Neural Networks. The accuracy was measured using the NDEI (Non-Dimensional Error Index) and resulted in small number of rules. A number of batch algorithms have had better performance in terms of accuracy, but generated significantly more rules or nodes. Andreu & Angelov (2010) applied the eTS to a NN GC1 time-series which was provided for the forecasting competition. The error measures have not been shared, nor they have been compared with any other forecasting approach.

In Angelov & Filev (2005) the authors applied both eTS and Simpl_eTS algorithms to Box-Jenkins data set. It is a common benchmarking problem of modelling the CO_2 content in the output of the gas furnace. The simplification from using the Simpl_eTS resulted in a smaller number of clusters and faster computational time. At the same time similar error values were achieved for Root Mean Squared Error (RMSE) and NDEI.

The exTS algorithm has been tested on the artificially generated Mackey-Glass time-series dataset and the real data from the car engine where the NO_x emissions were predicted based on the several existing inputs. The results showed a slight improvement in terms of the accuracy over the eTS method but higher number of rules. The authors claimed that although more clusters were generated, the resulting rule structure is more easily interpretable linguistically; however, this has not been shown in the paper and can only be assumed.

The most comprehensive comparison of the fuzzy evolving approaches has been carried out in Angelov et al. (2010). The eTS+ has been compared with other

Fuzzy Evolving algorithms such as eTS and exTS, FLEXFIS, DENFIS and Feed-forward Neural Networks. The data-set used for comparison consisted of data-sensor readings from a car engine test bed (Angelov 2011), which aim was to automatically evaluate the content of the NOx emissions from the exhaust and from artificially generated time series using Mackey-Glass function. The results (Angelov et al. 2010) showed slightly lower RMSE error values and lower computational time for the eTS+ algorithm. The algorithm used utility-based rule simplification (4.31) and automatic input selection (it was able to reduce the number of inputs from 180 to 4, and number of rules from 40 to 4). However, the conditions such as the size of the training period, number of testing data samples or the period for which the predictions was carried out have not been described in the paper and require further clarification. The test on the Mackey-Glass generated time-series provide more information on the testing environment. The results showed that, in this case, exTS algorithm outperformed all others. eTS+ generated smaller number of rules and computed faster under the circumstances. The differences in obtained results suggest that eTS+ perform better for more complicated cases, however having only two tests one can't get to meaningful conclusions. Those methods will be compared with the modified version of the algorithm in Chapter 7 on a bigger and different dataset of leakage data.

Other applications explored the use of adaptive classifiers (Angelov, Lughofer & Zhou 2008) or adaptive sensors (Angelov & Kordon 2010). Some less related work, but still applying the principles of evolving approach, used Expectation Maximization algorithm, both off-line and the on-line, in time series approach for stream-flow forecasting (Luna et al. 2007). The modification of that approach added soil moisture accounting procedure and used subtractive clustering for initialization and recursive version of Expectation Maximization (Luna et al. 2009). Additional work exploring evolving fuzzy systems for daily forecasting was considered by Luna & Ballini (2011).

4.7 Conclusions

The fuzzy evolving methods presented in this chapter show promising results when applied to problems with non-linear relationships (artificial MG time-series). However, they have been commonly tested on the same data, such as modelling and prediction of the physical systems (estimation of gas content of a chemical process or emissions from a vehicle). Usually the relations between the parameters of those processes have been well understood. They can be easily compared, but more thorough analysis is needed. There seems to be a lack of evaluation of those methods on forecasting problems, when more complex, or unknown, relationships exist between the variables. The choice of the parameters of the algorithms are also not well understood as they have been based on the evaluation on the same group of problems.

In the next chapter a problem of leakage forecasting will be presented. It's a complex issue which doesn't have well defined relationships. In the following chapters a more top-down approach will be shown, which led to additional modification to the eTS algorithm. The results of the modified algorithm will be compared with other fuzzy evolving approaches, but also with other established forecasting methods to give a better overview on the performance.

Chapter 5

Use case - forecasting leakage in the water industry

5.1 Introduction

The aim of this chapter is to outline the problem of water leakage forecasting and the factors which influence it. The current method of leakage forecasting is described and some issues and open questions are presented along with possible approach to solve those. The chapter has the following structure: Section 5.2 describes the scope of the water industry sector in the UK, focusing on Severn Trent Water, the industrial collaborator for this project. In Section 5.3 the importance of water demand prediction is outlined. The issue of leakage in water systems, the social aspect and its impact on the economy and different types of detection methods of leaks will be presented in Section 5.4. Section 5.5 is dedicated to pointing out the factors, methods and issues with leakage forecasting. It also describes the current leakage forecasting approach and outlines the existing issues and possible approach to solving those. Section 5.6 concludes the chapter with

closing remarks.

5.2 Water Industry - setting the scene

As demand for water increases every year together with an increase in population and a rise in demand for agriculture and industry (Babovic et al. 2002), water distribution companies face the problem of maintaining and constantly improving their water networks. The issue of frequent structural failures, water leakages and interruption in services is especially visible in countries such as the United Kingdom, which has many pipelines dating back to 1940s (Savic et al. 1997). These problems can be addressed by continuous investment in the infrastructure and appropriate leakage control policy. However, to be able to assign the resources required to perform those tasks, the company needs to have the estimated leakage values, which can be delivered through a leakage forecasting process.

This research has been conducted in cooperation with Severn Trent Water, one of the biggest companies responsible for water management and supply, and waste water treatment and disposal, in the UK. The company was established in 1973, when numerous small local water companies were merged to create regional authorities, which were responsible for water supply, sewage treatment and river protection within their areas (Severn Trent Water 2013). The local authorities were privatized in 1989 and regulatory body called OFWAT (Office of Water Services) was established. The changes meant that water companies have to provide not only a certain level of service to their customers, but also satisfy the requirements of the company's board and shareholders. Leakage and demand forecasting procedures became a crucial part of resource planning and decision making for the water industry.

To comply with regulations issued by OFWAT, companies are required to provide

the estimates of the expected leakage and, as a result, the ways to maintain it at the required annual target level. The task is not straightforward, as leakage is not constant throughout the year and depends on the leakage control policy applied, state of the infrastructure, weather, seasonal events and fluctuations in demand (for example cultural events, such as music festivals).

5.3 Water demand and supply

Demand forecasts are necessary to keep the associated costs of collecting and keeping the water in the reservoirs to the minimum (Fildes et al. 1997). They are also very important for environmental planning and sustainable utilization of water resources (Nasseri et al. 2011).

Water demand is influenced by domestic, commercial and industrial use as well as agriculture and horticulture. Out of these factors, the domestic use accounts for over 60% of all water supplied to the network (Market Transformation Programme 2008). Therefore, understanding domestic water use is vital for efficient water management and planning.

To properly measure household water demand, water consumption is often split into a number of micro-components or types of water use. These could be the use of bath or shower, toilet, internal tap, various appliances, such as washing machine, and water gardening. Three aspects of those micro-components are usually considered for forecasting (Marshallsay 2005): ownership (who is using the water), volume and frequency of use. Some of them can be measured (using water meters) and others can be assessed by various studies (for example demographic or economic) by carrying on surveys among customers.

As mentioned in Section 2.6 of Chapter 2, the problem of water demand prediction is complex due to existing non-linearities. Although water demand is a crucial part

of the strategy of every water supplying company, the scope of this Ph.D. research was focused on leakage forecasting, which is presented in more details in the next section.

5.4 Water leakage

Leakage accounts for the majority of water losses in the water network. Not only does it have an impact on consumers, due to service interruptions, but it also presents an economic challenge to water supplying companies. Lost water cannot be billed, and finding and fixing the leaks is a time and resource consuming process. Companies employ different strategies in order to face this challenge, from dividing the network into controllable areas, where the water balance can be measured, to applying Active Leakage Control (ALC) policies. Leakage forecasting plays a crucial role in this process, as those activities require careful planning and managing of available resources.

5.4.1 Social impact of leakage

Water scarcity has been recognized as the second most important risk the world will face in the years ahead at the World Economic Forum in Davos in 2013. It is affecting people around the globe as a result of the alteration of rain patterns and the influence of climate change. Supply problems in countries such as the UK are usually not as dramatic as in other parts of the world. Every now and then, however, extreme weather events, such as the drought in southern Britain in 2006, have an impact on society and industries with high water demand profile, such as agriculture. As little can be done at the moment, at least in the short term, to overcome the problems caused by the weather, careful leakage management can be one of the approaches to decrease the impacts of those events on the water

supply. This is confirmed by the study from Water UK (Hoyle 2010). Since 1995, due to ongoing investments in the water network and improvements to the leakage control policies, water leakage in England and Wales has decreased by 30%.

5.4.2 Economic impact of leakage

Although losses of water should be considered unacceptable from a social point of view, leakage control and related activities can be very expensive. Water companies have to work with limited budgets and will try to achieve a balance between the costs related to controlling the leakage, and the financial benefits this will accrue (Pearson & Trow 2005). Therefore, they often look to maintain what is called the Economic Level of Leakage or ELL.

Figure 5.1: Economic Leakage Level.

The key aspect of ELL is the understanding of the relationships between the total costs to be incurred to keep the leakage at a given level and the leakage itself. Fig. 5.1 shows the relationship between the value of total costs and the water leakage. In general, as the leakage increases, the cost of the lost water increase as well. At the same time however, the costs related to the detection and repair effort are lower. This is due to the fact, that it is easier and cheaper to find and fix large, visible pipe bursts which usually are a cause of high leakage within the system. The ELL is found as a trade-off between those two factors. The losses associated

with water networks may be divided into two groups (Morais & de Almeida 2007):

- Apparent losses (background leakage): these are losses obtained from utility operation (company own water usage), meter inaccuracies, unexpected increases in water demand which are recognized as leakage and data errors. These losses cannot be measured with current technology. Therefore, the extent to which the background leakage can be discovered increases with the introduction of new, improved measurement techniques.
- Real losses: connected with physical losses of water from the distribution network through leakage, storage overflows, unaccounted usage of water and unauthorized and illegal use. These can be controlled by frequent checks and proper maintenance.

Table 5.1: Water Balance Table

<table>
<tr><td rowspan="9">System Input Volume</td><td rowspan="4">Authorized Consumption</td><td rowspan="2">Billed Authorized Consumption</td><td>Billed Metered Consumption</td><td rowspan="2">Revenue Water</td></tr>
<tr><td>Billed Unmetered Consumption</td></tr>
<tr><td rowspan="2">Unbilled Authorized Consumption</td><td>Unbilled Metered Consumption</td><td rowspan="7">Non Revenue Water</td></tr>
<tr><td>Unbilled Unmetered Consumption</td></tr>
<tr><td rowspan="5">Water Losses</td><td rowspan="2">Apparent Losses</td><td>Unauthorized Consumption</td></tr>
<tr><td>Customer Meter Inaccuracies</td></tr>
<tr><td rowspan="3">Real Losses</td><td>Leakage on Transmission & Distribution Mains</td></tr>
<tr><td>Leakage and Overflows at Reservoirs</td></tr>
<tr><td>Leakage on Service Connections up to metering point</td></tr>
</table>

The breakdown of the water balance in the system results in the Revenue and Non Revenue Water (NRW) (Lalonde 2004). This concept and more detailed overview of types of losses incurred in the water network are presented in Table 5.1. NRW is

a measure which accumulates all the water which has been produced but does not reach the customer or does not bring any profit. This not only includes leakage, but also apparent losses and unbilled authorized consumption.

The assessment of ELL has a long history in the United Kingdom, with a major national research programme that finished in 1994, which led to a greater understanding of the relationships between different factors, like pressure and leakage. However, the situation in many other countries around the world is different. Many small water companies exist, which cover limited areas. Although a lot of properties may be metered, only limited hydraulic data is available, and very few companies have their networks modelled (Pearson & Trow 2005).

5.4.3 Leakage detection methods

Based on the conversations with the experts from STW, in general, the work-flow for leakage detection and repair can be summarized by the following sequence of actions:

1. Consider agreed leakage targets and demand forecasts and assign staff and resources accordingly between detection and repair groups.
2. Target possible leakage locations and find the leaks.
3. Leaks can also be reported by customers when they become visible.
4. Schedule and plan the work.
5. Carry out the repairs.
6. Report the results.

The extent to which leakage exists in the system depends very much on external factors, such as weather. The impact of those factors can be minimized by applying

the Active Leakage Control (ALC) measures. This is defined as an active effort to locate and repair unreported main leaks (Lalonde 2004). The ALC measures consist of:

- Sonic & correlation surveys
- Noise logging surveys
- Maintaining temporary or permanent District Metered Areas (DMA)
- Step testing

Sonic & correlation surveys and noise logging take advantage of the acoustic noise that water generates when leaking under pressure. Sonic (or listening) surveys employ the use of experienced engineers, who manually work their way listening at the fittings for the leak noise. That technique can be applied to metallic pipes but it cannot be used efficiently on plastic or large diameter pipes due to noise attenuation. Correlation surveys overcome those drawbacks, by measuring pipe noise from two locations that surround the suspected leak. A simple relation is then used, which involves time delay, distance and propagation velocity of the sound, to locate the leak. These methods, although considered essential in order to prevent the leaks from getting bigger, are very labour intensive and may be inefficient, especially for large distribution systems.

To overcome the drawbacks of scalability, district metering is used, which allows for a higher level of control over leakage. A district is defined as an area of distribution network, which is limited by valves and for which the amount of water entering and leaving can be measured. The isolation of different regions of water system, allows for analysis of flow and pressure in those areas and provides means for leakage experts to calculate the levels of leaks in the district (Hoyle 2010). As a result, a decision can be made whether work should be undertaken in the area. It

also allows a company to compare the leakage levels in other DMAs and focus the resource effort on areas which will have the biggest impact on the leakage.

Figure 5.2: Leakage distribution observed over 2 days of measurement.

A common approach is to measure the flow of water in different areas of the network during the night, when the demand for water is at its lowest level. Any difference in the night flows may be considered as a leakage and is investigated (Fig. 5.2). The drawback of this approach is exposed when a sudden increase in demand occurs. This activity, referred to as unaccounted night use, leads to misreading the increase in usage of water as a leakage. Although considerable investments have been made, lack of metered households is a major cause of errors in leakage estimation as it is very difficult to distinguish between a sudden increase in demand and an actual leakage.

5.5 Leakage forecasting

As mentioned in the previous section, leakage can be decreased by applying ALC and responding to customers' reports in a timely manner, which requires the resources to be allocated in advance. In the UK it is a requirement set by OFWAT, that water companies present regular leakage forecasts and plan their resource effort in order to decrease leakage to an acceptable level. Together with a broad, five year business plan, more detailed forecasts are produced one year ahead, from April until March.

5.5.1 Factors influencing leakages

5.5.1.1 Seasonal factors

Leakage does not remain constant throughout the year and is subject to the influence of seasonal factors. Various weather effects can affect the rate and amount of leakage, such as temperature, rainfall, number of consecutive frost days or soil moisture deficit.

Figure 5.3: Example of a real leakage distribution at the company level throughout the year.

An example of a real leakage distribution at the company level over the period of one year can be seen in Fig. 5.3. It is common for two distinguished peaks to be observed during the year.

The smaller peak occurs around the summer period and may be explained by an unusual increase in demand due to unaccounted night use and an increase in temperature which causes AC-type (asbestos cement) pipes to expand and results in displacements, leading to pipe bursts.

The much more significant peak is usually observed in the winter which affects mainly iron pipes, due to the physical characteristics of the material. The iron has good contraction properties, but is much worse in expansion. Severn Trent Water has over 45 000 km of metallic pipes, and although the infrastructure is constantly upgraded, they still account for almost 90% of the network. More bursts occur

during winter as the water trapped in the soil freezes, and therefore increases its volume. Also, the temperature of water in the pipes is a major influence (Almeida & Ramos 2010), as it can affect the internal corrosion rate.

The seasonal factors have a big influence on the service levels and economic losses of water companies. It is however difficult to forecast the leakage, especially for 1 year ahead, based only on the weather factors. Research in the related field of electricity demand forecasting usually uses weather forecasts for up to 10 days ahead (Taylor & Buizza 2003). Therefore, although in the following chapters of the thesis the results with the weather factors as one of the inputs will be included, it is only to show that this can improve the accuracy, but at the moment does not have a practical application when forecasting for 1 year ahead.

5.5.1.2 Equivalent Service Pipe Bursts

Together with the expected leakage, the company presents the resource effort expressed in the form of Equivalent Service Pipe Bursts ($ESPB$). Different types of pipes have different flow rates and the leakage from these pipes cannot be compared in a straightforward manner. Therefore, $ESPB$ numbers represent monthly repair figures recalculated to account for different flow rates.

$$ESPB_{type} = leaks_{type} \times \frac{flowRate_{type}}{flowRate_{CSP}} \tag{5.1}$$

where *type* is a type of a pipe (Mains, Communication, etc.), *leaks* is the number of leaks, $flowRate$ (in $\frac{\mathrm{m}^3}{\mathrm{hr}}$) is a flow rate and $flowRate_{CSP}$ is a flow rate of Communication Service Pipe which is used as a reference value for recalculation.

Two types of $ESPB$ values can be distinguished:

- Detected: number of fixed leakage events found by the service engineers, recalculated to account for the pipe type

- Reported: number of fixed leakage events reported by customers, recalculated to account for the pipe type.

5.5.1.3 Natural Rate of Rise

In addition to $ESPB$ numbers, the Natural Rate of Rise (NRR) is used to indicate seasonality in the leakage. NRR $[Ml/day]$ relates to the underlying rate at which leakage increases within a network in the absence of any leak repairs. It is calculated based on each year's expected starting leakage, NRR Profile, which accounts for seasonal factors (peaks in summer and winter periods) and annual NRR (an overall increase in leakage throughout the year). Similar to $ESPB$, it is also created for detected and reported leakages and the sum of both represents total NRR or NRR_t.

5.5.2 Leakage forecasting methods

The company produces forecasts for monthly leakage and resource effort one year ahead at different organisational levels (company and regional). A lot of decisions on how to deal with the produced forecasts are based on the experience of the experts, which is usually not incorporated into the mathematical or forecasting models. Therefore it is a common practice to adjust the results obtained by those mathematical models according to the knowledge of some special events or other factors. This may increase the accuracy of the forecast if applied correctly, but on the other hand introduces biases and inconsistencies if incorporated incorrectly.

In the particular case of leakage forecasting, first the leakage target at the company level and related costs for the whole year are obtained based on the economic factors, required ELL and the constraints set up by OFWAT. Then, the forecasts for eight WRZ (Water Resource Zones) are generated, so that the leakage can be

distributed between them and throughout the year. The previous year's data is used as a baseline and it is judgementally adjusted to fit the requirements.

The forecasts are updated as time passes with the current data and numerous parameters, such as:

- Flow Rates of pipes, different for every type of pipe,
- $ESPB$ numbers,
- Hours to detect (HTD): hours/$ESPB$ describes how many hours it takes to detect, but not fix, the leak,
- Reported Leakage, which is often judgementally set by the expert,
- Estimated and measured demand, very often judgementally adjusted (based on the population, results obtained from the metered properties, weather forecasts),
- NRR profile,
- Actual hours of work to fix the leak (estimated); these are only hours spent on fixing the leak,
- Related costs

Many of these parameters are judgementally adjusted and relations between them are not fully understood or are not formally described. They can also be weak or even non-existent, as they have not been properly modelled and evaluated.

5.5.3 Issues with producing leakage forecasts

In general, producing forecasts for a long period ahead, such as 12 months is a difficult task, especially when one of the crucial factors is weather. One of the

ways to overcome this problem is to rely on historical data and the assumption that leakage can be influenced by the number of leaks found and fixed through detection and customer feedback ($ESPB$ numbers). Relying on the weather forecasts is not feasible, as 12 month ahead weather forecasts are not sufficiently accurate. Instead, seasonal factors, such as general decrease of temperature during the winter, or less rain during the summer, are included in the NRR figures which increase throughout the year with a high peak in a winter season. Another issue is the general influence of $ESPB$ numbers (both detected and reported) and the relationship between them. It is difficult to establish if investment made in detecting leaks and, therefore, increasing $ESPB$ detected numbers has a great influence on leakage values or if the change in leakage depends more on other factors including $ESPB$ reported numbers.

5.5.4 Current leakage forecasting procedure

The company carries out the leakage prediction using the estimated flow rates to calculate the $ESPBs$. This is done by calculating the adjustment factor so that the leakage matches the estimated start leakage and target leakage values. This gives an estimation of how much work needs to be carried out in order to bring the leakage down to the target level.

The following inputs used in the estimation model are known or assumed to be known:

- HTD - hours to detect: constant describing how many hours on average does it take to detect the leak
- start and target leakage
- annual NRR (detected and reported)

- number of leaks per type of pipe and the flow rates for each type of pipe in the past year
- NRR profile (distribution of NRR month to month)

The principal of the system is to adjust the ESPB values in such a way that the average distribution of the leakage over the course of the year matches the annual leakage target level. This is done using following procedure:

1. First start leakage is optimized (also known as NRR leakage). This is done by adjusting the $ESPB$ values by changing the adjustment factor.
2. The adjustment factor is increased from -100% by 10% with every step (where -100% means that the leakage is set to be without any active leakage control (ALC) - the number of detected $ESPBs = 0$).
3. In each step the $ESPBs$ are adjusted using the adjustment factor.
4. This allows to calculate the estimated savings.
5. To obtain the estimated leakage, the difference between start leakage and the estimated savings is calculated.
6. The optimization is finished when:
    - The adjustment factor is $>= 100\%$ and the average estimated leakage is bigger than the average leakage with ESPBs = 0, or
    - If the difference between the average estimated leakage and start leakage is < 0
7. The adjustment factor is then fine tuned by minimizing the objective function (minimize the total savings), so that the average estimated leakage equals to start leakage.

8. The second step is to optimize the target leakage.

9. The algorithm is very similar to the one used to optimize start leakage.

10. The difference which is taken into account during the adjustment is the one between average estimated leakage and the annual target leakage, instead of start leakage.

11. This step aims at calculating the additional resource effort ($ESPBs$) of bringing down the leakage from the starting point to the target annual value.

The simplified flowchart of the algorithm describe above can be seen on Fig. 5.4.

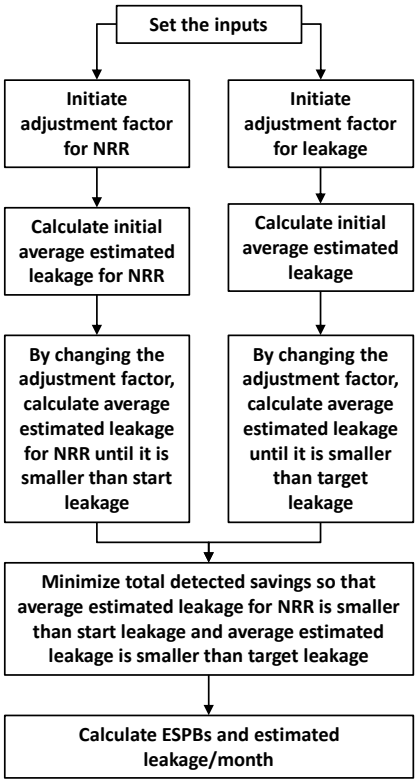

Figure 5.4: Flowchart of the leakage estimation algorithm used by the company

The accuracy of the predicted leakage using this forecasting method hasn't been assessed. The aim of the system was to ensure that the average leakage achieved at the end of the year meets the target leakage. This was done by varying the ESPBs across different periods of the year in order to plan the resource to be available when the big leakage events are most likely to occur. This was carried out through judgemental adjustments by the company expert.

As the current forecasting system does not use the historical leakage data and other relevant inputs to establish the predicted leakage for the upcoming year, a possibility of improving the system has opened. The analysis of the leakage forecasting system revealed several points:

- The relationship between ESPBs and the leakage was not fully understood
- The impact of ESPBs on leakage in different areas of operation was not known
- The ability to predict the leakage based on historical data was not understood
- From conversations with experts from the company, it has been established that months with highest leakage values also have the highest impact on the total leakage accumulated throughout the year.

In order to establish the relationships between relevant data, the fuzzy clustering techniques and evolving fuzzy systems described in previous chapters were investigated. In the next chapter the modified version of the fuzzy evolving algorithm will be presented, which was inspired by the problem of leakage forecasting.

5.6 Conclusions

In this chapter an overview of the issues related to water supply, management and distribution have been presented. A focus was placed on the problem of existing water leakage due to its social and economic impact. Leakage detection methods have been presented and the concept of a leakage forecasting process has been introduced. It can be seen that there is a need for a multivariate approach to forecasting the leakage, due to the necessity of resource planning and the existing non-linear relationships of leakage and other factors. In the next part of the thesis, a method based on fuzzy evolving approach (which was already described in the previous chapters) will be modified and adapted to suit the problem of leakage forecasting. The application and assessment of that method on the leakage forecasting, but also on other available datasets, will be presented in following chapters.

Chapter 6

Modified Evolving Takagi-Sugeno (Mod eTS) algorithm for forecasting

6.1 Introduction

In the previous chapter, the issues encountered in leakage forecasting were outlined. In this chapter a new algorithm, Mod eTS (Birek et al. 2014), will be introduced. The algorithm was developed in order to increase the accuracy of prediction of the leakage forecasts, especially for periods and areas with high level of leakage. Mod eTS is an algorithm belonging to a group of Evolving Takagi-Sugeno Fuzzy algorithms, described in Chapter 4. It incorporates a dynamic radius adjustment of each input variable of each cluster. Introducing a radius for each cluster dimension which is dynamically changing allows for better coverage of the data by clusters, as radii may differ in different dimensions. It also limits the number of clusters (and consecutively rules) as it is not necessary to create more clusters when data is

already well covered by the existing ones. This chapter will introduce the algorithm in Section 6.2 with more detailed description of how the algorithm can be utilized in forecasting in Section 6.3. In the last section, Section 6.4, the focus will be put on the main modification, the dynamic, unique radius of each cluster.

6.2 Mod eTS algorithm

The structure of the Mod eTS algorithm is similar to other eTS based algorithms. The approach is inspired by various Fuzzy Evolving methods: the normalisation is done using the maximum and minimum of available data, potential is calculated and updated using the standard eTS formula and parameters are updated using RLS algorithm. The difference lays in a way the radius of the clusters is calculated and updated and how the distance to each of the clusters is obtained. The steps of the algorithms are described below.

The algorithm can be initiated with a previously generated cluster structure or it can start from the first observed data sample. The data samples should be organized in row vectors with inputs followed by resulting output (Eq. 6.1) of the form:

$$x_k = \begin{bmatrix} x_{k1} & x_{k2} & \cdots & x_{kj} & \cdots & x_{kh} & y_k \end{bmatrix} \tag{6.1}$$

where k is the index of the data sample, $j = 1, \ldots, h$ is the input index and y is a resulting output. In addition to this, the initial values of the spread or cluster radius $r_{1,1j}$ for each input and output needs to be set up in advance. The radii will be updated dynamically as new clusters are added to a cluster structure. The radii are updated using the parameter γ which is set in advance empirically. The investigation of the influence of this parameter will be presented later in the following chapters.

The proposed Mod eTS algorithm contains the following steps.

1. Normalise the incoming data sample x_k so that the range of all the data samples, $x_{kj} \in [0, 1]$:

$$x_{kj} = \frac{x_{kj} - \min(x_{\_j})}{\max(x_{\_j}) - \min(x_{\_j})} \tag{6.2}$$

The $\min(x_{\_j})$ and $\max(x_{\_j})$ denote the minimum and the maximum of all previously obtained samples of j^{th} variable (component of data sample x_k). As the variables differ in their maximum and minimum values, it is recommended that they are normalized (Dovzan et al. 2012). Although the information of the maximum and minimum values of data may not be available for the user in real life, it is often possible to estimate the maximum and minimum values from a subset (training set) of available data or those values can be chosen empirically based on the expertise. Another approach to the problem has been described in Dovzan et al. (2012) where the normalization was done through pre-defined constants for each variable, and in Angelov et al. (2010), where normalization was done based on the recursive calculation of mean and standard deviation.

2. If the algorithm starts from an empty rule base and the first data sample x_1 is considered, then:

   a) Initialize the cluster structure c_1 with the first data sample $x_1 : c_1 = x_1$

   b) Set the initial potential $P_1(c_1)$ of the first cluster to 1: $P_1(c_1) = 1$.
   The formula for the potential $P_k(x_k)$ is given by:

$$P_k(x_k) = \frac{k-1}{(k-1)(\vartheta_k + 1) + \sigma_k - 2\nu_k} \tag{6.3}$$

   The description on how to calculate the values in the potential equation above has already been given through Eq. 4.2 in Chapter 4.
   The values used to calculate the potential (Eq. 6.3) of the data sample in the next steps need to be initialized as well: $\vartheta_1 = 0$, $\sigma_1 = 0$, $\nu_1 = 0$, $\beta_1 = \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix}$, where size of the β_1 vector is h (same as x_k)

c) Create the first rule based on the cluster centre c_1:

$$\textbf{IF } x_{11} \textbf{ is } A_{11} \textbf{ AND } x_{12} \textbf{ is } A_{12} \textbf{ AND } \ldots \textbf{ AND } x_{1h} \textbf{ is } A_{1h} \qquad (6.4)$$
$$\textbf{THEN } y_1 = a_{11}x_{11} + a_{12}x_{12} + \ldots + a_{1h}x_{1h} + a_{1h+1}$$

with A_{1j} being a Gaussian membership function calculated using Eq. A.9.

d) Set the number of rules R to 1: $R = 1$

e) Calculate the variance σ^2 (σ measures the width of the Gaussian function) of the Gaussian membership functions (Eq. A.9) of each fuzzy set in the first cluster based on the starting cluster radius $r_{1,1j}$:
$\sigma_{1j} = \frac{r_{1,1j}(\max(x_{\_j}) - \min(x_{\_j}))}{\sqrt{8}}$.
The initial radius $r_{1,1j}$ for each of the inputs is a tunable parameter which can be selected based on the results obtained on a subset of data, using a simple grid search. The influence of the choice of the initial parameter r will be investigated in the following chapters.

f) Set the parameter vector θ_1 of the consequent linear equation of the first rule to 1: $\theta_1 = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^{h+1}$, where $h+1$ indicates the number of columns in the vector (h being a number of inputs)

g) Set the global parameter matrix (if global learning is used) Θ of the consequent linear equations to θ_1, $\Theta = \theta_1$

h) Set the fuzzy weight λ_1 to 1

i) Initialize global/local covariance matrix Cov_1 with a high number Q multiplied by the $(h+1) \times (h+1)$ identity matrix:

$$Cov_1 = \begin{bmatrix} Q & 0 & \cdots & 0 \\ 0 & Q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Q \end{bmatrix}^{h+1 \times h+1} \qquad (6.5)$$

j) Initialize a regressor vector ψ_1 which will store weighted input values for the parameter estimation purpose $\psi_1 = \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix}^{h+1}$, $\Psi_1 = \psi_1$

3. If the cluster and rule structure is already initiated or $k > 1$:

a) Recursively calculate the value of the potential $P_k(x_k)$ of the data point x_k using (Eq. 6.3)

b) Update the potential of all already established clusters. The potentials of already created centres depend on the distances to all data points, therefore, it is necessary to update them.

$$P_k(c_i) = \frac{(k-1)\, P_{k-1}(c_i)}{k - 2 + P_{k-1}(c_i) + P_{k-1}(c_i) \sum_{j=1}^{h+1} \left(\frac{c_{ij} - x_{kj}}{r_{k,ij}} \right)^2} \tag{6.6}$$

c) The following steps carry on the process of deciding when to add a new cluster, change the already existing one or when to leave the cluster structure unchanged. Angelov et al. (2004) investigated and benchmarked a number of ways on how to make that choice. The method which resulted in lowest error involved comparing the potential $P_k(x_k)$ to maximum and minimum value of the potentials of all of the existing clusters and then making the decision based on the distance to the closest cluster. That method has also been applied in Mod eTS. The process is based on the values of the potential, distance between cluster centres and new data vector, centre radii and the threshold, which has been chosen arbitrary:

1: Compare the potential of the candidate cluster $P_k(x_k)$ with the updated potentials of all previously selected clusters. The reason the potential is also compared to the minimum value of the potentials from the existing clusters is that it makes it possible to create clusters in areas with less information as well. The minimization of the information potential allows recovering some of the missing details (Ramos & Dourado 2003).

2: $d_{\min} = \min_i (dist_i)$, where $dist_i = \left\| \frac{c_i - x_k}{r_i} \right\|^2$, $d_{\min}$ is the minimum distance $dist_i$ between the data point x_k and the cluster with the centre c_i with respect to r_i, r_i is the radii of the i^{th} cluster;

3: **if** $P_k(x_k) > \max(P_k(c_i))$ **or** $P_k(x_k) < \min(P_k(c_i))$ **then**

4: **if** $d_{\min} < 0.5$ **then** {change the already existing cluster}

5: The closest cluster centre c_i is replaced by the current data point x_k (the corresponding rule is also changed);

6: The potential of the changed cluster is replaced by the potential $P_k(x_k)$ of the data point x_k;

7: Update the radii $r_{k,ij}$ of the closest cluster:

$$r_{k,ij} = \gamma r_{(k-1),ij} + (1 - \gamma) c_{ij} \tag{6.7}$$

where $r_{k-1,ij}$ is the j^{th} radius input of the i^{th} cluster c_i at previous time step $k-1$ and γ is the weighting factor that is pre-set.

8: The linear parameters of the consequent part of the rule which was created from the replaced cluster remain the same, as well as the covariance matrix Cov_k;

9: **else** {add new cluster}

10: New cluster is added with the coordinates of data point x_k and the potential $P_k(x_k)$;

11: $R = R + 1$;

12: The initial vector of linear parameters θ_R is obtained based on the weighted average of all vectors from remaining fuzzy rules;

$$\theta_R = \sum_{i=1}^{R-1} \lambda_i \theta_i \tag{6.8}$$

13: For global learning, the global covariance matrix Cov_k needs to be extended and reset (Eq. 6.9) with $\rho = \left(R^2+1\right) / \left(R^2\right)$ being a resetting factor based on the current number of rules R and *cov* representing the elements of the covariance matrix at step $k-1$.

$$Cov_k = \begin{bmatrix} \rho cov_{11} & \cdots & \rho cov_{1R(n+1)} & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \rho cov_{R(n+1)1} & \cdots & \rho cov_{R(n+1)R(n+1)} & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & Q & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & \cdots & Q \end{bmatrix} \tag{6.9}$$

If local learning is used, the local covariance matrix is initiated using Eq. 6.5;

14: **end if**

15: **else if** $d_{\min} < 0.5$ **then** {ignore the sample}

16: Update the radius of the closest cluster c_i (Eq. 6.7) and proceed further;

17: **end if**

4. Update the parameters of the consequent part using the global or local RLS algorithm, as follows.

   a) Update the data matrix Ψ_k using the firing degrees $\tau_1, \tau_2, \ldots, \tau_R$ calculated using Eq. 3.26-3.27 and the extended input data vector x_k^e (Eq. 3.29), $\psi_i = \begin{bmatrix} \tau_i x_{k1} & \tau_i x_{k2} & \cdots \tau_i x_{kh} & \tau_i \end{bmatrix}$, $i = 1, \ldots, R$:

$$\Psi_k = \begin{bmatrix} \psi_1^T & \psi_2^T & \ldots & \psi_R^T \end{bmatrix}^T \tag{6.10}$$

   b) Apply the global or local RLS algorithm to estimate the parameters of the

linear consequent part of the fuzzy If-Then rules:

$$L = \frac{Cov_k \Psi_k}{1 + \Psi_k^T Cov_k \Psi_k} \tag{6.11}$$

$$\hat{\varepsilon} = y_k - \Psi_k^T \Theta_{k-1} \tag{6.12}$$

$$\Theta_k = \Theta_{k-1} + L\hat{\varepsilon} \tag{6.13}$$

$$Cov_{k+1} = Cov_k - L\Psi_k^T Cov_k \tag{6.14}$$

The main difference is that in local learning the parameters are estimated separately for each rule, using the local covariance matrix cov_i, local data matrix ψ_i and local vector of parameters θ_i at time instance k.

5. Estimate the output for the next period based on the input values and obtained parameter estimates given in Eq. 6.14 or in the vector form (for global learning):

$$\hat{y}_{k+1} = \Psi_k^T \Theta_k \tag{6.15}$$

or in local learning:

$$\hat{y}_{k+1} = \sum_{i=1}^{R} \psi_{ik}^T \theta_{ik} \tag{6.16}$$

6. Collect next data vector and go to step 3.

Figure 6.1: The flow-chart of the Mod eTS algorithm.

Fig. 6.1 illustrates the flow-chart of the Mod eTS. The markings in green show where main improvements over previous iterations of the algorithm are included.

To summarise, the main differences in the Mod eTS as compared to the previous methods lays in the way the radius is updated and how it is used. The updated radius is used in the modified potential update formula (Eq. 6.6) which is a novel approach as compared to other Evolving Fuzzy methods. The way the radius is updated itself is also changed. The γ value (the smoothing parameter) is used to smooth the values of centroids in the considered dimensions (Eq. 6.7). The reasoning behind this approach is explained in Section 6.4 of this chapter. Additionally, for the first time, the Evolving algorithm has been tailored to a forecasting problem, with the approach summarised in the Section below.

6.3 Application of Mod eTS algorithm to forecasting

The algorithm can be adapted and used in multivariate forecasting for n-periods ahead. First, the available historical data should be arranged into data points forming data vectors. The algorithm can start from the first data point and the cluster formed by the first data point, and afterwards, the rule structure is gradually generated through an evolving process until all k data points are assessed. Assuming that the input values $x_{(k+1)1}, x_{(k+1)2}, \ldots, x_{(k+1)h}$ are given, the forecast for the next period $\hat{y}_{k+1}$ can be calculated using the generated fuzzy If-Then rules and the estimated parameters through Takagi-Sugeno inference (Eq. 6.15). In the next step the vector of data is created using the obtained forecast $\hat{y}_{k+1}$ in place of the output, i.e.:

$$x_{k+1} = \begin{bmatrix} x_{(k+1)1} & x_{(k+1)2} & \cdots & x_{(k+1)h} & \hat{y}_{k+1} \end{bmatrix} \tag{6.17}$$

The new data point x_{k+1} is then used to update the rule structure through the Mod eTS algorithm. The approach is similar to the rolling forecast method where generated forecasts for period k are used to obtain a forecast for the next period $k+1$. The process continues until all desired forecasts $\hat{y}_{k+1}, \ldots, \hat{y}_{k+n}$ are obtained. The flow-chart of the Mod eTS algorithm for forecasting is given in Fig. 6.2.

Figure 6.2: The flow-chart of the Mod eTS algorithm for forecasting.

6.4 Investigation in dynamic radius

It is worth noting that the idea of dynamic radii has been included in the extensions of the classic eTS approach in the form of exTS in Angelov & Zhou (2006) and eTS+ in Angelov et al. (2010). The comparison and difference of the radii updates introduced in this approach and different fuzzy evolving algorithms is given in Table 6.1.

Table 6.1: Comparison of the different radii updates in different fuzzy evolving algorithms.

Method	Characteristic	Equation
exTS	use of the scatter	$r_{k,ij} = \gamma r_{(k-1),ij} + (1-\gamma)\, s_{k,lj}$, l - closest cluster index, γ - parameter, $s_{k,lj}$ - scatter: $s_{k,ij} = \sqrt{\frac{1}{N_{k,i}} \sum_{l=1}^{N_{k,i}} (c_{ij} - x_{lj})^2}$, $N_{k,i}$ - support of the i^{th} cluster
eTS+	use of the scatter	$r_{k,ij} = \gamma r_{(k-1),ij} + (1-\gamma)\, s_{k,ij}$, γ - parameter, $s_{k,ij}$ - scatter: $s_{k,ij} = \sqrt{\frac{1}{N_{k,i}-1} \sum_{l=1}^{N_{k,i}-1} (c_{ij} - x_{lj})^2}$, $N_{k,i}$ - support of the i^{th} cluster
Mod eTS	use of the cluster centre	$r_{k,ij} = \gamma r_{(k-1),ij} + (1-\gamma)\, c_{ij}$, γ - parameter

The radius in Mod eTS is updated using the following formula:

$$r_{k,ij} = \gamma r_{(k-1),ij} + (1-\gamma)\, c_{ij} \tag{6.18}$$

Radius is a crucial piece of information about the cluster, as data points which are outside of the radii have little or no influence on the potential of this cluster (Chiu 1994). The update of the radii is conducted at the stage of modifying the existing cluster. Each dimension (input variable) of the cluster is updated separately. A similar approach was implemented in Angelov & Zhou (2006) and Angelov et al. (2010), where the radius was calculated using the local scatter over the input data which resembles the variance. In the method proposed in this research, the scatter is not calculated, but instead the value of the centre in the considered dimension is used. This approach has been dictated by the importance of high values of inputs and is problem specific. The higher the cluster centre value is, the bigger the radius

and the higher the impact of that cluster will be. This will promote the clusters created in the regions with high values which is crucial for some applications, such as leakage forecasting (described in the previous Chapter). High values of leakage contribute considerably to the average leakage over a year, even though in the time series considered they do not occur frequently. It is worth noting that it is also important to choose appropriate smoothing parameter γ and initial values of the radii. Those can be established through grid search on a subset of data. Initial values of radii have been a subject of investigation in a number of papers (see review in Ramos & Dourado (2003)), therefore the conclusions from those can be used. It is important to remember that the final choice of parameters depends on the problem and the characteristics of time-series used.

6.5 Conclusions

In this chapter the principles of a novel Mod eTS were presented. In the next section, the algorithm will be applied to a leakage forecasting problem. The results will be analysed and the influence of tunable parameters, such as initial values of the spread or cluster radius $r_{1,1j}$ will be assessed. Finally the algorithm will be compared with other forecasting approaches and other Evolving Fuzzy algorithms in terms of accuracy and the performance (speed of execution and number of rules, when possible).

Chapter 7

Application of the Mod eTS algorithm to leakage forecasting

7.1 Introduction

In this chapter the novel Mod eTS fuzzy forecasting algorithm, introduced in Chapter 6 is applied to leakage forecasting problem described in Chapter 5.

The aim of this chapter is to describe the process of choosing the explanatory variables and the pre-processing steps on the provided datasets and to prove that the Mod eTS algorithm can perform better than other forecasting methods on those datasets. As only 9 datasets will be analysed in this chapter, it is not possible to generalize the conclusions. However having less data allows for more in depth analysis which will be carried out here. Also, in order to overcome this limitation additional analysis will be carried out in the next chapter in order to verify those findings.

The chapter can be split into two parts. In the first one the problem of the leakage

forecasting is described in more details and focuses on the provided data, especially on what pre-processing steps had to be carried out to enable the use of that data in the algorithm and what explanatory variables were used in the process.

The second part focuses on analysing the results from applying the Mod eTS algorithm to the leakage forecasting problem. The configurable parameters of the Mod eTS algorithm will be analysed to determine what the most appropriate configuration in order to generate the most accurate forecast is. The results from the application of the method to the leakage forecasting problem will be compared with the results obtained from using other fuzzy forecasting and statistical methods. Finally the chapter will conclude with final remarks and observations.

7.2 Understanding the data

The data that was used to evaluate the algorithm was kindly provided by Severn Trent Water - one of the biggest water supplying companies in UK. The company operates across Midlands and Wales and is structured around 8 distribution areas. The data that was provided was not suitable to be used in the algorithm directly, so the first step was to understand what it represented and how to pre-process it.

The leakage data was given as a weekly summary of leakage per District Metered Area (*DMA*). These are smaller areas (the area of operation is split into over 3200 DMA's) for which the flow at the isolated entry and at the exit is measured. The leakage is calculated by measuring the difference between the exit flow and the entry flow and subtracting that value by the metered and estimated usage of the water in that area. If the obtained value was high this could indicate the leakage and the area was being checked. The leakage value provided was an average leakage per day across a week and was expressed in litres.

Another important input given was the number of jobs carried out in each *DMA* for each type of pipe when the leakage was reported or detected. This value would later be used to calculate the Equivalent Service Pipe Bursts (*ESPB*), which is the normalised value indicating the amount of water lost in the leak and the amount of work needed to carry out repairs.

Lastly the Natural Rate of Rise (*NRR*) was provided, which was shown as a profile for each month summing up to 1. The example of such a profile can be seen on Fig. 7.1:

Figure 7.1: NRR profiles.

Profiles for both *NRR* reported (*NRRr*) and detected (*NRRd*) were provided. This profile is used by the company in conjunction with initial (start) leakage (*SL*) and the annual *NRR* (*aNRR*) to derive the value of the *NRR* for each month of the year. This is the approach documented in UKWIR (2005) and is also used in this research.

The data spanned from March 2007 until February 2012 (5 years) - this was limited by how much data was provided by the company.

7.3 Data pre-processing

7.3.1 Leakage data pre-processing

First, the leakage data and number of jobs had to be mapped to 8 water distribution areas. As the data was provided on *DMA* level, the mapping was used to collate and assign each data point to its distribution area. The leakage data was cleaned up from any empty or incorrect readings (obtained as a result of the meter errors). The direction on how to identify those was given after the conversation with the company expert - the values were either empty (value of 0) or negative.

The leakage data and data on number of jobs were provided on a week by week basis. As the aim of the forecast was to generate a monthly prediction, the data from each week for each of the *DMA* assigned to the distribution area was summed up, generating monthly leakage and job summaries for each water distribution area. The normalised sum of the leakage data (detected and reported) for each of the 8 regions can be seen on Fig. 7.2.

Figure 7.2: Leakage for all regions presented separately.

In order to generate a generic forecast at the company level, the leakage from each of the region was added. This resulted in the overall leakage at company level and is plotted on Fig. 7.3.

Figure 7.3: Combined company leakage.

The leakage at the company level will be used during the experiment establishing the influence of the weather factors on leakage.

7.3.2 ESPB data pre-processing

In order to obtain the *ESPBs*, each type of pipe fixed during the repair job had a weighting factor assigned. This indicated an impact on the leakage associated with the size (diameter) of the pipe. The weighting factor was higher for bigger pipes as the impact on leakage was also considered higher. The number of jobs per pipe were multiplied by the weighting factor on monthly basis and split between detected and reported leaks. This resulted in obtaining *ESPBs* (both detected and reported) for each distribution area per month.

7.3.3 NRR data pre-processing

The Natural Rate of Rise (NRR) was calculated using monthly $NRRp$ profiles for each year (the company determines those profiles on year by year basis), initial leakage values for each year (start leakage SL) and the annual $aNRR$ values that were provided by the company. The NRR was calculated separately for detected and reported leaks and then summed up to obtain overall annual NRR profile.

The NRR for first month of the year is calculated using the SL value:

$$\begin{aligned} NRR_R(1,y) &= SL(y) + aNRR_R(y)NRRp_R(1,y) \\ NRR_D(1,y) &= SL(y) + aNRR_D(y)NRRp_D(1,y) \\ NRR_A(1,y) &= NRR_D(1,y) + NRR_R(1,y) \end{aligned} \tag{7.1}$$

where y is the particular year the NRR is calculated for, NRR_R is reported NRR, NRR_D is detected NRR and NRR_A is an overall NRR.

Each subsequent month is then calculated using the following set of formulas for each year:

$$\begin{aligned} NRR_R(m,y) &= NRR_R(m-1,y) + aNRR_R(y)NRRp_R(m,y) \\ NRR_D(m,y) &= NRR_D(m-1,y) + aNRR_D(y)NRRp_D(m,y) \\ NRR_A(m,y) &= NRR_D(m,y) + NRR_R(m,y) \end{aligned} \tag{7.2}$$

where m is the month of the year for which the NRR is calculated for.

7.3.4 Weather data pre-processing

The weather data was obtained from the Met Office UK (2015) website which contains the historically recorded min. temperature, frost days and rainfall data for many weather stations in the UK. The data used in this study was taken from

the Wales - Midland region which corresponds to the STW operation area. It was used at the company level due to lack of weather variation across regions. Weather data is used in one of the experiments to establish if the knowledge of some of the weather aspects could increase the accuracy of the forecast. The data used in that experiment can be seen on Fig. 7.4.

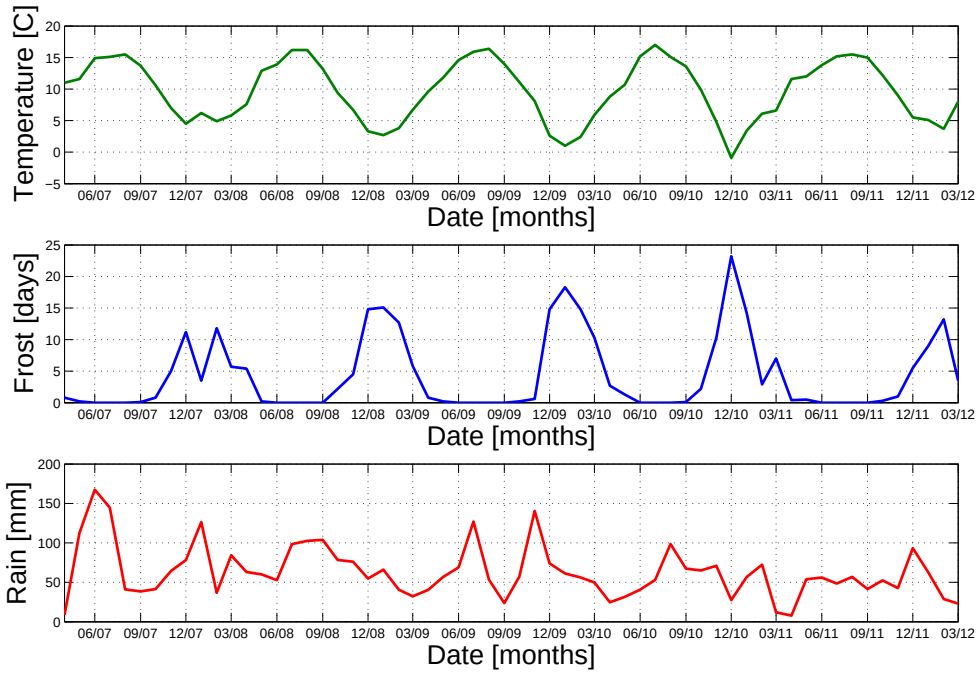

Figure 7.4: Weather conditions in West Midlands and Wales.

7.4 Data preparation for the use in the algorithm

The complete dataset consisted of 60 data points for each of the 8 regions. The datasets were divided into two parts: 4 years (48 months) of data was used as a training period and 1 year (12 months) for out of sample testing. As these are seasonal time-series it was necessary to include at least one full year (12 months) in

the out-of sample period and in order to cover as much data as possible in building the model the remaining data was used for training.

For algorithms which can utilize multiple inputs, the data vectors $x_k : k = 2, \ldots, 48$, were created from each dataset and consisted of four input values: y_{k-1} - previously obtained leakage, $u1_k$ - *ESPB detected* numbers, $u2_k$ - *ESPB reported* numbers and $u3_k$ - *NRR* values. The output value y_k is observed leakage. The choice of the parameters to include as inputs was dictated by the availability of the data and the discussion we had with the company experts.

$$x_k = \begin{bmatrix} y_{k-1} & u1_k & u2_k & u3_k & y_k \end{bmatrix}, k = 2, \ldots, 48 \tag{7.3}$$

Previous leakage value was used due to the longer lasting influence that it usually has on the next month of leakage. The *ESPB* values were used as the amount of work carried out in the previous month on fixing the leaks usually corresponds to the drop of leakage in the following month. The *NRR* values were used due to the fact that they normally incorporate the seasonal factor of leakage (higher leakage during the summer and winter). The additional input $u4_k$ was added for one of the experiments which corresponded to the weather factor (temperature, rainfall or number of air frost days). This input was used in order to determine if including the weather factors could increase the accuracy of the leakage forecast.

For time-series methods which use one input the leakage values were utilised.

For fuzzy clustering methods, the cluster structure was built using training data of 48 months separately for each of 8 datasets representing 8 regions of operation. The forecasting process was carried out using the last 12 months of data for a 12 month ahead period on a monthly basis. The forecast for testing data was calculated 1 period ahead and then the model evolved after each step based on that forecast (i.e forecast for period 49 was calculated using data 1-48, forecast for period 50 was calculated using data 1-48 + forecast of 49). The explanatory

variables were considered a known factor as they are set 12 months ahead.

7.5 Error measures

In order to report and compare the performance of the algorithm with other methods a set of standard error measures commonly used in forecasting applications will be used. These include: Mean Absolute Percentage Error (*MAPE*), Mean Absolute Scaled Error (*MASE*), Root Mean Squared Error (*RMSE*) and Non-dimensional Error Index (*NDEI*) and will be described below.

MAPE and *RMSE* are amongst the most used error measures. The lower the values of the *MAPE* and *RMSE* (the lower the forecasting error) the better:

$$MAPE = \frac{100\%}{n} \sum_{t=1}^{n} \left| \frac{y_t - \hat{y}_t}{y_t} \right| \tag{7.4}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^{n} (y_t - \hat{y}_t)^2} \tag{7.5}$$

where y_t is actual output value and $\hat{y}_t$ is predicted output at time (index) t. *MAPE* error measure is used due to its ability to express the result regardless of the scale at which the measure is computed as the result is presented in percentage. Also, the considered time-series have no zero values, which makes the use of *MAPE* possible.

As for *RMSE*, since the errors are squared before they are averaged, this gives a relatively high weight to large errors. This means that the *RMSE* is useful when large errors are particularly undesirable.

NDEI is the ratio of the *RMSE* to the overall standard deviation of the real output data and is being commonly used to compare a variety of Fuzzy Evolving methods

(simply due to the convenience in comparing the results).

$$NDEI = \frac{RMSE}{std(Y)} \tag{7.6}$$

where $std(Y)$ is a standard deviation of the actual output data Y.

Finally the Mean Absolute Scaled Error ($MASE$) is also used. It is an error measure proposed by Hyndman & Koehler (2006). It mitigates some issues present in the previous error measures. In case of $MAPE$, the disadvantage is that the error is not symmetric (interchanging y_t and $\hat{y}_t$ does not yield the same result), for $RMSE$ measure it is its sensitivity to outliers. The scaled error is a measure which scales the relative error based on the *in-sample* (training sample) Mean Average Error (MAE) of the naive forecast.

$$q_t = \frac{e_t}{\frac{1}{n-1}\sum_{i=2}^{n}|y_i - y_{i-1}|} \tag{7.7}$$

where e_t is a relative forecast error ($y_t - \hat{y}_t$) at time t and denominator is the average forecast error of the naive method (which takes the previous actual value as the forecasted value).

The $MASE$ measure is the mean over all of the absolute values of the scaled errors:

$$MASE = \frac{1}{n}\sum_{t=1}^{n}(|q_t|) \tag{7.8}$$

As data was split between training and testing periods, the errors for both were calculated. For methods that utilised all of the inputs, the accuracy on training data was calculated based on the final choice of parameters obtained after the training period. The errors were calculated starting from the 2nd sample until the last (48th) sample. In some cases, especially for time-series methods, the training

error had to be calculated on a different sample size. For example for Seasonal Naive method, the training error had to be calculated from the 13th sample as the seasonality was established to be 12 months (or 12 samples).

For testing period (out-of-sample data), the forecast was calculated using the parameters obtained from the training period, except for the evolving methods, which evolve the set of parameters over time as soon as data becomes available. As the *ESPBs* and *NRR* is planned a year in advance, real values of those inputs were used in generating the test results. However no real values of leakage were used during that period of testing, as the aim was to predict the leakage 12 months ahead. When previous leakage value was required as an input, the forecasted value of leakage for the previous period was used.

7.6 Analysis of the results for Mod eTS

In this section the results of applying the Mod eTS algorithm to the leakage forecasting problem will be analysed. The performance of the algorithm will be measured using 4 error measure techniques described in Section 7.5.

Two tables below (Table 7.1 and Table 7.2) show the results of applying the algorithm to the data from each of the 8 regions as well as to the sum of data from all regions (reg_all). The standard deviation and the mean of the error measures from all of the regions is also calculated. The tables also indicate what values of parameters have been used (initial radius r and radius change rate γ) to obtain those results. Those parameters are used in the formula that updates the radii of clusters, therefore impacting the final cluster (and rule structure). As a reminder the update formula is presented below:

$$r_{k,ij} = \gamma r_{(k-1),ij} + (1-\gamma)\, c_{ij} \tag{7.9}$$

Table 7.1 shows the results from training period (48 months of data for each of the regions). The table represents how well the created model fits the training data. In order to obtain the results presented in that table, the Mod eTS algorithm was applied to the training data using the combination of parameters from columns r and γ for each region separately. This combination of parameters was obtained through grid search (the resolution was 0.01) and the combination that resulted with the lowest *MASE* error was recorded. The regions with lowest (region 8) and highest (region 7) *MASE* error are highlighted in the table using green and red colours respectively. The mean and standard deviation of errors from all regions is also calculated.

Table 7.1: Training period results of Mod eTS for parameters minimizing *MASE*

region_code	**MASE**	**MAPE**	**RMSE**	**NDEI**	r	γ
reg_all	1.063	5.387	42.551	1.101	0.95	0.65
reg_1	1.081	5.688	5.899	0.726	0.80	0.70
reg_2	0.807	4.830	4.557	0.774	0.70	0.95
reg_3	1.035	5.494	5.299	0.919	1.00	0.35
reg_4	0.766	4.197	3.002	0.596	0.60	0.25
reg_5	0.802	4.212	3.475	0.576	0.90	0.80
reg_6	0.886	4.954	3.667	0.807	0.85	0.70
reg_7	1.425	8.021	7.302	1.630	0.60	0.40
reg_8	0.478	2.385	1.158	0.397	0.50	0.95
std	0.265	1.505	12.875	0.361	0.18	0.25
mean	0.927	5.019	8.545	0.836	0.77	0.64

Table 7.2 shows the results from testing (out-of-sample) period (12 months of data for each of the regions).

Table 7.2: Testing period results of Mod eTS for parameters minimizing *MASE*

region_code	MASE	MAPE	RMSE	NDEI
reg_all	0.5	2.734	12.688	1.007
reg_1	0.627	3.282	3.878	1.039
reg_2	0.499	3.431	1.615	0.892
reg_3	0.826	4.719	2.954	1.428
reg_4	0.733	4.629	2.297	1.381
reg_5	1.145	6.554	4.179	1.278
reg_6	0.469	3.297	1.541	1.262
reg_7	0.486	3.329	1.614	1.49
reg_8	0.852	4.561	1.894	1.223
std	0.229	1.178	3.538	0.204
avg	0.682	4.059	3.629	1.222

In this case the model obtained for the training data was applied to previously unseen testing data. The model was constantly evolving after each data sample was becoming available (after each forecast). The parameters used to obtain the results for each of the regions for that table were the same as in Table 7.1. The red and green colour represent the highest (region 5) and lowest (region 6) *MASE* errors. The high *MASE* error in region 5 (over 1, which means that it is worse than simple Naive method) indicate a lack of correlation with the explanatory variables. This can be explained by the unusual increase in reported leaks during test period even though leakage was actually decreasing, which was not the case during the training period when usually the increase in leakage sparked the increase in reported leaks. Fig. 7.5 shows data for region 5 and Fig. 7.6 for region 6 for reference.

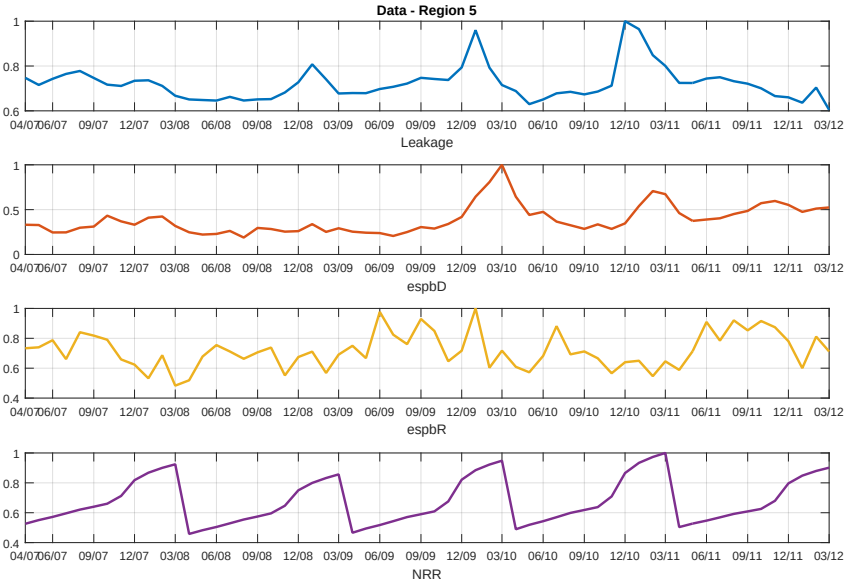

Figure 7.5: Data for region 5.

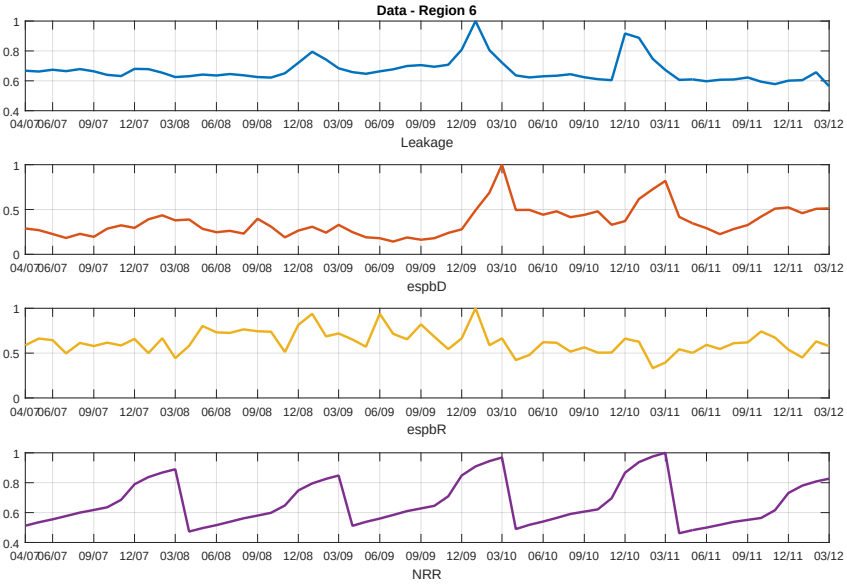

Figure 7.6: Data for region 6.

The results for both training and testing periods indicate that the forecast error is lower than the error of the simple Naive method (the average error of *MASE* is lower than 1 in both cases).

The *MAPE* has a value of 5% for training and 4% for testing periods on average, with the smallest error of 2.38% for training (region 8) and 3.28% for testing (region 1). The error was highest for region 7 (*MAPE* of 8.02%) for training period, and region 5 (*MAPE* of 6.55%) for testing period. In terms of error for both training and testing, it was easiest to predict for region 8 (average *MAPE* of 3.47%) and hardest for region 7 - with the highest average training and testing MAPE of 5.67%. The results also show the discrepancy between different error measures. While *RMSE* is the highest for reg_all dataset, the other error measures do not confirm that. This is due to the fact that *RMSE* measure penalizes the outliers much more than other methods, which focus more on the average error.

Based on the investigation above, in order to obtain best results the values of initial radius r and the rate of change γ should be set high. In most cases when initial value of r was higher than 0.65 (average of 0.77) and γ was set to value above 0.60 (average of 0.64) the accuracy of the forecast for most of the regions was highest. This aspect (sensitivity to change of parameters) is investigated in more details in the next subsection.

7.6.1 Parameter sensitivity analysis

The results obtained in the previous subsection indicate that the best performance is achieved when values of r and γ are set high. Fig. 7.7 shows the heat-map of the average *MASE* for the test period for all of the combinations of r and γ:

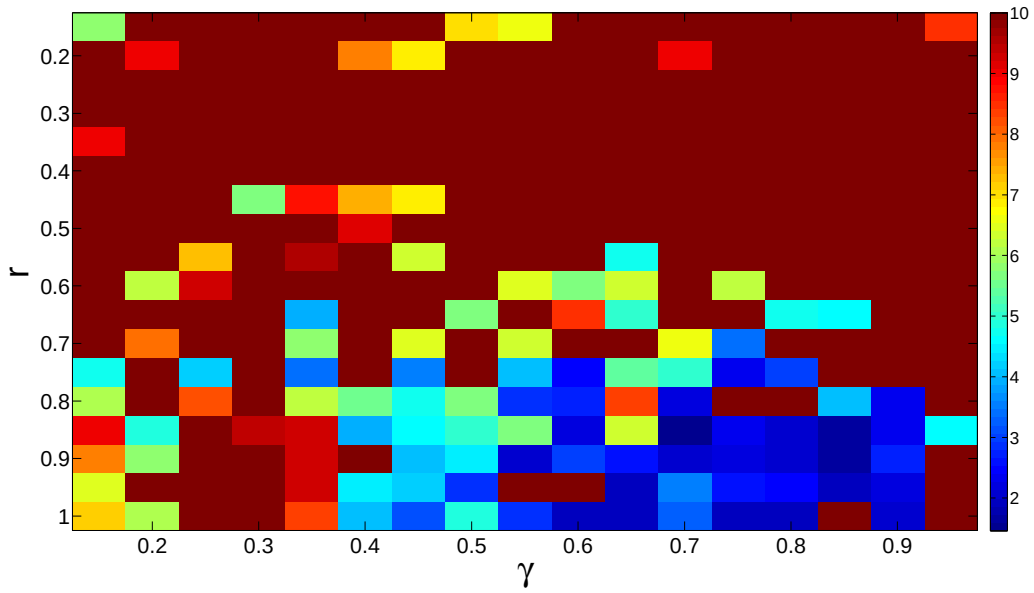

Figure 7.7: Average out-of-sample MASE error across all regions for varying values of r and γ.

It can be seen that the lowest errors are obtained when the model had values of r higher than 0.7 and values of γ higher than 0.5, which corresponds to findings from the previous subsection.

The reason for those thresholds can be explained by looking at the Eq. 7.9. The initial value of r has a big impact on the radius of newly generated clusters. When the value is low, there is a higher chance that when new data sample arrives at the next time step, the new cluster will be generated (the radius has an impact on the decision if a new cluster should be generated or if the existing cluster structure should be updated, see Section 6.2), which may lead to overfitting.

Likewise, values of γ play similar role, when the radii are updated. As an example, if the value of γ was set to 1, the radius would not update at all. By decreasing it's value, the radius updates are more weighted towards taking into account the values of the centroid and less the current radius, which increases the rate of change. By keeping the update rates (γ) of the cluster around 0.5 to 0.9, the radii updates at

moderate pace, but can still adapt to high values of leakage (where the values of c are high).

The number of rules created has direct impact on the speed of the algorithm, as more rules require more computational power to evaluate them at each step. The flexibility of algorithm suffers, when a lot of rules are generated during training period - the algorithm tends to over-fit the data and the predictive model becomes less flexible - see further results of Subtractive clustering algorithm in Section 7.7.1 where almost 40 rules were generated on average during training period).

The progress of creating clusters at each step of the algorithm is illustrated on Fig. 7.8:

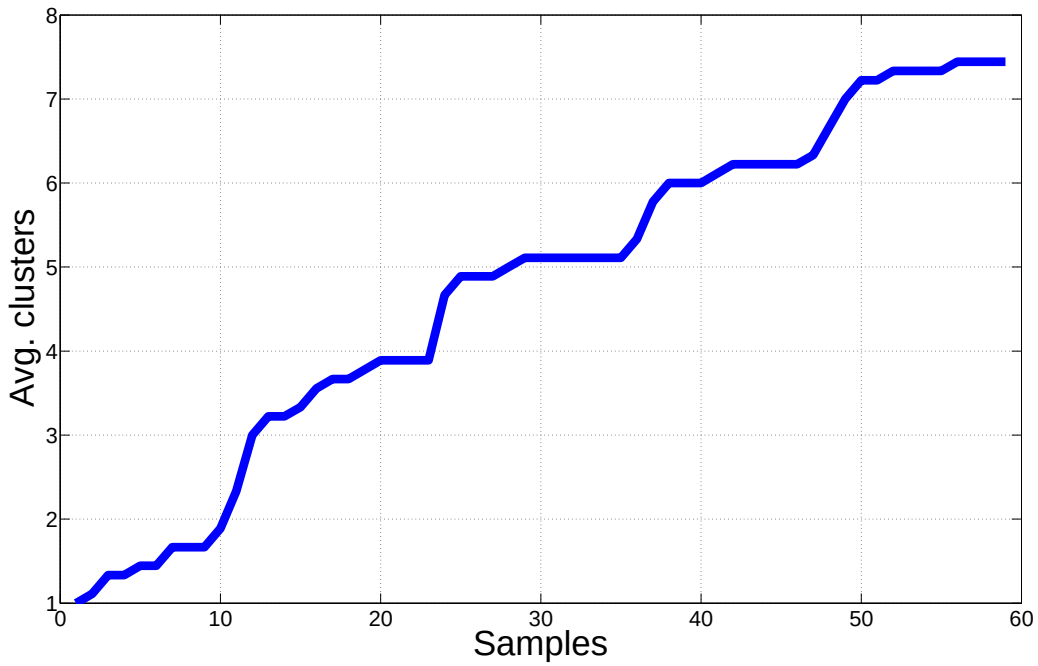

Figure 7.8: Number of clusters

The plot shows the number of clusters at each time instance averaged over all of the data regions. The average number of clusters created at the end of the forecasting process comes to 7.5. During the clustering process most of the clusters were

generated between 10-12 sample, 24 - 25 sample, 36 - 37 sample and 48 - 49 sample which corresponds to spikes in the leakage for all of the regions. This is in-line with the design of the algorithm which encourages clusters to be generated when leakage spikes up, so that those events can be recognized easier in the future.

7.6.2 Influence of weather factors

It is a known fact (see Chapter 5) that weather factors, such as temperature, frost days and rainfall, influence the amount of leakage in the water system. In this section the impact of using those factors on leakage forecast will be assessed. This will be achieved by adding one more input to the vector of explanatory variables. The specific weather data for each of the company regions of operation was not available at the time of this investigation. Nonetheless, data obtained from the UK Met Office for the region of Wales and Midlands allowed the use of the overall company leakage data to perform a basic assessment of the influence of the weather factors on the leakage forecast.

First, the influence of the weather factors on the choice of most appropriate parameters in order to obtain the most accurate forecast has been checked. Previously, the lowest forecasting error was obtained when the value of the parameters was higher than 0.7 for r and higher than 0.5 for γ.

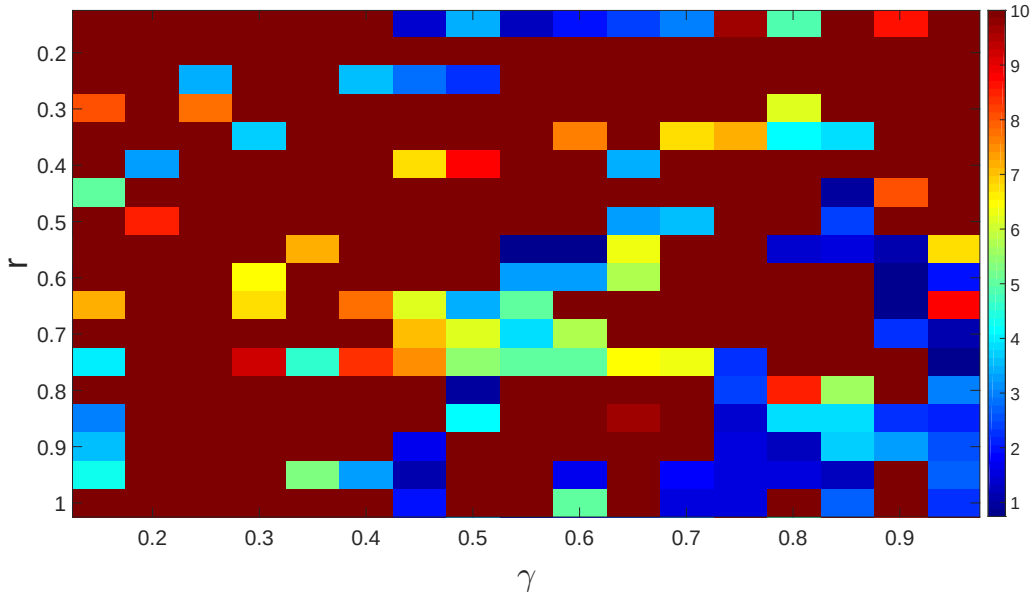

Figure 7.9: Out-of-sample MASE error for all regions combined with temperature data for varying values of r and γ.

Fig. 7.9 shows that the same applies if weather factors are added. The only difference being that the value of γ should be higher than 0.7.

To assess the influence of the additional inputs, 3 additional weather factors: min. temperature, number of air frost days and rainfall; were added to the input vector. The weather factors were added one at a time in order to assess the influence of each one of them on the forecast accuracy. Table 7.3 shows the results of the addition of those factors to the existing input vector.

Table 7.3: Comparison of results for data with and without the weather factors based on best testing $MASE$

	No weather		**Temp**		**Frost**		**Rain**	
Error	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
MASE	1.063	0.500	0.763	0.744	1.710	0.670	0.653	0.847
MAPE	5.387	2.734	3.853	4.167	9.138	3.753	3.131	4.768
RMSE	42.551	12.688	23.245	23.259	48.456	20.101	22.995	25.016
NDEI	1.101	1.007	0.601	1.846	1.254	1.595	0.595	1.985
r	0.95		0.65		0.65		1	
γ	0.65		0.9		0.9		0.95	
clusters	7	6	7	6	7	7	7	6

It can be seen that in most cases adding the weather factors decreased the training error. However the error for the testing period was still lower for dataset without weather factors included. Adding the information about the rainfall to the input vector decreased the training error, but increased the error during the testing period. This still resulted in the lowest combined testing and training error (average $MAPE$ of 3.95), however use of data without any additional weather information still resulted in the lowest testing $MASE$. The lower forecasting accuracy during the out-of-sample period indicate that forecasting with weather for longer forecasting horizons may be unreliable, which is in line with the general characteristics of weather forecasts, where the ability to accurately predict the weather for such a long forecast horizon is quite low (Taylor & Buizza 2003).

The values of parameters used to obtain the best results were still high, with initial radius r being 6.5 or 1 and radius change γ of 0.9.

The cluster progression and the final number of clusters were almost the same, with the exception of the frost, where 7 clusters were created in the end. This

means that the addition of inputs was not important enough to generate more clusters. One of the reasons for lack of cluster progression or increase in accuracy, is that the *NRR* values already incorporate the expected increase of leakage during certain times of the year, which makes additional incorporation of weather factors not having such a big influence on the forecast.

7.7 Comparison with other forecasting methods.

In this section the results obtained from the Mod eTS algorithm will be compared to the results from a number of fuzzy clustering and statistical forecasting methods. As the aim of this section is to determine if the Mod eTS algorithm can perform better than other methods on the considered dataset, only *MASE* error measure will be used. This approach is supported by the research by Hyndman & Koehler (2006), who suggest that measures based on scaled errors should become the standard approach in comparing forecast accuracy.

This section is split into two subsections. The first one compares the method to 4 other fuzzy forecasting and clustering methods. In the second subsection the algorithm will be compared to a number of simple, but effective and well established statistical methods. The performance on both training and testing parts of the dataset will be compared using parameters of each forecasting method optimised for both training and testing error. The number of clusters (rules) that were created while building the predictive model for fuzzy methods and an example plot of the forecasts for chosen regions will also be shown.

7.7.1 Fuzzy and clustering algorithms

In this subsection the performance of the Mod eTS will be compared to results obtained from a number of fuzzy clustering algorithms. 4 algorithms have been chosen for this comparison:

- eTS
- eTS+
- Subtractive clustering
- C-means fuzzy forecasting

eTS is used in the comparison, as it is a standard fuzzy evolving algorithm used as a foundation to develop Mod eTS. eTS+ is used as it is more advanced version of eTS with additional ways of handling cluster changes. Both methods were described in more detail in Chapter 4. The exTS was not included in this comparison as it is a version of the algorithm that includes some of the improvement from eTS+ (it was developed between eTS and eTS+). Two non-evolving fuzzy methods are also included. The subtractive clustering method and c-means fuzzy forecasting algorithm described in Chapter 3.

The tables below present the results of application of the Mod eTS algorithm and the other 4 fuzzy forecasting algorithms. Table 7.4 shows the results based on the parameters optimised on the training *MASE*, Table 7.5 on the training *MAPE*, Table 7.6 on the training *RMSE* and Table 7.7 on the training *NDEI*. The use of training error to optimise the parameters simulates the real life scenario (optimising parameters on the historical data). The combination of parameters r and γ is determined using grid search with the step of 0.01. The results for all regions are shown, for both training and testing periods, together with the average and the standard deviation of error. The average cluster numbers generated during

the forecasting process are also presented. The green and red colour show the lowest and highest error for each region across all methods for both training and testing data.

Table 7.4: Comparison of Mod eTS MASE accuracy results with other fuzzy forecasting methods - parameters determined to optimise the training MASE

	Mod eTS		**eTS**		**eTS+**		**c-means**		**Sub. clust.**	
	train.	**test.**	**train.**	**test.**	**train.**	**test.**	**train.**	**test.**	**train.**	**test.**
reg_all	0.631	0.704	0.756	1.903	0.547	2.066	0.574	1.399	0.000	2.518
reg_1	0.778	2.182	1.081	1.172	0.852	0.623	0.628	0.868	0.000	3.479
reg_2	0.464	1.002	5.279	18.344	0.832	1.538	0.494	0.614	0.000	1.090
reg_3	0.583	1.044	0.812	1.376	0.640	1.915	0.471	1.864	0.000	2.451
reg_4	0.553	1.580	5.248	5.535	0.725	2.860	0.571	2.216	0.000	2.921
reg_5	0.802	1.145	1.945	10.459	0.839	1.808	0.829	1.348	0.000	2.226
reg_6	0.630	0.796	0.707	0.989	0.833	0.966	0.663	1.632	0.000	1.470
reg_7	0.598	2.060	0.795	1.568	0.694	1.230	0.521	2.057	0.000	1.632
reg_8	0.478	0.852	0.764	2.366	0.772	1.827	0.495	1.114	0.000	2.291
std	0.116	0.549	1.927	5.917	0.106	0.659	0.112	0.539	0.000	0.742
avg	0.613	1.263	1.932	4.857	0.748	1.648	0.583	1.457	0.000	2.231
avg clust.	6.889		3.000		1.778		28.111		39.778	

Table 7.5: Comparison of Mod eTS MAPE accuracy results with other fuzzy forecasting methods - parameters determined to optimise the training MAPE

	Mod eTS		**eTS**		**eTS+**		**c-means**		**Sub. clust.**	
	train.	**test.**	**train.**	**test.**	**train.**	**test.**	**train.**	**test.**	**train.**	**test.**
reg_all	3.128	3.976	3.864	10.609	2.696	11.578	2.799	7.871	0.000	13.917
reg_1	4.128	11.876	5.908	6.328	4.397	3.293	3.332	4.690	0.000	18.571
reg_2	3.160	6.906	33.454	125.088	5.391	10.810	3.005	4.421	0.000	7.809
reg_3	2.915	5.920	4.183	7.787	3.245	10.792	2.412	10.635	0.000	13.800
reg_4	2.777	10.044	29.331	35.251	3.848	18.402	2.983	14.170	0.000	18.832
reg_5	4.212	6.554	10.822	60.771	4.255	10.640	4.183	7.959	0.000	12.853
reg_6	3.758	5.623	4.023	6.997	4.829	6.865	3.536	11.492	0.000	10.153
reg_7	3.511	13.900	4.581	10.631	4.012	8.406	3.065	14.018	0.000	11.126
reg_8	2.385	4.561	3.835	12.595	3.788	9.789	2.338	6.044	0.000	12.073
std	0.620	3.437	11.751	39.764	0.803	4.064	0.568	3.732	0.000	3.631
avg	3.331	7.707	11.111	30.673	4.051	10.064	3.073	9.033	0.000	13.237
avg clust.	6.889		3.667		1.778		28.111		39.778	

Table 7.6: Comparison of Mod eTS RMSE accuracy results with other fuzzy forecasting methods - parameters determined to optimise the training RMSE

	Mod eTS		**eTS**		**eTS+**		**c-means**		**Sub. clust.**	
	train.	**test.**	**train.**	**test.**	**train.**	**test.**	**train.**	**test.**	**train.**	**test.**
reg_all	19.681	17.933	23.263	44.787	19.615	54.809	18.197	33.281	0.000	55.185
reg_1	4.864	9.681	6.443	5.612	5.550	4.430	3.899	4.140	0.000	16.643
reg_2	1.781	4.192	26.895	72.578	3.167	4.602	1.985	2.052	0.000	4.308
reg_3	2.855	3.869	3.467	5.242	3.192	8.123	1.990	6.452	0.000	11.840
reg_4	3.175	4.461	19.315	16.477	3.322	9.384	2.516	6.206	0.000	9.417
reg_5	3.475	4.179	10.577	35.599	3.778	6.355	3.675	4.830	0.000	7.922
reg_6	1.944	2.211	2.481	2.782	2.746	3.023	3.179	3.914	0.000	4.833
reg_7	2.232	6.074	3.071	4.920	2.418	3.601	1.852	5.586	0.000	5.010
reg_8	1.158	1.894	1.921	4.800	1.899	4.028	1.713	2.541	0.000	6.398
std	5.770	5.008	9.794	24.485	5.547	16.593	5.262	9.724	0.000	16.125
avg	4.574	6.055	10.826	21.422	5.076	10.928	4.334	7.667	0.000	13.506
avg clust.	6.889		3.667		1.778		28.111		39.778	

Table 7.7: Comparison of Mod eTS NDEI accuracy results with other fuzzy forecasting methods - parameters determined to optimise the training NDEI

	Mod eTS		**eTS**		**eTS+**		**c-means**		**Sub. clust.**	
	train.	**test.**	**train.**	**test.**	**train.**	**test.**	**train.**	**test.**	**train.**	**test.**
reg_all	0.509	1.423	0.602	3.554	0.507	4.350	0.487	2.641	0.000	4.380
reg_1	0.598	2.594	0.792	1.504	0.683	1.187	0.496	1.109	0.000	4.459
reg_2	0.302	2.315	4.568	40.073	0.538	2.541	0.347	1.133	0.000	2.378
reg_3	0.495	1.870	0.601	2.534	0.554	3.928	0.353	3.119	0.000	5.724
reg_4	0.630	2.682	3.834	9.905	0.659	5.642	0.516	3.731	0.000	5.661
reg_5	0.576	1.278	1.753	10.885	0.626	1.943	0.628	1.477	0.000	2.422
reg_6	0.428	1.811	0.546	2.278	0.604	2.476	0.722	3.205	0.000	3.958
reg_7	0.498	5.610	0.686	4.544	0.540	3.326	0.421	5.159	0.000	4.627
reg_8	0.397	1.223	0.659	3.101	0.652	2.602	0.607	1.642	0.000	4.133
std	0.104	1.350	1.553	12.230	0.063	1.356	0.127	1.370	0.000	1.188
avg	0.493	2.312	1.560	8.709	0.596	3.110	0.509	2.580	0.000	4.194
avg clust.	6.889		3.667		1.778		28.111		39.778	

The best average results for the testing period are achieved by the Mod eTS for all 4 error measures. The method also has the lowest standard deviation of error, which proves consistency in the obtained results. Low standard deviation means that method consistently performs better on average, which is one of the most important measures in forecasting (Hyndman & Athanasopoulos 2014). The c-means algorithm was the second best with eTS being the least accurate for both the training and the testing period. In terms of the training error, the subtractive clustering has the lowest training error, however this is because it generates a new cluster for each data sample in the training period, which leads to a considerable over-fitting of data for the testing period (Tables 7.4, 7.5, 7.6, 7.7,). When subtractive clustering is taken out of the picture for training period, c-means clustering beats other methods when training period is considered for 3 out of 4 error measures. The Mod eTS is close behind the c-means algorithm, with training *MASE* of 0.613, *MAPE* of 3.331, *RMSE* 4.574 and beating it with *NDEI* of 0.493 (c-means clustering in this case has *NDEI* of 0.509). The slight variation in the results among various error measures and regions is normal. As described in Section 7.5 and indicated in Hyndman & Koehler (2006), various error measures suffer from a bias depending on how they are calculated.

An additional Table B.1 in the Appendix shows results with the parameters optimised on the testing *MASE*.

When looking at the number of clusters generated during the forecasting process, it can be seen that c-means needs to generate a considerable amount of clusters to achieve high accuracy. Another algorithm with higher than usual cluster count is the subtractive clustering algorithm, which generated over 39 clusters on average (Table 7.4). eTS+ generated the least number of rules on average due to the more advanced method of keeping control over the size of the rule base through the mechanism of disabling some of the obsolete rules. The Mod eTS algorithm generated about 7 clusters on average.

Fig. 7.10 presents the evolution of the rule base averaged over all datasets.

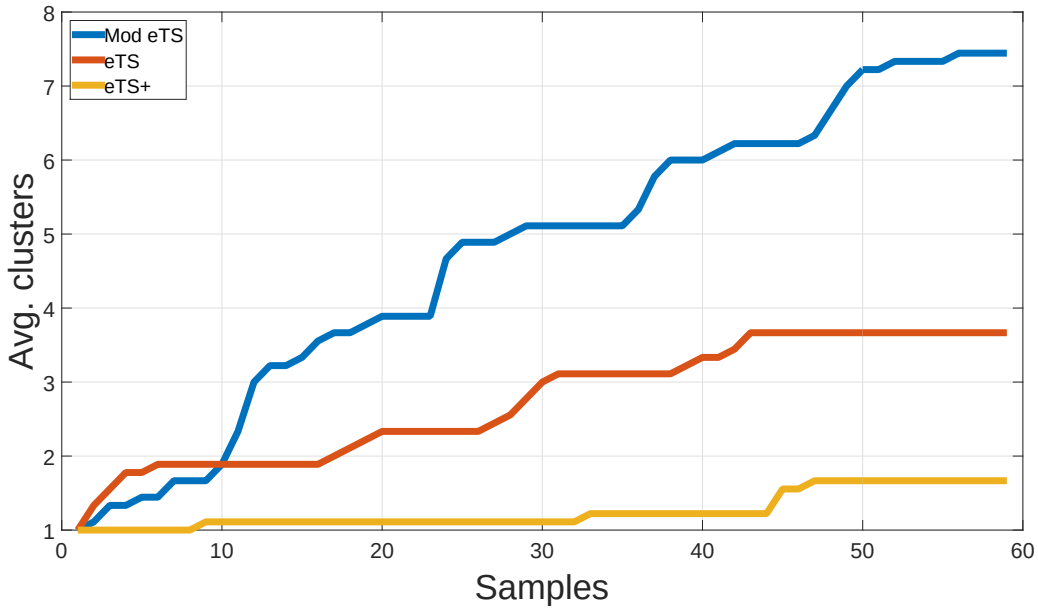

Figure 7.10: Rule evolution of Mod eTS, eTS, eTS+ averaged over all datasets.

The plot does not contain the cluster counts for c-means and subtractive clustering as they have a fixed cluster structure for all samples. The average number of clusters was 28.11 for c-means and 39.78 for subtractive clustering. For evolving methods, it can be seen that the Mod eTS generates more clusters on average than the other methods. With eTS+ having more strict rules around rule generation it comes with no surprise that it generates the least amount of clusters. As mentioned in the previous section, Mod eTS generates new clusters around areas of data-space with higher than usual leakage values, which corresponds to the overall higher number of generated rules.

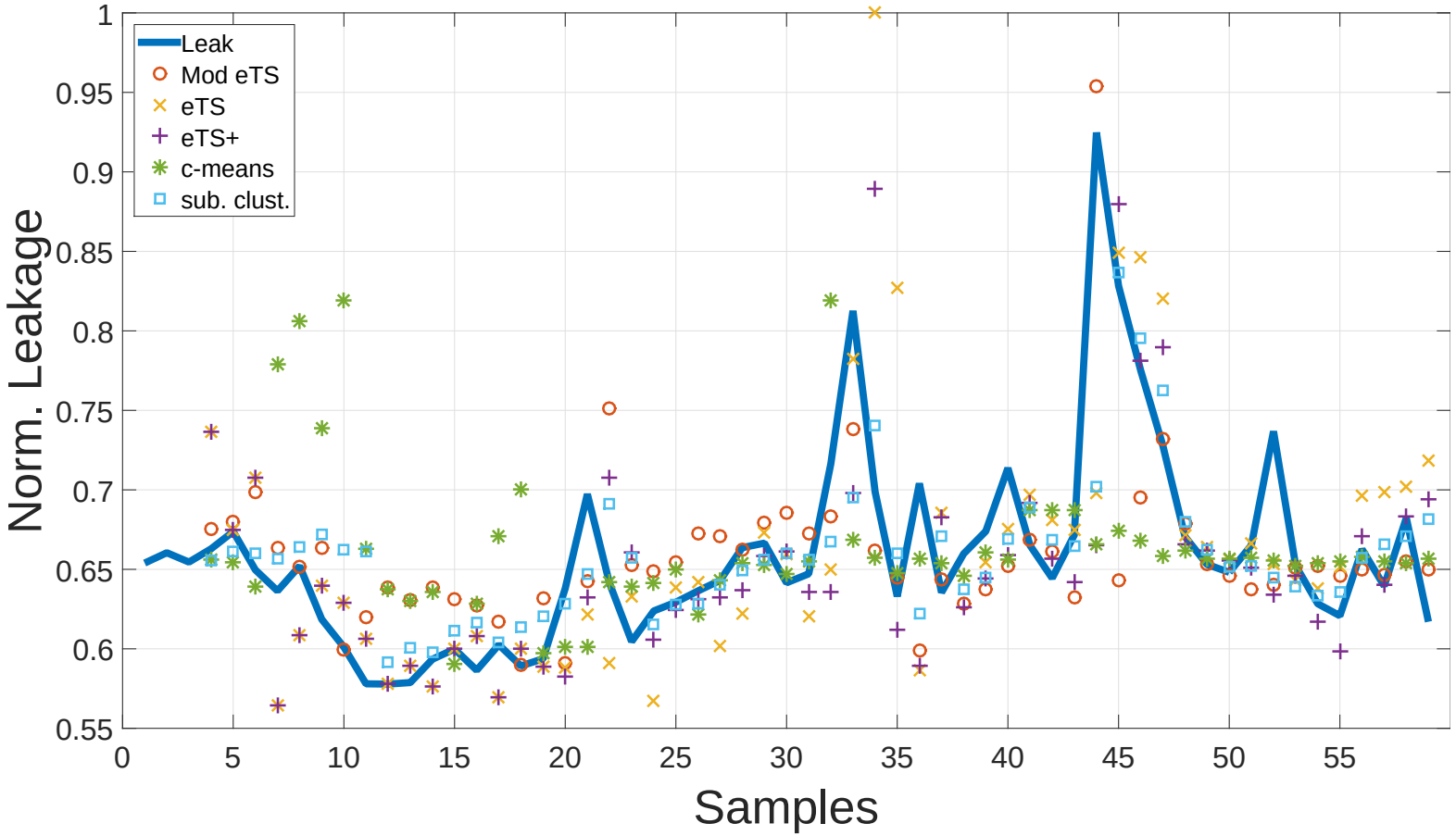

Figure 7.11: Leakage and combined forecast plot (training and testing forecast) of 5 years of data for one of the regions of operation.

In order to visualise the forecasts from different methods, Fig. 7.11 shows the leakage and forecast plots for one of the regions of operation. The forecast presented here is the combined training and testing forecast. For the first 48 samples the forecast was calculated using the model build on those samples. For the last 12 samples the forecast was calculated 1 step ahead (the model was evolving after each forecast, however no real leakage values were used - the forecasted values were used to evolve the model in order to simulate 12 step ahead real-world scenario). It can be seen that Mod eTS usually over-forecasts slightly, but in most cases is able to predict the more significant spikes in leakage, which other methods struggle to do.

7.7.2 Statistical forecasting methods

In this subsection the Mod eTS algorithm is compared with 3 statistical methods:

- Multiple linear regression (MLR)
- Seasonal Naive (S-Naive)
- Holt-Winters method (HWM)

The chosen methods are established and effective forecasting algorithms which, in a lot of cases, are difficult to be outperformed. All of them, apart from MLR, are time-series methods (they are not utilizing the additional inputs) but only use the past values of the leakage.

The MLR method is the only one utilizing the explanatory variables. It is a simple method utilizing multiple inputs, which tries to model a relationship between two or more explanatory variables by fitting a linear equation. It is a good benchmark for problems using explanatory variables, as the method itself is simple and only uses the Least Squares algorithm to estimate the parameters of linear equation.

The Seasonal Naive method is used rather than Naive to account for seasonality of the time series. The S-Naive method is a modification of the Naive method, which takes into account the seasonality of the time-series. Instead of using the previous value of the output as a forecast it will take into account the last 12 month values of leakage and simply apply it as a forecast for the next period. In this case the seasonality was established as 12 (due to the fact that year has 12 months), meaning the predicted value for March 2008 is simply a value from March 2007. In general, Naive methods should be used in benchmarks, as they are frequently difficult to beat (Hyndman & Athanasopoulos 2014). If the new method does not perform better, the it is generally not worth considering.

Finally the Holt-Winters method (HWM) is used as it's very popular in business setting (Goodwin 2010). HWM comprises of forecast equation which takes into account the level, trend and the seasonality of the data. Many companies use it, because it is simple, has low data-storage requirements, and is easily automated. It has the advantage of being able to adapt to changes in trends and seasonal patterns when they occur. This means that changes in usage or demand (for example for water) can all be accommodated.

All of those methods are well established, standard statistical forecasting methods which serve as appropriate benchmark to a Mod eTS method.

Below, the tables show the results of the comparison. As previously, the MASE (Table 7.8), MAPE (Table 7.9), RMSE (Table 7.10) and NDEI (Table 7.11) error measures are used to compare the algorithms. Both training and testing periods are assessed for all regions and an average and standard deviation of the accuracy is calculated. The tables shows the results obtained from optimizing the algorithm parameters based on training period. The green and red colours highlight the methods which achieved the highest and lowest accuracy for each of the regions.

An additional Table B.2 in the Appendix shows the results when parameters are optimised on testing period.

Table 7.8: Comparison of Mod eTS accuracy results with other statistical forecasting methods - parameters determined to achieve best training MASE

	Mod eTS		**S-Naive**		**MLR**		**HWM**	
	train.	**test.**	**train.**	**test.**	**train.**	**test.**	**train.**	**test.**
reg_all	0.631	0.704	0.911	1.693	0.734	1.364	0.716	3.490
reg_1	0.778	2.182	1.334	1.647	0.911	0.633	0.659	3.742
reg_2	0.464	1.002	0.839	1.895	0.838	0.618	0.733	2.114
reg_3	0.583	1.044	0.643	1.622	0.724	1.532	0.622	1.776
reg_4	0.553	1.580	1.012	2.431	0.744	2.169	0.770	3.782
reg_5	0.802	1.145	1.413	2.739	0.838	1.808	0.825	4.817
reg_6	0.630	0.796	1.154	1.643	0.833	0.965	0.711	2.728
reg_7	0.598	2.060	1.329	0.964	0.695	1.228	0.764	0.717
reg_8	0.478	0.852	0.815	1.599	0.772	1.827	0.708	2.828
std	0.116	0.549	0.271	0.515	0.071	0.542	0.060	1.234
avg	0.613	1.263	1.050	1.804	0.788	1.349	0.723	2.888
avg both	0.938		1.427		1.069		1.806	

Table 7.9: Comparison of Mod eTS accuracy results with other statistical forecasting methods - parameters determined to achieve best training MAPE

	Mod eTS		**S-Naive**		**MLR**		**HWM**	
	train.	**test.**	**train.**	**test.**	**train.**	**test.**	**train.**	**test.**
reg_all	3.128	3.976	5.217	11.017	3.597	7.684	4.761	27.736
reg_1	4.128	11.876	8.268	10.605	4.668	3.348	4.616	29.583
reg_2	3.160	6.906	5.522	14.835	5.405	4.396	6.276	21.513
reg_3	2.915	5.920	3.540	10.581	3.670	8.682	4.172	14.188
reg_4	2.777	10.044	5.947	17.276	3.944	13.913	5.816	35.972
reg_5	4.212	6.554	8.570	18.190	4.252	10.638	5.586	37.788
reg_6	3.758	5.623	7.931	13.569	4.829	6.863	5.536	27.203
reg_7	3.511	13.900	9.559	7.700	4.022	8.399	6.119	6.629
reg_8	2.385	4.561	4.446	9.550	3.787	9.791	4.064	18.171
std	0.620	3.437	2.082	3.587	0.609	3.191	0.831	10.162
avg	3.331	7.707	6.556	12.592	4.242	8.191	5.216	24.309
avg both	5.519		9.574		6.216		14.763	

Table 7.10: Comparison of Mod eTS accuracy results with other statistical forecasting methods - parameters determined to achieve best training RMSE

	Mod eTS		**S-Nave**		**MLR**		**HWM**	
	train.	**test.**	**train.**	**test.**	**train.**	**test.**	**train.**	**test.**
reg_all	19.681	17.933	30.080	68.746	23.833	35.091	37.383	112.484
reg_1	4.864	9.681	7.769	12.065	5.747	3.986	6.588	23.269
reg_2	1.781	4.192	4.521	9.178	3.137	2.637	4.915	11.496
reg_3	2.855	3.869	4.319	10.158	3.412	5.792	5.111	10.564
reg_4	3.175	4.461	4.541	10.870	3.297	6.535	5.240	16.900
reg_5	3.475	4.179	5.397	11.570	3.778	6.354	5.153	20.244
reg_6	1.944	2.211	3.712	7.021	2.746	3.027	3.947	9.993
reg_7	2.232	6.074	5.074	4.661	2.417	3.600	4.282	3.012
reg_8	1.158	1.894	2.321	5.002	1.899	4.029	2.865	7.330
std	5.770	5.008	8.582	20.161	6.927	10.297	10.921	33.812
avg	4.574	6.055	7.526	15.475	5.585	7.895	8.387	23.921
avg both	5.314		11.500		6.740		16.154	

Table 7.11: Comparison of Mod eTS accuracy results with other statistical forecasting methods - parameters determined to achieve best training NDEI

	Mod eTS		**S-Naive**		**MLR**		**HWM**	
	train.	**test.**	**train.**	**test.**	**train.**	**test.**	**train.**	**test.**
reg_all	0.509	1.423	0.709	5.456	0.617	2.785	0.774	8.927
reg_1	0.598	2.594	0.891	3.232	0.707	1.068	0.762	6.234
reg_2	0.302	2.315	0.719	5.068	0.533	1.456	0.658	6.347
reg_3	0.495	1.870	0.687	4.911	0.592	2.800	0.702	5.107
reg_4	0.630	2.682	0.835	6.535	0.654	3.929	0.816	10.160
reg_5	0.576	1.278	0.817	3.538	0.626	1.943	0.713	6.190
reg_6	0.428	1.811	0.753	5.750	0.604	2.479	0.707	8.184
reg_7	0.498	5.610	1.079	4.305	0.540	3.325	0.817	2.782
reg_8	0.397	1.223	0.733	3.232	0.652	2.603	0.800	4.735
std	0.104	1.350	0.123	1.174	0.055	0.891	0.057	2.269
avg	0.493	2.312	0.803	4.670	0.614	2.488	0.750	6.519
avg both	1.402		2.736		1.551		3.634	

The 2 methods: Seasonal Naive and MLR do not have any specific configurable parameters (such as r and γ for Mod eTS, or the level, trend and seasonality parameters for HWM) that need to be fixed before the forecast can be calculated (MLR has parameters that can be estimated based on the characteristic of time-series using, for example, Least Squares method).

The results presented in Tables 7.8 - 7.11 show that Mod eTS achieves the lowest overall *MASE*, *MAPE*, *RMSE* and *NDEI* errors when looking at average training and average testing error across all of the regions, as well as the accumulative average of both. With *MASE* value of 0.938, it is the only method to beat the simple Naive forecast (as the *MASE* error is lower than 1). HWM is not coping well, and with the average *MASE* of 1.806 it has the highest error out of all of the considered statistical methods, due to high testing error (2.89). This is also confirmed when looking at other error measures. MLR method achieves second highest accuracy, with good balance between training and testing error. That method also performs most consistently with lowest standard deviation across *MAPE* and *NDEI* error measures. Seasonal Naive has the least accurate forecast for training period across 3 out of 4 methods and was only been able to beat the HWM in terms of the average accuracy (*MASE* of 1.427, *MAPE* of 9.574, *RMSE* of 11.5 and *NDEI* of 2.736). Again, small discrepancies across different error measures are normal, but consistent results across most of the regions and error measures prove that the Mod eTS performs best out of the considered algorithms.

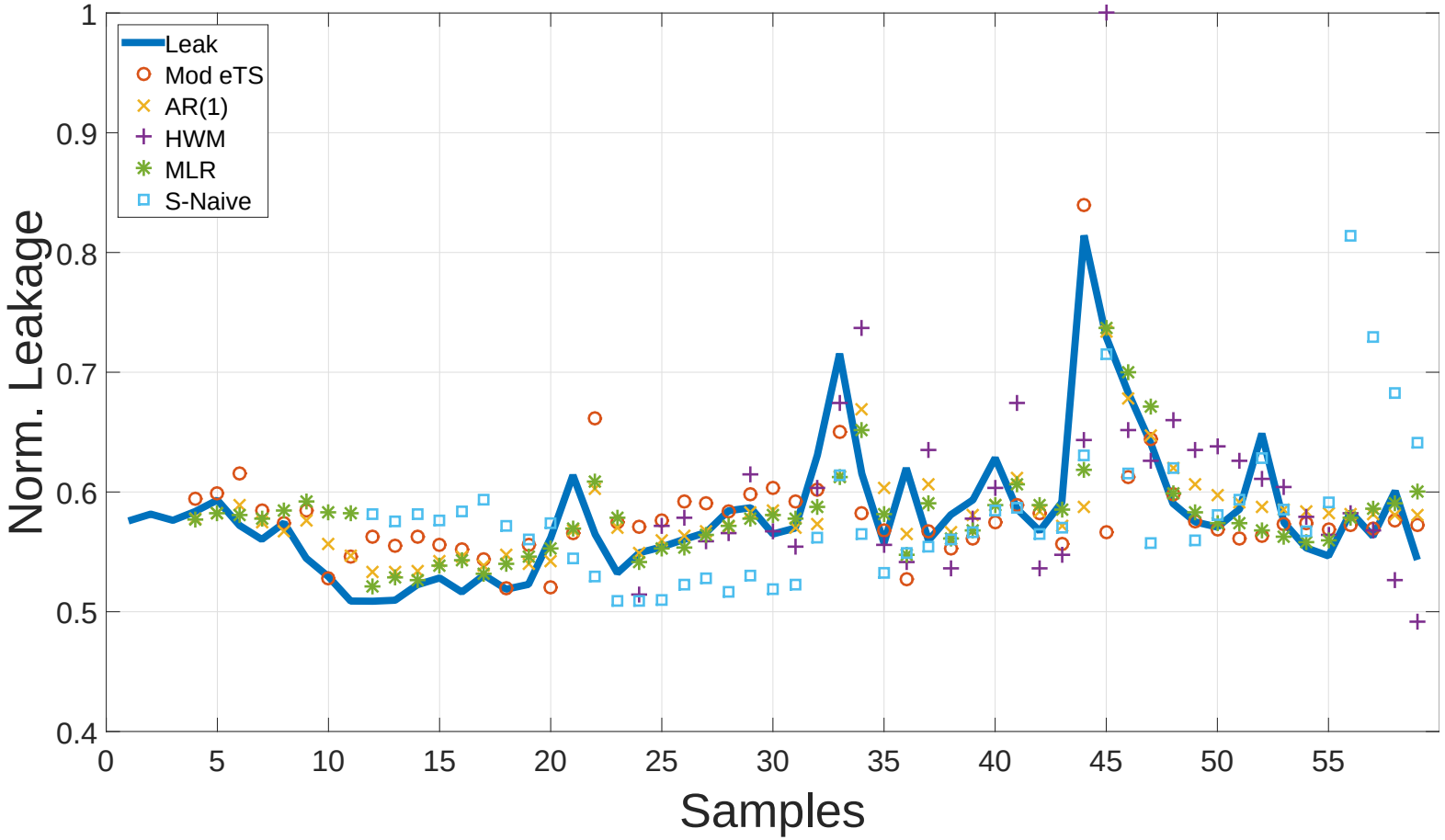

Figure 7.12: Normalized leakage and forecast of 5 years of data for one of the regions of operation.

Similar to when assessment of the fuzzy forecasting algorithms was carried out, Fig. 7.12 shows a plot of the real leakage and the forecasts for Mod eTS and all other 4 statistical methods for one of the regions of operation. It can be seen that Seasonal Naive method is not performing too well due to differences in leakage in the consecutive years. Even though HWM predicts the peak of leakage at the sample 33, it overshoots the prediction for the peak of leakage at sample 45, which in turn decreases the overall accuracy of the forecast and explains the lower accuracy of the predictions. The Mod eTS is able to more accurately predict the peaks and the size of them. The higher accuracy of prediction is due to the fact that those statistical methods do not make use of the explanatory variables, and for the one that makes use of them (MLR) it assumes that the relationship between those variables is linear.

7.8 Conclusions

In the first part of this chapter the detailed description of the data that was provided for this project and the data pre-processing steps were explained. Considerable amount of work had to be carried out in order to pre-process data to be suitable for use in the algorithms, which only proves that data pre-processing plays a very important role in the overall process of forecasting. As it has been shown later in this chapter, adding new explanatory variables does not always increase the accuracy, but may actually decrease it due to increase in noise and unnecessary complexity. Therefore carefully choosing the inputs and the explanatory variables may considerably increase the accuracy of the forecast.

In the second part of the chapter a number of fuzzy algorithms and some widely accepted statistical methods were compared with the novel Mod eTS algorithm and applied to the leakage forecasting problem. The results showed that the use of the Mod eTS algorithm resulted in smaller errors as compared to other

fuzzy forecasting and statistical methods. The algorithm can be used effectively in forecasting and performs well on the tested datasets. Results obtained by applying the method to real-world leakage data indicate that the proposed method generally performs better than other methods for testing (out-of-sample) data. This proves that the model built on the training data-set can be reliably applied to the unseen testing data, and the confidence that this model will produce accurate forecast is higher as compared to other methods.

In terms of the fuzzy rules, when compared with other standard fuzzy clustering forecasting methods, the Mod eTS generates less rules, which yielded (in this case) better results for test data. The comparison with other fuzzy evolving algorithms, which also utilize radii update, showed lower values of error for testing period but generated more rules. This is caused by the lack of more complex rule management in Mod eTS. However, decreasing the number of rules was not the aim when the Mod eTS algorithm was being developed, as usually deceasing number of rules improves the speed of the execution of the algorithm, which was not an issue in this case.

It is worth noting though, that the Mod eTS method is sensitive to the choice of the configurable parameters (r and γ), which can be seen when looking at the accuracy heat-map (Fig. 7.7). The recommendation on what range of parameters should be considered in order to obtain the best results was made, but it may require confirmation on other datasets.

The forecasting accuracy achieved by applying Mod eTS needs to be verified on other datasets as well, to assess a problem specific approach to rule update algorithm. In the next chapter the Mod eTS algorithm will be applied to the artificial Mackey-Glass time-series in order to verify the conclusions from this chapter.

Chapter 8

Evaluation of the Mod eTS algorithm on an additional data set

8.1 Introduction

In the previous chapter the Mod eTS algorithm has been applied and evaluated on the leakage forecasting problem. In this chapter the Mod eTS algorithm will be tested on an additional dataset in order to establish if the conclusions and results from the previous chapter are still applicable to other data sets. The data used in this chapter is a well known non-linear time-series which is generated by the Mackey-Glass process. In addition to the Mod eTS three other algorithms will be evaluated and the accuracy results will be compared. This will be achieved by using the same error measures as in Chapter 7: *MASE*, *MAPE*, *RMSE* and *NDEI*. The evaluation of the algorithms will also include sensitivity to parameters change and the analysis of the number of generated clusters (fuzzy If-then rules).

The chapter is structured as follows. In Section 8.2, the generation process of the test time-series will be explained. The major part of this chapter is included in Section 8.3, where analysis of the application of the algorithms to the Mackey-Glass time-series will be carried out. The last Section 8.4 will conclude the chapter.

8.2 Testing environment

8.2.1 Mackey-Glass time series

To perform the further assessment and the comparison of the algorithms, the time series generated from the Mackey-Glass equation (Mackey & Glass 1977) will be used. Mackey-Glass equation generates a chaotic, non-periodic and non-convergence time series, which has been commonly used to asses the performance and compare various fuzzy evolving algorithms. The time series is generated using the following equation (8.1):

$$\frac{dx}{dt} = \beta \frac{x_{t-\tau}}{1 + {x_{t-\tau}}^n} - \gamma x_t, \quad \gamma, \beta, n > 0, \tag{8.1}$$

where β, γ, τ, n are real numbers, and $x_{t-\tau}$ represents the value of the variable x at step $(t - \tau)$. Depending on the values of the parameters, this equation displays a range of periodic and chaotic dynamics.

The parameters to generate the time-series in this chapter are pre-set to the same parameters as in other publications (Angelov 2004*b*, Angelov & Zhou 2006, Vrbanek & Wang 2009): $\beta = 0.2$, $\gamma = 0.1$, $\tau = 17$, $n = 10$ with the initial value of $x_0 = 1.2$ (Eq. 8.2).

$$\frac{dx}{dt} = 0.2 \frac{x_{t-17}}{1 + {x_{t-17}}^{10}} - 0.1 x_t \tag{8.2}$$

In this benchmark the value of the time series 85 steps ahead (x_{t+85}) will be

predicted based on the values at the current moment, 6, 12 and 18 steps back. Those values and the parameters mentioned above were used to allow consistency and ability to compare the results from this thesis to the results reported in other research so far, as the same values were picked when this time series was used previously (Angelov et al. 2004).

The data vector is represented as Eq. 8.3.

$$y = x_{t+85}, \quad X = [x_{t-18} \quad x_{t-12} \quad x_{t-6} \quad x_t]^T \tag{8.3}$$

The training data consists of 3000 data samples and the validation (testing) dataset consists of 500 previously unseen data samples. The integration step (resolution) has been chosen as 0.1. The resulting time series plot can be seen below, with the training dataset in blue and testing period marked in red (Fig. 8.1).

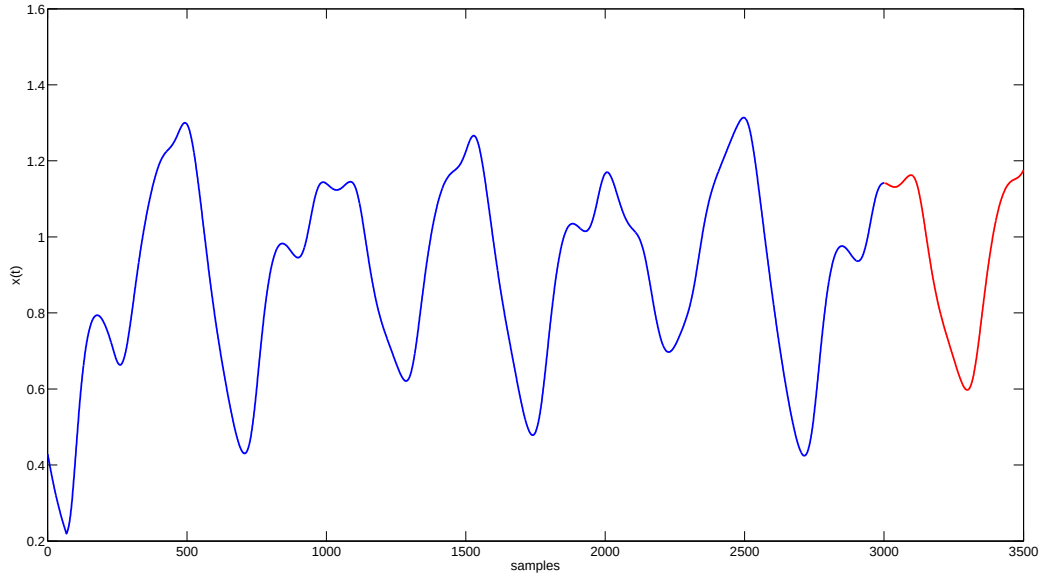

Figure 8.1: Mackey-Glass time series plot.

8.3 Accuracy results on Mackey-Glass time series

8.3.1 Introduction

In this section the results on the Mackey-Glass time series described at the beginning of this chapter will be presented. The aim of this exercise is to investigate the influence of various parameters of the algorithms and also to compare the results obtained from the application of the Mod eTS algorithm to the results from other fuzzy forecasting methods, mainly c-means fuzzy forecasting algorithm, eTS and eTS+. For each error measure, the set of parameters that resulted in achieving the lowest error will be presented. This was achieved by evaluating the combinations of parameters for each algorithm. The lowest training error has been selected for each of the error measures. The parameters which led to obtaining that lowest error for each measure have been recorded and are presented in the Tables with the results, along with the corresponding out of sample error. The mean accuracy and standard deviation of all of the results obtained from all combination of parameters were also calculated, in order to indicate the sensitivity to parameter change. The standard deviation also gives information on the robustness of the methods to mis-specification in the parameters.

8.3.2 Analysis of the results from the application of the Mod eTS forecasting algorithm

In Mod eTS the main configurable parameters are the initial value of radius r and the smoothing parameter γ. The parameters will be chosen through grid search, using following values (Table 8.1):

Table 8.1: Parameters used in the assessment of the Mod eTS forecasting algorithm.

r	0.15 to 1 with 0.05 step
γ	0.15 to 0.95 with 0.05 step

This gives 306 combinations of parameters. For each combination, 4 error measures are calculated for in sample (training) and out of sample (testing) data. Additionally, as the clusters are not set in advance, their number is calculated for each error metric. The results are summarised in Table 8.2. The table has three sections. The top section presents the results and parameters that were obtained to achieve the lowest MASE error for the training data. Both training and testing errors are presented. The middle section presents the results and parameters that were obtained to achieve the lowest testing error. And finally, the bottom section of the table shows average values across all combination of parameters.

Table 8.2: Accuracy results of the application of the Mod eTS algorithm to the Mackey-Glass time series forecasting problem.

	MASE	**MAPE**	**RMSE**	**NDEI**
r **train.**	0.25	0.25	0.25	0.25
γ **train.**	0.15	0.15	0.15	0.15
c **train.**	18	18	18	18
train. error	0.367	9.427	0.102	0.403
test. error	0.542	14.473	0.141	0.695
r **test.**	0.25	0.15	0.25	0.25
γ **test.**	0.50	0.70	0.35	0.35
c **test.**	17	24	18	18
test. error	0.536	14.111	0.141	0.692
mean train.	0.420	11.013	0.116	0.458
std. train.	0.0030	0.0912	0.0007	0.0027
mean test.	0.608	16.443	0.167	0.820
std. test.	0.0025	0.1002	0.0013	0.0063

The best results for training data are obtained for lower values of both the initial r ($r = 0.25$) and the update rate γ ($\gamma = 0.15$). Low values of initial r limit the size of the clusters which results in a higher number of the clusters required to cover the data space (18 in this case). The results obtained when parameters are determined based on the lowest testing error are not aligned with the results obtained when

parameters are determined based on the lowest error on training dataset. This is a result of much shorter testing period (500 samples), however the difference is not substantial (MASE of 0.542 for training and 0.536 for testing). The initial value of radius r stays consistent ($r = 0.25$), however the γ is higher ($\gamma = 0.5$) which results in a smaller number of clusters (17). The standard deviation of the obtained results is also low, which suggests consistent performance regardless of the set of used parameters.

Fig. 8.2 and 8.3 show the distribution of training and testing error. The colours represent the scale of the error, with dark blue being the lowest error and dark red being the highest. It is obvious that the best results are obtained for low values (0.15 - 0.25) of r and low values (up to 0.5) of γ. This has an impact on the number of clusters (Fig. 8.4), which increase with the smaller value of r (again, red colour represents the highest number of clusters with blue being the lowest).

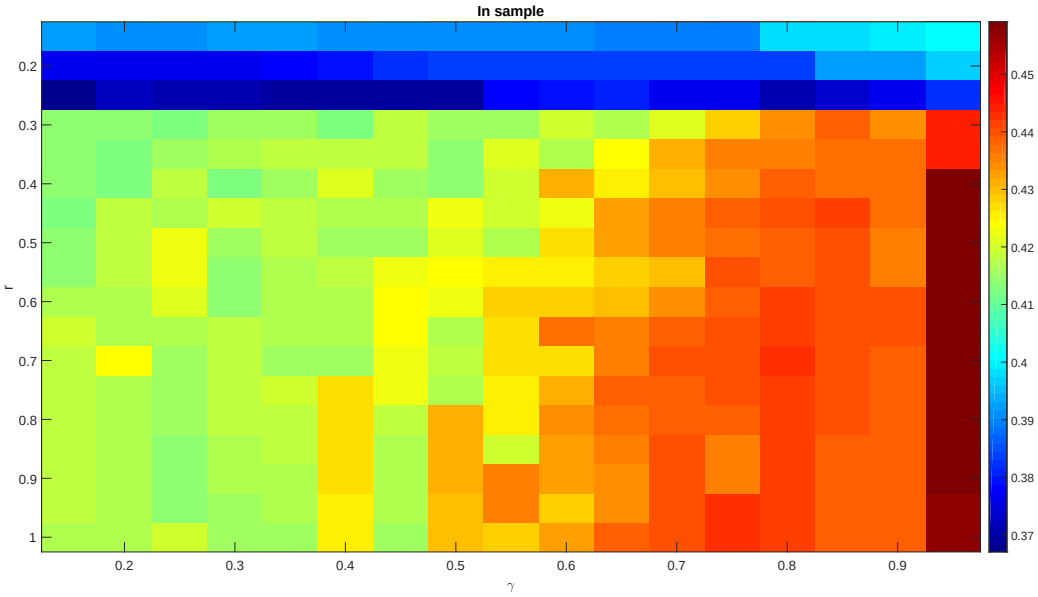

Figure 8.2: Training MASE distribution of Mod eTS forecasting algorithm applied to a Mackey-Glass time series forecasting problem.

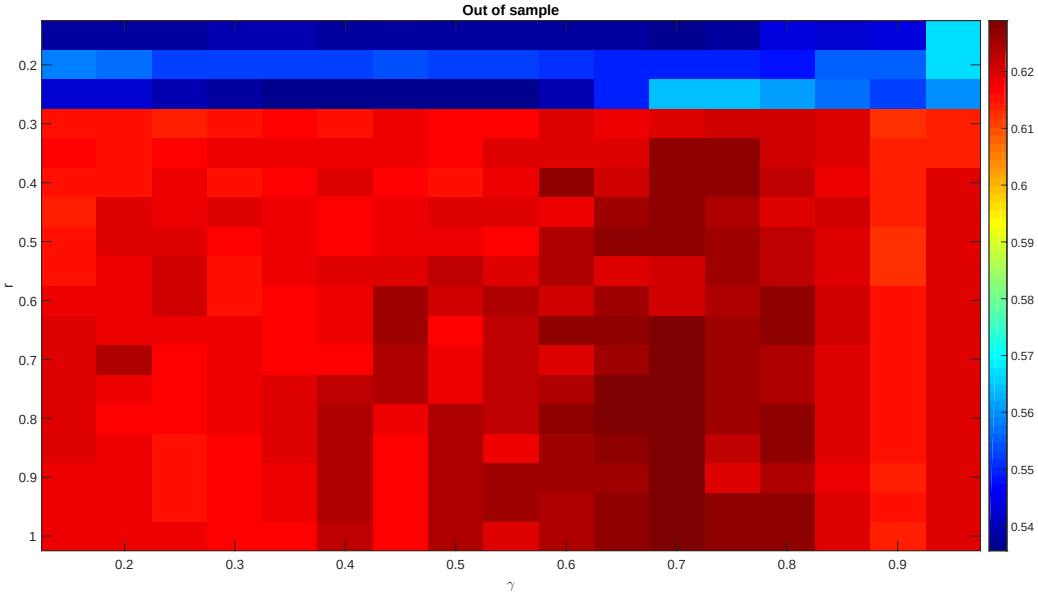

Figure 8.3: Testing MASE distribution of Mod eTS forecasting algorithm applied to a Mackey-Glass time series forecasting problem.

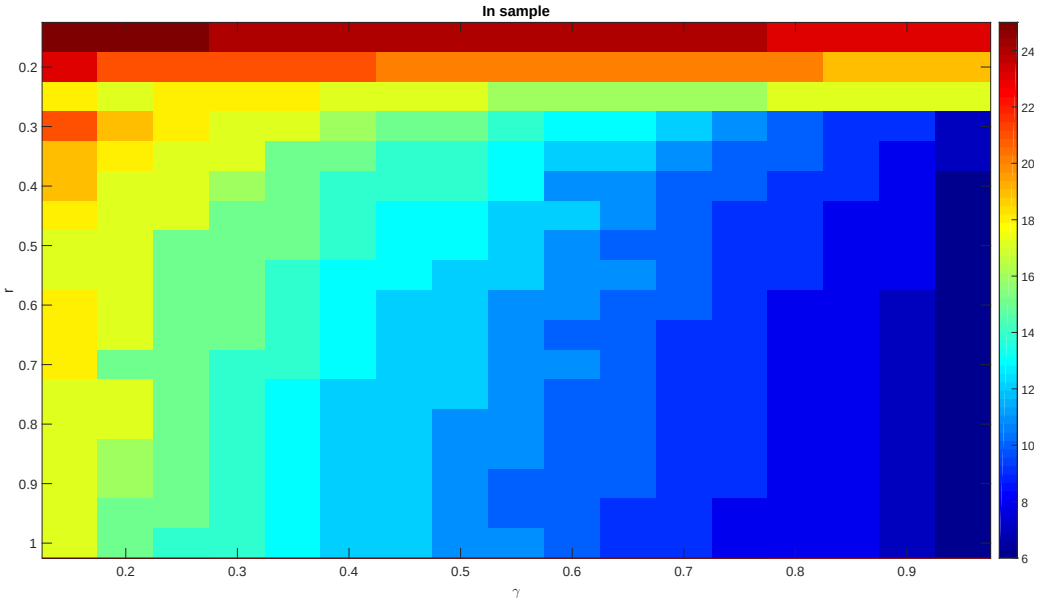

Figure 8.4: Final number of clusters for different values of r and γ of Mod eTS forecasting algorithm applied to a Mackey-Glass time series forecasting problem.

8.3.3 Comparison with other fuzzy clustering methods

In this section the results obtained from Mod eTS methods will be compared with other fuzzy evolving methods. The methods chosen are the same as the ones selected in the previous chapter: c-means forecasting algorithm, eTS and eTS+. The choice has been dictated by the similarity to the Mod eTS method, which is important as it gives the idea if the modification to the algorithm is effective.

8.3.3.1 Analysis of the results from the application of the c-means forecasting algorithm

C-means algorithm needs to be configured before the first run by setting the value of m (fuziness factor) and the number of clusters c the data is going to be split into. The parameters are selected using the grid search over the following combination of parameters (Table 8.3):

Table 8.3: Parameters used in the assessment of the c-means forecasting algorithm.

m	1.1 to 2 with 0.1 step	2.2 to 3 with 0.2 step	3.5 to 10 with 0.5 step
c	2 to 10 with 1 step	15 to 25 with 5 step	50, 75 and 100

which results in 29 values of m and 15 values of number of clusters c.

The accuracy is reported using the 4 error measures used in the previous chapter (*MASE*, *MAPE*, *RMSE* and *NDEI*) across all combinations of the parameters. The results are summarised in Table 8.4. This includes best training and testing results (together with corresponding combination of parameters), mean and standard deviation of accuracy.

Table 8.4: Accuracy results of the application of the c-means algorithm to the Mackey-Glass time series forecasting problem.

	MASE	**MAPE**	**RMSE**	**NDEI**
m **train.**	1.7	1.7	1.9	1.9
c **train.**	100	100	100	100
train. error	0.384	11.255	0.112	0.441
test. error	0.550	12.901	0.142	0.700
m **test.**	1.9	2	2	2
c **test.**	75	100	50	50
test. error	0.452	10.225	0.125	0.613
mean training	0.840	25.842	0.219	0.861
std. training	0.089	2.725	0.021	0.081
mean testing	0.783	18.616	0.192	0.942
std. testing	0.054	1.271	0.011	0.053

It can be seen that there is a slight difference as to which parameters should be chosen in order to obtain the lowest error. This is due to the fact that those error measures are calculated in a slightly different way. There is a slight difference of parameter m: 1.7 for *MAPE* and *MASE*, and 1.9 for *RMSE* and *NDEI*. The number of pre-set clusters is the same for all error measures ($c = 100$). However, selecting such a high number of clusters has an impact on the testing data accuracy, as the best results for that metric have been obtained using lower cluster counts for *MASE*, *RMSE* and *NDEI*: $c = 50 - 75$.

To investigate the impact of change in those parameters, Fig. 8.5 and 8.6 presents the distribution of errors across the whole range of the parameter grid for both training and testing data.

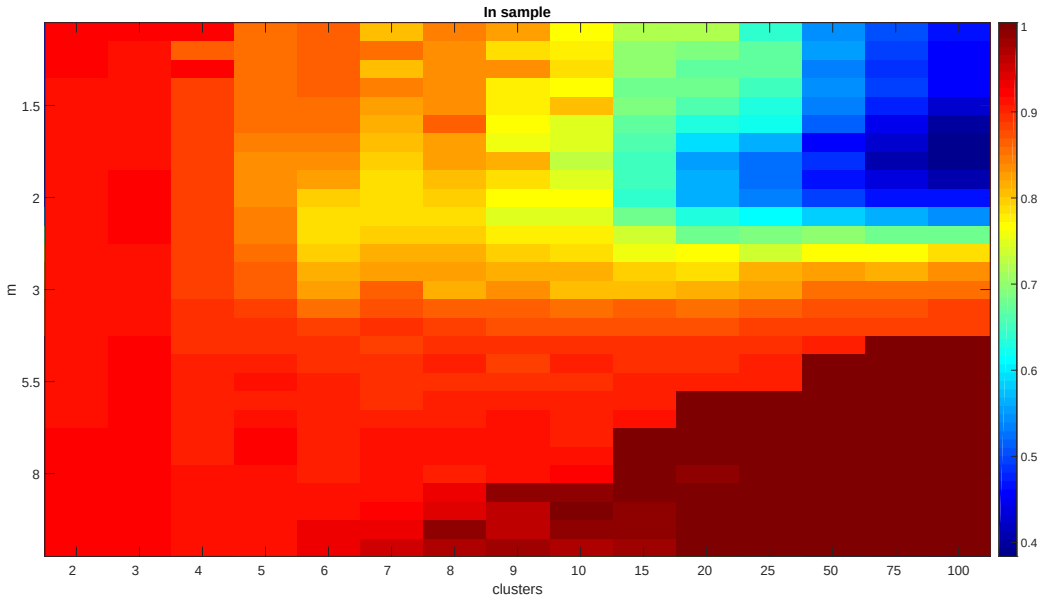

Figure 8.5: Training MASE distribution of c-means forecasting algorithm applied to a Mackey-Glass time series forecasting problem.

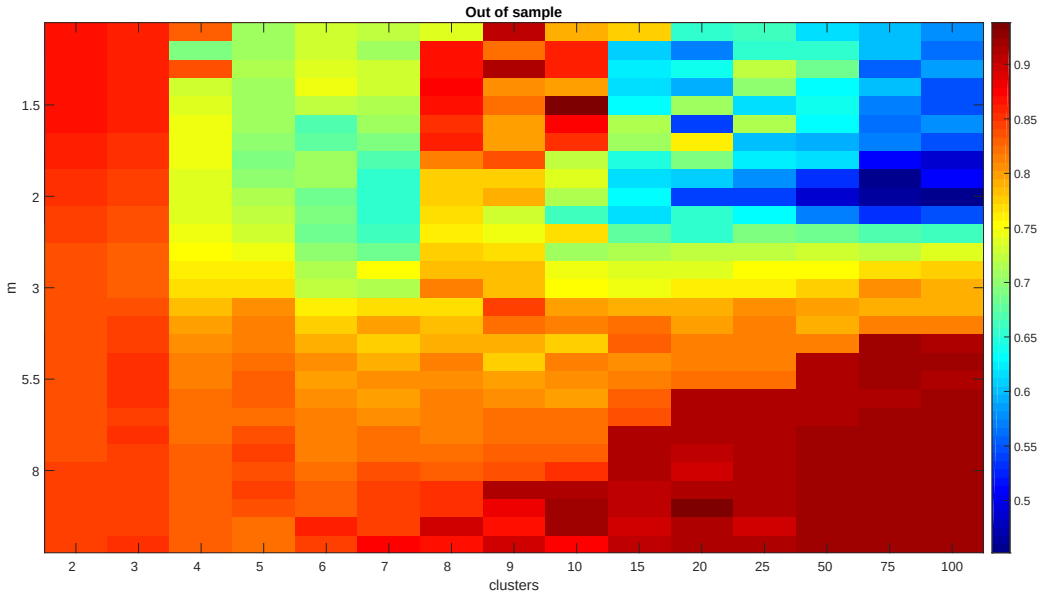

Figure 8.6: Testing MASE distribution of c-means forecasting algorithm applied to a Mackey-Glass time series forecasting problem.

The best results are obtained for small values of m (between 1.5 - 2.5) and higher values of c (above 30). Analysing further, the number of clusters seems to have a smaller impact on the accuracy, as it can be seen that for values of m higher than 3, the performance is consistently low, even with an increase in number of clusters. The lowest accuracy is observed for high values of m and high number of clusters c. The findings are consistent with other research (Chiu 1994), where the value of $m = 2$ has been suggested. The number of clusters are more problem dependent and will vary with the amount of available data and its structure. In this case setting the number of clusters above 40 yields good results. One has to be careful not to set the number too high as the resulting structure can cause over-fitting and will not be flexible enough to accommodate new, unseen data.

8.3.3.2 Analysis of the results from the application of the eTS algorithm

For eTS, the radius r has to be pre-set before the algorithm is initiated. The investigation included a range of r between 0.1 to 1. The results are summarised in Table 8.5:

Table 8.5: Accuracy results of the application of the eTS algorithm to the Mackey-Glass time series forecasting problem.

	MASE	**MAPE**	**RMSE**	**NDEI**
r **train.**	1	1	1	1
c **train.**	12	12	12	12
train. error	0.356	9.392	0.099	0.391
test. error	0.721	18.925	0.187	0.919
r **test.**	0.10	0.10	0.10	0.10
c **test.**	45	45	45	45
test. error	0.485	11.344	0.136	0.668
mean training	0.380	10.076	0.106	0.417
std. training	0.011	0.397	0.003	0.013
mean testing	0.588	15.566	0.158	0.775
std. testing	0.068	2.021	0.016	0.078

The best results have been obtained by setting the value of r to 1, which resulted in 12 clusters. This is however not consistent with the results obtained from the testing data, where the value of 0.1 gave the best results, but generated high

number of clusters (45). Fig. 8.7 shows the distribution of the MASE errors for both training and testing, as well as the final number of clusters based on the chosen values of r. It can be seen that, as expected, the number of clusters increases with the decrease in the chosen radius value.

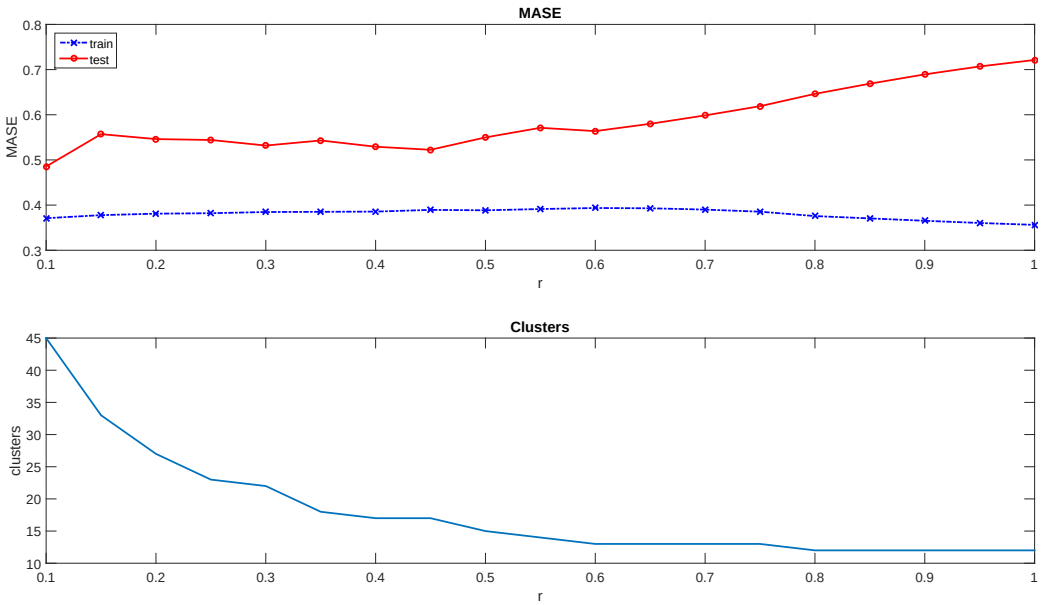

Figure 8.7: MASE distribution and final number of clusters of eTS forecasting algorithm for different values of r.

8.3.3.3 Analysis of the results from the application of the eTS+ algorithm

In case of eTS+, similarly to Mod eTS both initial radius r and the value of γ needed to be pre set. The range of the parameters is the same as in Mod eTS (Table. 8.1). The results for all of the error measures can be seen in Table 8.6.

Table 8.6: Accuracy results of the application of the eTS+ algorithm to the Mackey-Glass time series forecasting problem.

	MASE	**MAPE**	**RMSE**	**NDEI**
r **train.**	0.25	0.25	0.25	0.25
γ **train.**	0.9	0.9	0.9	0.9
c **train.**	1	1	1	1
train. error	0.525	13.323	0.135	0.531
test. error	0.368	8.944	0.099	0.486
r **test.**	0.25	0.25	0.25	0.25
γ **test.**	0.9	0.9	0.9	0.9
c **test.**	1	1	1	1
test. error	0.368	8.944	0.099	0.486
mean training	0.862	22.818	0.249	0.983
std. training	0.112	3.140	0.038	0.150
mean testing	1.477	35.742	0.354	1.739
std. testing	0.370	8.946	0.085	0.417

In eTS+ the best results are obtained by pre setting the r to a low value of 0.25 and γ set to high 0.9. This is consistent for both training and testing data sets. For both settings the final number of clusters is equal to 1 due to strict cluster removal and modification rules in eTS+. The mean values are not as good as in the case of other algorithms, and standard deviation is particularly high, which can indicate a big discrepancy between the accuracy for different values of parameters. This

can be seen on Fig. 8.8, where there is a big difference between the values from $\gamma = 0.5$.

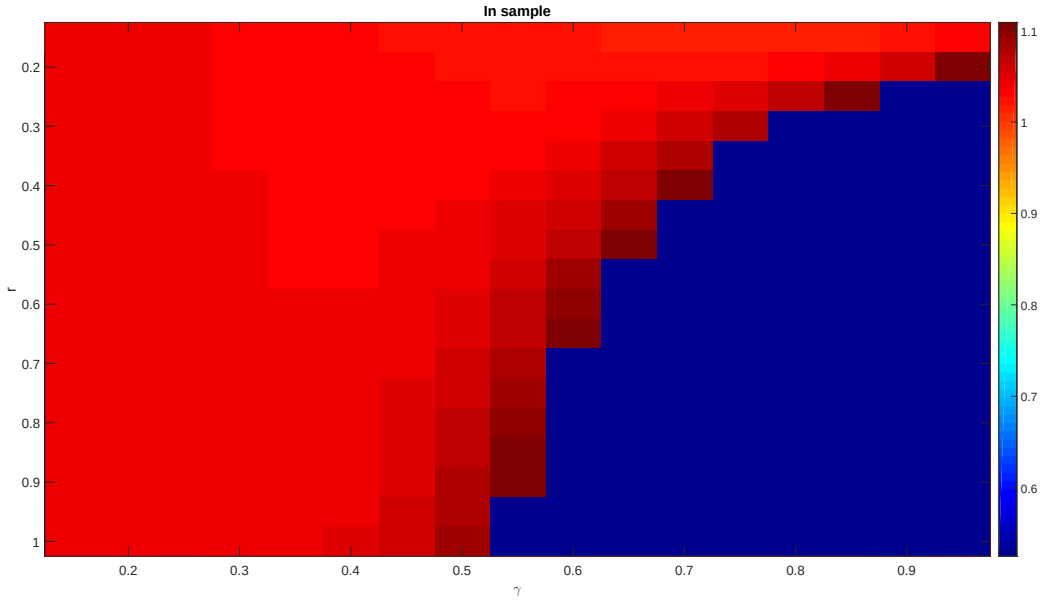

Figure 8.8: Training MASE distribution of eTS+ forecasting algorithm applied to a Mackey-Glass time series forecasting problem.

8.3.3.4 Comparison of the results from Mod eTS and other fuzzy forecasting algorithms

The resulting MASE from all of the algorithms are summarised in Table 8.7.

Table 8.7: Comparison of MASE for all considered algorithms applied to Mackey-Glass time series forecasting problem.

	c-means	**Mod eTS**	**eTS**	**eTS+**
r **train.**	-	0.25	1	0.25
γ **train.**	-	0.15	-	0.9
c **train.**	100	18	12	**1**
train. error	0.384	0.367	**0.356**	0.525
test. error	0.550	0.542	0.721	0.368
r **test.**	-	0.25	0.1	0.25
γ **test.**	-	0.5	-	0.9
c **test.**	75	17	45	**1**
test. error	0.452	0.536	0.485	**0.368**
mean training	0.840	0.420	**0.380**	0.862
std. training	0.089	**0.003**	0.011	0.112
mean testing	0.783	0.608	**0.588**	1.477
std. testing	0.054	**0.003**	0.068	0.370

For the MASE assessed over training period the best results have been achieved by the eTS algorithm, followed by Mod eTS and c-means, with eTS+ having the poorest performance. eTS+, however, had the lowest error on the testing period. Overall, Mod eTS shows promising results here, as it balances high training and testing accuracy with low number of clusters. c-means algorithm has high accuracy on the training period, but requires high number of clusters in order to achieve it. eTS+ has the lowest error for the testing period and achieves it only with 1

cluster, but has the worst performance over the training period. eTS has the best training accuracy, but high MASE in the testing period.

When averaging over all of the results, Mod eTS has the lowest values of standard deviation and satisfactory values of mean accuracy. This can suggest that this method is much more resistant to mis-specification of the input parameters for this particular problem, however the parameter tuning still has to be carried out on an initial data-set in order to ensure best possible accuracy. The eTS algorithm has the highest accuracy for both mean training MASE and the testing MASE.

8.4 Conclusions

In this chapter Mod eTS algorithm has been assessed in detail on the common benchmarking problem of predicting future values of the Mackey-Glass time series using 4 different error metrics. The performance has been compared with c-means, eTS and eTS+ fuzzy evolving algorithms. Although the modification introduced in the Mod eTS algorithm was inspired by the problem of leakage forecasting, it is important to assess its performance on a common benchmarking problem. The results showed a promise, as the algorithm was well balanced in terms of the number of clusters and the accuracy over both training and testing period.

When comparing the results from leakage forecasting and from the application to the Mackey-Glass time-series, it can be seen that the optimal parameters used to achieve the best accuracy were different for both problems. For leakage forecasting, the highest performance was achieved for high numbers of both r and γ ($r > 0.8$ and $\gamma > 0.6$), whereas for Mackey-Glass time-series the highest performance was achieved for low values of r and γ ($r < 0.25$ and $\gamma < 0.5$). This proves that the algorithm is flexible, but the parameters need to be carefully tuned based on the problem and the characteristic of data to which the algorithm is applied.

In terms of clusters, the Mackey-Glass time-series has considerably more samples than the leakage forecasting problem, which results in higher number of rules created to cover the whole data space, which is also a result of smaller optimal radii r and radii update rate γ.

The $MASE$ results for the Mackey-Glass time-series are much lower which indicate that it was much easier to beat the Naive prediction for this problem than in the case of leakage forecasting. However when looking at the $MAPE$, the error is much lower in case of leakage forecasting (average of 5.019), which can indicate that the method is more suitable for problems with seasonal time series with supporting explanatory variables.

It can also be seen that, in general, the training $MASE$ is lower than the testing $MASE$. This is due to the fact that the parameters are optimised based on the error calculated over the training period and the cluster structure is also evolving during the training. This will mean that the error calculated over the testing period will almost always be higher, as the algorithm is applied to previously unseen data.

In the next chapter the conclusions and summary from the work outlined in this thesis will be presented. The directions and suggestions on what can be improved in the method and what additional research can be carried out in this area will also be depicted.

Chapter 9

Conclusions and Further Work

9.1 Conclusions

In this thesis developing and applying the evolving fuzzy system to forecasting problems have been explored. Evolving fuzzy systems overcome the drawback of the off-line fuzzy identification methods where in order to make a prediction all data needs to be available for the model to be created. Therefore, when the off-line identification methods are applied, the conditions under which the system operates need to be constant for the generated model to keep its validity. If the conditions change, and the behaviour has not been captured by the data before, the whole model needs to be generated from scratch.

The review of the fuzzy evolving methods showed promising results when applied to problems described with non-linear relationships. However, those methods have often been tested on the data generated by physical systems where the relations between the input variables have been well understood. There seems to be a lack of evaluation of those methods on forecasting problems, when more complex, or unknown, relationships exist between the problem variables. The choice of the

parameters of the algorithms are also not well understood as they have been based on the evaluation on the same group of problems.

In order to evaluate the idea of fuzzy evolving methods on forecasting problems, two different sets of data were used: a use case of leakage forecasting and the artificially generated periodic Mackey-Glass time series.

Existing leakage detection methods have been presented and the concept of a leakage forecasting process has been introduced. In leakage forecasting the forecasts are made for a long period ahead, such as 12 months, and for a number of regions of operations, each having its own, unique structure (type of pipes, soil, weather conditions, urban or rural environments, etc.). This increases the complexity of the problem, as it becomes difficult to establish the general influence and relationships between various explanatory variables on the leakage. Those variables were thoroughly investigated, and after the data analysis and the conversations with the company experts, it was determined that a number of factors have an influence on the leakage. First factor was Natural Rate of Rise (NRR), which relates to the underlying rate at which leakage increases within a network in the absence of any leak repairs. The value of NRR depends on the type of pipes used in the network but also on the month of the year and forecasted weather conditions. The second and third variable were Equivalent Service Pipe Bursts (ESPBs), which are split into two parts: detected and reported. They represent the resource effort from both an active search for leaks (detected) and when the engineering team responds to customers (reported) and depend on the expected leakage, the structure of the network and the amount of resources that are available for each region. The last factor was the previous value of leakage, as very often the effects of leaks carry over to the next periods.

The number of factors included meant that the forecasting model had to be tailored to the area of operation that it was applied to. It was discovered that

the relationship between the explanatory variables and the impact on leakage in different areas of operation was not fully understood, the ability to predict the leakage based on historical data was not measured and that months with highest leakage values also have the highest impact on the total leakage accumulated throughout the year. The necessity for the automatic way of building the forecasting model based on the historical data arose, due to the need of forecasting for different areas of operation and the existing non-linear relationships of leakage and other factors. The research focused on using evolving fuzzy systems. This approach combined the ability to express non-linear relations among variables through the use of fuzzy sets, with the automatic rule creation and adaptation of the rule structure to changing conditions based on the historical data.

A novel method (Mod eTS) based on fuzzy evolving algorithms was developed in order to increase the accuracy of prediction of the leakage forecasts, especially for periods and areas with a high level of increase in leakage. The algorithm incorporates a dynamic radius adjustment of each input variable of each cluster. Introducing a radius for each cluster dimension which is dynamically changing allows for better coverage of the data by clusters, as radii may differ in different dimensions. It also limits the number of clusters (and consecutively rules) as it is not necessary to create more clusters when data is already well covered by the existing ones.

The novel algorithm was then applied to leakage data set and the data generated by the Mackey-Glass process and the results from both were compared with a number of fuzzy and statistical forecasting methods. In real world applications, carefully choosing the inputs and the explanatory variables can considerably increase the accuracy of the forecast. Therefore a considerable amount of work had to be carried out in order to pre-process data to be suitable for use in the algorithms.

Results obtained by applying the Mod eTS algorithm to real-world leakage

data indicated that the proposed method generally performed better than other methods for testing (out-of-sample) data. This proves that the model built on the training data set can be reliably applied to the unseen testing data, and the confidence that this model will produce accurate forecasts is higher as compared to other methods for this application. The Mod eTS had lower values of error for testing period but generated more rules when applied to the leakage forecasting problem as compared to other fuzzy algorithms. This was caused by the lack of more complex rule management in Mod eTS.

The Mackey-Glass time-series had considerably more samples than the leakage data set, which resulted in even higher number of rules created to cover the whole data space. The method is sensitive to the choice of the configurable parameters and it was seen that the optimal parameters used to achieve the best accuracy were different for both problems. This proved that the algorithm is flexible, but the parameters need to be carefully tuned based on the problem to which the algorithm is applied.

The $MASE$ results for the Mackey-Glass time-series were much lower than those obtained from the application to the leakage problem, which indicated that it was much easier to beat the Naive prediction. However when looking at the $MAPE$, the error was much lower in case of leakage forecasting. Conflicting results like this occur often when various methods and data sets are used, as it is unlikely that one method will be the best for all applications. Hyndman & Athanasopoulos (2014) wrote that *"what is required from a forecasting method are consistently sensible forecasts, and these should be frequently evaluated against the task at hand"*.

The fact that the method performed well on both datasets is a good start, however more research needs to be carried out in order to generalize that statement for Mod eTS.

9.2 Future work

While working on this thesis a number of possible areas for further research were identified, which could potentially be used to further expand on the method developed during this study.

9.2.1 Use of linguistic variables to describe the fuzzy rules automatically generated through clustering

Forecasting support systems are used to aid the forecasters in choosing the right method to generate the best forecasts. Very often however, the experts have no knowledge on how the particular method works and they cannot understand the generated forecasting model. The possibility of using linguistic variables to interpret the fuzzy rules could mean that the fuzzy model may be easier to understand for the experts, and therefore, could be used more effectively, even by experts not familiar with the theoretical concepts of the evolving fuzzy methods.

9.2.2 Forecast adjustments

One of the interesting extensions of algorithm could be the incorporation of the automatic forecast adjustments. It can be interesting from both research and business perspectives, as in the case of leakage forecasting, the company was required to produce the forecast in accordance to their average leakage targets. This means that the outcome of the forecast could be influenced by varying the predictor variables, so that the average leakage target is met. The embedding of the forecasting adjustments into the method can be carried out as a future work. One possible extension could be the use of the judgemental adjustments that could be automatically included in the forecast.

9.2.3 Better handling of cluster creation

In terms of the improvements of the algorithm, more research should be carried out on how and when the new clusters are created. The eTS+ algorithm incorporates several heuristics which aim at decreasing the number of clusters, which do not always improve the accuracy of the prediction. More research needs to be carried out in this area, as smaller number of clusters improve the computational speed and simplify the model structure, making it easier to understand.

9.2.4 Evaluation on more datasets

Finally, the evaluation of the algorithm should be carried out on more real-world forecasting problems. Even though the algorithm has been evaluated on several leakage datasets and one benchmarking dataset, it is necessary to include more data in order to generalize the conclusions. The research included in this thesis laid the foundations for this work to be carried out.

Appendices

Appendix A

Equations and theory

A.1 Exponential smoothing

There are several forms of exponential smoothing which are applicable to time-series with different characteristics.

Simple exponential smoothing can be effectively used only on time-series without trend or seasonality (Eq. A.1):

$$\begin{aligned} F_{t+1} &= l_t \\ l_t &= \alpha y_t + (1-\alpha) l_{t-1} \end{aligned} \tag{A.1}$$

where F_{t+1} is the forecast at time $t+1$, l_t is level (smoothed value) of previous observations, $0 < \alpha \leqslant 1$ is a smoothing parameter (weight of the previous observations) and y_t is the actual observation at time t.

The simple exponential smoothing was extended by Holt (2004) by adding the

trend component (and allowing to forecast data with trend):

$$\begin{aligned} F_{t+h} &= l_t + hb_t \\ l_t &= \alpha y_t + (1-\alpha)(l_{t-1} + b_{t-1}) \\ b_t &= \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1} \end{aligned} \tag{A.2}$$

where b_t is an estimate of the trend of the series at time t and $0 \leqslant \beta \leqslant 1$ is the smoothing parameter for the trend. h is a linear parameter which is used for h-step ahead forecast (as forecast will no longer be flat). A number of modifications to Eq. A.2 have also been introduced to accommodate time-series with various trend patterns, such as Exponential, Additive Damped and Multiplicative Damped trend.

Winters (1960) further extended the method by adding the seasonal component. The forecast was now obtained by calculating 3 components: level l_t, trend b_t and seasonality s_t (Eq. A.3):

$$\begin{aligned} F_{t+h} &= l_t + hb_t + s_{t-m+h} \\ l_t &= \alpha(y_t - s_{t-m}) + (1-\alpha)(l_{t-1} + b_{t-1}) \\ b_t &= \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1} \\ s_t &= \gamma(y_t - l_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m} \end{aligned} \tag{A.3}$$

where m is seasonality factor (for example, $m = 12$ for yearly data), s_t is smoothed seasonality and γ is the seasonality smoothing factor. Similarly to Holt's trend exponential smoothing, seasonal smoothing also includes variants to handle additive, multiplicative and damped seasonality. This can give an overall 15 exponential methods, when considering all combinations (Taylor 2003).

A.2 ARIMA models

Autoregressive (AR) models are models that estimate the next observation as a function of previous observations. The number of p previous observations define the class of AR(p) model:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + e_t \tag{A.4}$$

where c is a constant, $\phi_1, \ldots, \phi_p$ are parameters defining the time-series pattern, e_t is an error term and y_{t-p} is the p lagged value.

Instead of lagged past observations, q lagged forecasting errors can be used. This model is called Moving Average - MA(q):

$$y_t = c + e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \ldots + \theta_q e_{t-q} \tag{A.5}$$

When both AR and MA models are combined, the ARIMA model is obtained. ARIMA stands for Auto Regressive Integrated Moving Average:

$$y'_t = c + \phi_1 y'_{t-1} + \ldots + \phi_p y'_{t-p} + \theta_1 e_{t-1} + \cdots + \theta_q e_{t-q} + e_t \tag{A.6}$$

The y'_t term is differenced, in order to obtain the non-differenced term it has to be integrated (hence the Integrated term in the ARIMA). ARIMA(p,d,q), like AR and MA models, is also defined by different orders of its parts, where p is the order of the AR, d is the degree of differencing and q is the order of the MA part. ARIMA models can be used to model non-seasonal data. In order to model seasonal data, additional seasonal terms (P,D,Q) are needed.

A.3 Fuzzy Sets

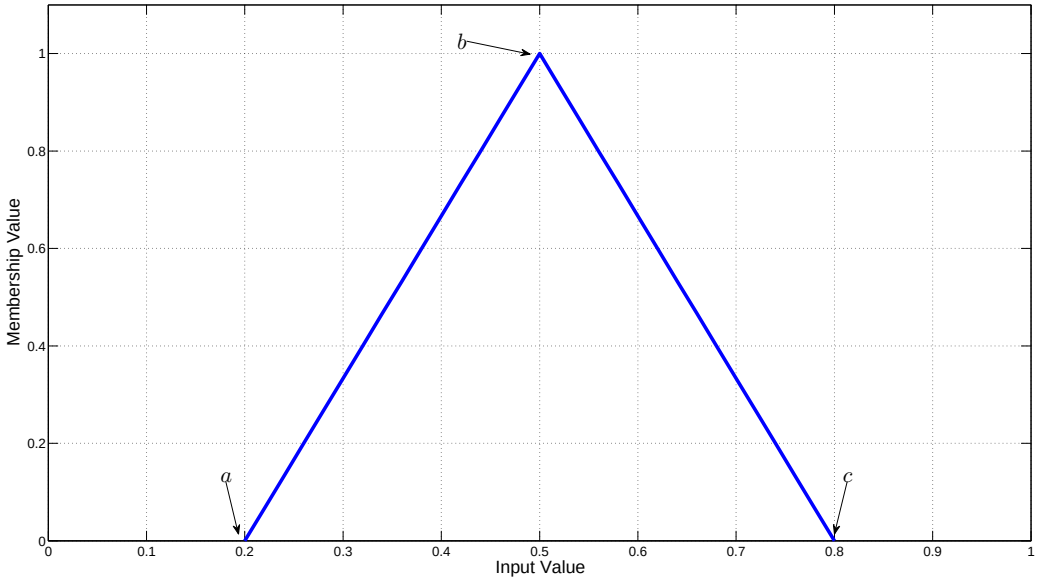

Figure A.1: Triangular fuzzy set.

The triangular fuzzy set (Fig. A.1) is characterized by three points: a and b are the leftmost and rightmost points on the triangle and c is its centre. The membership value of an input x belonging to a triangular fuzzy set characterized by a, b and c points is calculated using following set of equations:

$$\mu(x) = f(x, a, b, c) = \begin{cases} \dfrac{x-a}{b-a} & : a < x \leq b \\ \dfrac{c-x}{c-b} & : b < x < c \\ 0 & : otherwise \end{cases} \tag{A.7}$$

If the value of $x = 0.4$ the membership degree of the triangular fuzzy set $\mu(x) = 0.67$.

Another popular type of fuzzy sets are described using trapezoid function (Fig. A.2).

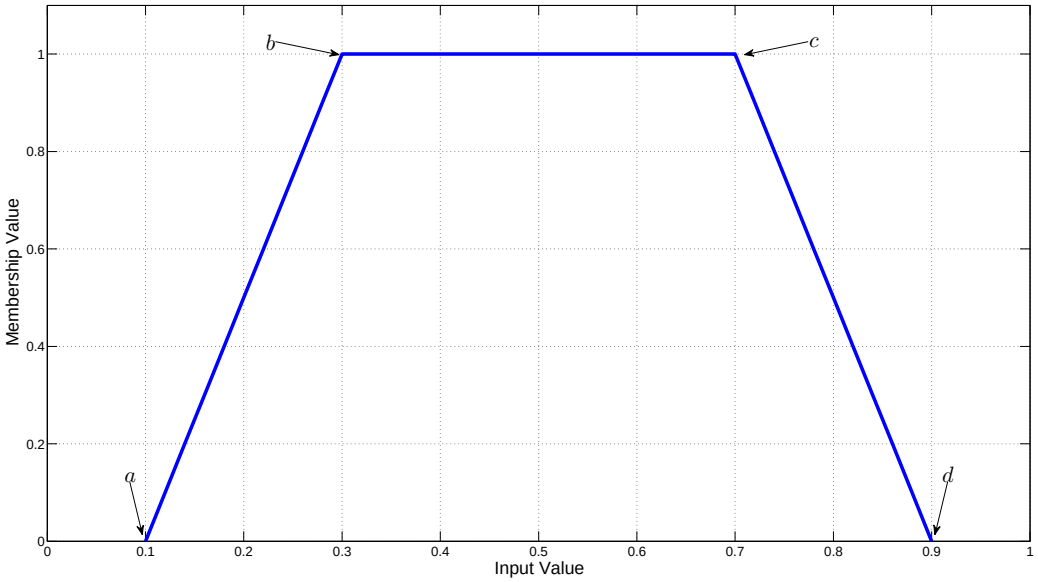

Figure A.2: Trapezoid fuzzy set.

The trapezoid is characterized by four points: a and d being the leftmost and rightmost points of the trapezium and b and c are corners of a shorter base. In principal, the trapezoid fuzzy sets are an extension of triangular sets and are described by following set of equations:

$$\mu(x) = f(x, a, b, c, d) = \begin{cases} \dfrac{x-a}{b-a} & : a < x \leq b \\ 1 & : b < x < c \\ \dfrac{d-x}{d-c} & : c \leq x < d \\ 0 & : otherwise \end{cases} \tag{A.8}$$

Similarly to previous example, if $x = 0.4$ in this case the output of the membership function will equal to $\mu(x) = 1$.

The 3rd most commonly used type of fuzzy set is Gaussian (Fig. A.3).

Figure A.3: Gaussian fuzzy set.

The Gaussian fuzzy set is defined by two values: c - the peak of the Gaussian curve and σ, standard deviation, also called spread, which controls the width of the Gaussian. The fuzzy set is described by the Gaussian equation:

$$\mu(x) = f(x, c, \sigma^2) = e^{-\frac{(x-c)^2}{2\sigma^2}} \tag{A.9}$$

For $x = 0.4$ the membership function equals $\mu(x) = 0.8007$.

A.4 Fuzzy operators

The intersection operator (equivalent to **AND** operation) takes the minimum of the membership functions of two fuzzy sets:

$$x_1 \text{ is } A_1 \textbf{ AND } x_2 \text{ is } A_2 : \min(\mu_1(x_1), \mu_2(x_2)) \tag{A.10}$$

The union operator (equivalent to **OR** operation) takes the maximum of the membership functions:

$$x_1 \text{ is } A_1 \textbf{ OR } x_2 \text{ is } A_2 : \max(\mu_1(x_1), \mu_2(x_2)) \tag{A.11}$$

The complement operator (equivalent to **NOT** operation) of a fuzzy set is defined as a negation of the membership degree:

$$x_1 \text{ is } \textbf{NOT } A_1 : (1 - \mu_1(x_1)) \tag{A.12}$$

A.5 Mamdani inference example

The inputs x_1 (temperature) and x_2 (month of the year) can be expressed using the combination of fuzzy sets presented on Fig. A.4.

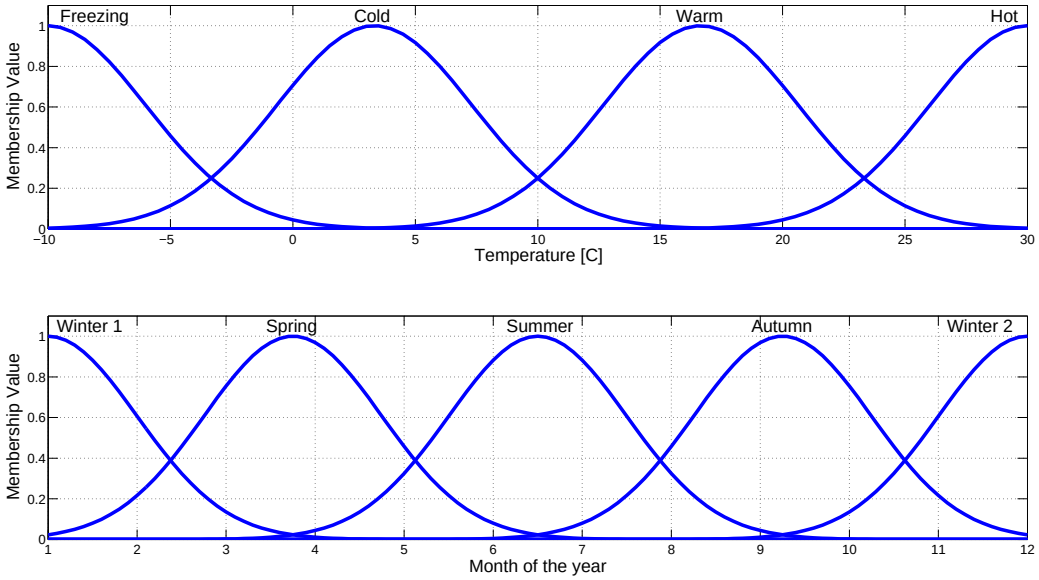

Figure A.4: Fuzzy inputs of leakage detection system.

The output y of the system can be expressed as 3 fuzzy sets describing the number

of engineers (Fig. A.5).

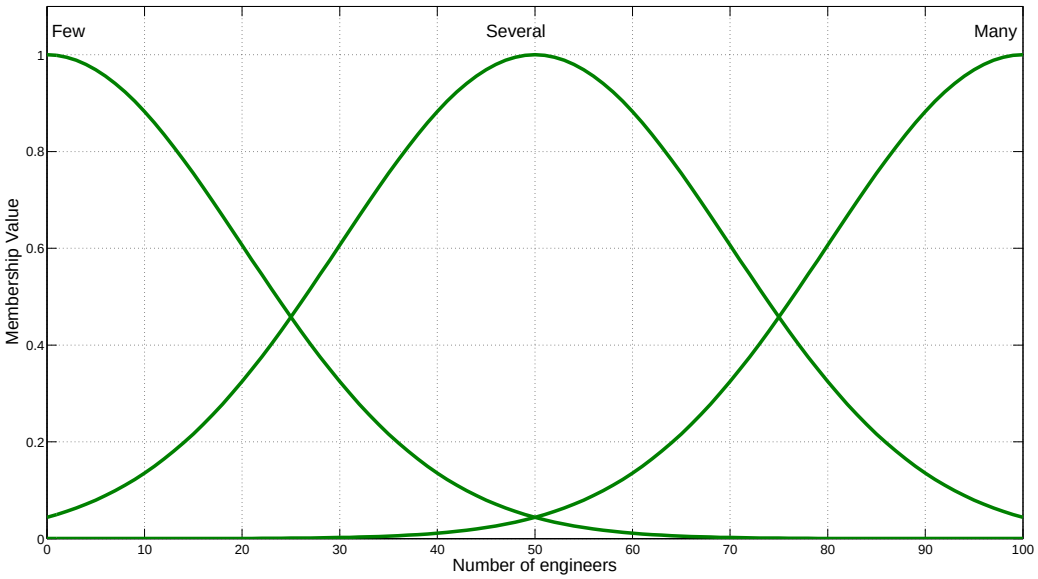

Figure A.5: Fuzzy output of leakage detection system.

The set of rules which define this fuzzy inference system are presented below. For simplification the rules don't cover all possible combinations of inputs:

R1: **IF** x_1 **is** $FREEZING$ **AND** x_2 **is** $WINTER1$ **THEN** $y = MANY$

R2: **IF** x_1 **is** $COLD$ **AND** x_2 **is** $WINTER1$ **THEN** $y = MANY$

R3: **IF** x_1 **is** $WARM$ **AND** x_2 **is** $WINTER1$ **THEN** $y = SEVERAL$

R4: **IF** x_1 **is** $FREEZING$ **AND** x_2 **is** $SPRING$ **THEN** $y = MANY$

R5: **IF** x_1 **is** $COLD$ **AND** x_2 **is** $SPRING$ **THEN** $y = SEVERAL$

R6: **IF** x_1 **is** $WARM$ **AND** x_2 **is** $SPRING$ **THEN** $y = FEW$

R7: **IF** x_1 **is** HOT **AND** x_2 **is** $SPRING$ **THEN** $y = FEW$

(A.13)

To demonstrate how Mamdani inference works in practice, let us assume that the inputs to the fuzzy inference system are the temperature $x_1 = 0$ and the month of February $x_2 = 2$. In order to obtain the output of the systems, all of the rules are evaluated against the inputs (Fig. A.6), meaning that all membership degrees for each rule for both inputs are calculated and then evaluated using the fuzzy operators (**AND** in this case) in order to obtain firing strengths.

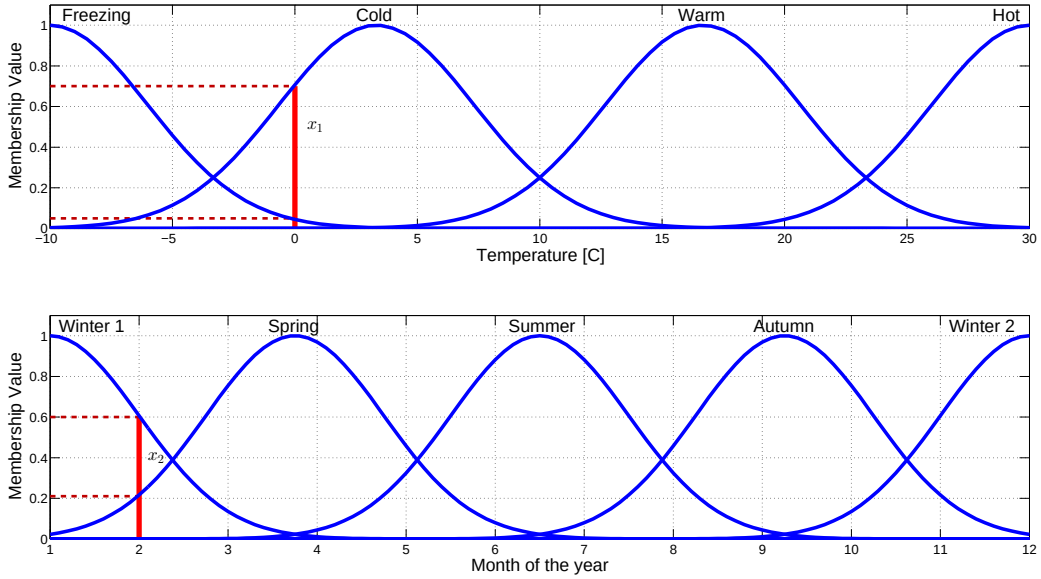

Figure A.6: Evaluating the inputs against the fuzzy rules.

Using the **AND** operator, to calculate the firing strength of each rule, the minimum of both membership functions is calculated. Due to the fact that Gaussian membership functions are used, the firing strength for each rule can be obtained (Eq. A.14).

$$
\begin{aligned}
&\text{R1: } \tau_1 = min(\mu_{11}(x_1) = 0.044, \mu_{21}(x_2) = 0.607) = 0.044 \\
&\text{R2: } \tau_2 = min(\mu_{12}(x_1) = 0.707, \mu_{21}(x_2) = 0.607) = 0.607 \\
&\text{R3: } \tau_3 = min(\mu_{13}(x_1) \approx 0, \mu_{21}(x_2) = 0.607) \approx 0 \\
&\text{R4: } \tau_4 = min(\mu_{11}(x_1) = 0.044, \mu_{22}(x_2) = 0.216) = 0.044 \\
&\text{R5: } \tau_5 = min(\mu_{12}(x_1) = 0.707, \mu_{22}(x_2) = 0.216) = 0.216 \\
&\text{R6: } \tau_6 = min(\mu_{13}(x_1) \approx 0, \mu_{22}(x_2) = 0.216) \approx 0 \\
&\text{R7: } \tau_7 = min(\mu_{14}(x_1) \approx 0, \mu_{22}(x_2) = 0.216) \approx 0
\end{aligned}
\tag{A.14}
$$

where τ_i is the firing strength of each rule R_i.

The next step is to calculate the consequent value based on the firing strength of each rule. This process is illustrated below (Fig. A.7). Rules 6 and 7 were omitted as the firing strength is close to 0:

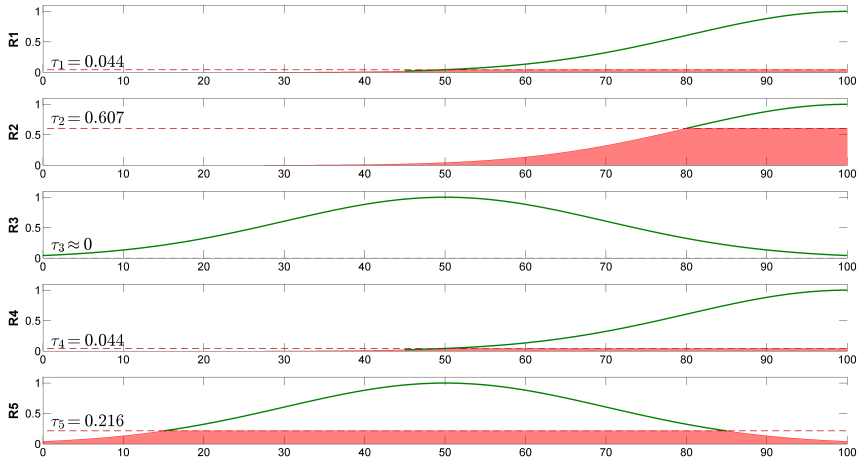

Figure A.7: Obtaining the consequent value of each rule.

The fuzzy set for the output of each rule is cut according to the firing strength calculated from the antecedent part of the rule using fuzzy partitions. The outputs

for each rule are then joined using the maximum criteria for each value of the output, i.e. for each y the maximum value of τ_i from each rule is taken. The result is then combined into an output surface (Fig. A.8).

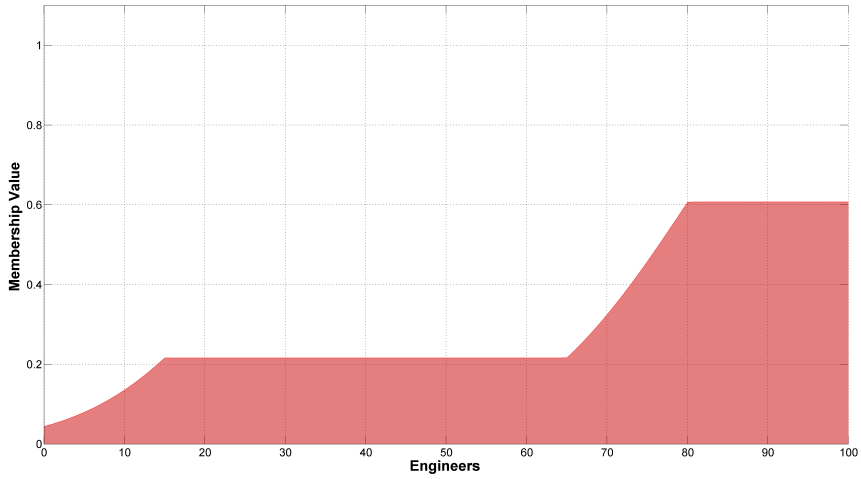

Figure A.8: Combined output surface.

Last step is to defuzzify the surface to obtain numerical output value. This can be achieved using various maximum methods (mean, largest or smallest maximum) or centre of gravity (COG - calculates the point which divides the area into two regions having the same area). The choice of the defuzzification techniques is extremely important because it can alter the result significantly. The choice depends on the problem, but it is generally accepted to use COG method as a starting point. The results of various defuzzification methods can be seen on Fig. A.9.

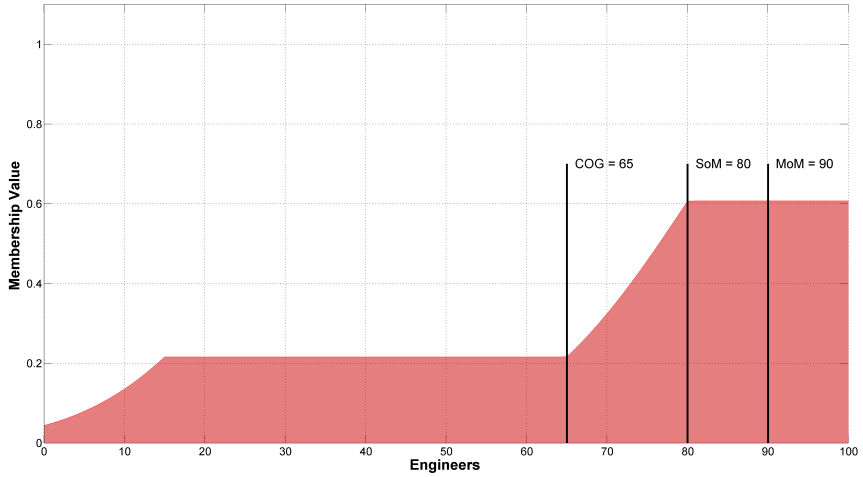

Figure A.9: Defuzzified outputs using severl defuzzification methods; $COG = 65$, Smallest Maximum $SoM = 80$, Mean of Maximum $MoM = 90$.

Appendix B

Data plots and additional results

B.1 Additional tables with results

Table B.1: Comparison of Mod eTS accuracy results with the other fuzzy forecasting methods - parameters determined to optimise the testing MASE

	Mod eTS		**eTS**		**eTS+**		**c-means**		**Sub. clust.**	
	train.	**test.**	**train.**	**test.**	**train.**	**test.**	**train.**	**test.**	**train.**	**test.**
reg_all	1.063	0.500	0.813	1.510	0.547	2.066	0.836	0.995	0.549	0.558
reg_1	1.081	0.627	1.104	0.871	0.852	0.623	1.594	0.610	0.911	0.633
reg_2	0.807	0.499	7.017	17.804	0.838	0.618	1.312	0.507	0.412	0.551
reg_3	1.035	0.826	1.123	1.302	0.724	1.651	0.605	1.155	0.483	0.789
reg_4	0.766	0.733	6.518	3.297	0.804	2.776	0.693	1.777	0.568	1.662
reg_5	0.802	1.145	6.059	6.862	0.839	1.808	1.025	1.050	0.838	1.808
reg_6	0.886	0.469	0.734	0.869	0.833	0.966	0.832	1.293	0.613	0.745
reg_7	1.425	0.486	0.920	1.438	0.694	1.230	0.824	1.598	0.545	0.503
reg_8	0.478	0.852	0.895	2.273	0.772	1.827	0.819	0.876	0.534	1.195
std	0.265	0.229	2.813	5.498	0.099	0.714	0.315	0.419	0.163	0.498
avg	0.927	0.682	2.798	4.025	0.767	1.507	0.949	1.096	0.606	0.938
avg clust.	7.444		3.000		1.667		43.889		2.444	

Table B.2: Comparison of Mod eTS accuracy results with other statistical forecasting methods - parameters determined to achieve lowest testing MASE

	Mod eTS		**Seasonal Naive**		**MLR**		**AR(1)**		**HWM**	
	train.	**test.**	**train.**	**test.**	**train.**	**test.**	**train.**	**test.**	**train.**	**test.**
reg_all	1.063	0.500	0.911	1.693	0.734	1.364	0.853	1.461	1.054	0.530
reg_1	1.081	0.627	1.334	1.647	0.911	0.633	0.865	0.842	1.127	0.886
reg_2	0.807	0.499	0.839	1.895	0.838	0.618	0.889	0.990	1.049	0.718
reg_3	1.035	0.826	0.643	1.622	0.724	1.532	0.849	1.536	1.217	0.437
reg_4	0.766	0.733	1.012	2.431	0.744	2.169	0.877	2.278	1.224	0.427
reg_5	0.802	1.145	1.413	2.739	0.838	1.808	0.908	1.138	0.965	0.636
reg_6	0.886	0.469	1.154	1.643	0.833	0.965	0.946	1.963	0.849	0.550
reg_7	1.425	0.486	1.329	0.964	0.695	1.228	0.922	2.352	0.898	0.305
reg_8	0.478	0.852	0.815	1.599	0.772	1.827	0.845	1.135	1.026	0.819
std	0.265	0.229	0.271	0.515	0.071	0.542	0.035	0.559	0.130	0.192
avg	0.927	0.682	1.050	1.804	0.788	1.349	0.884	1.522	1.045	0.590
avg both	0.805		1.427		1.069		1.203		0.818	

B.2 Data plots for each region of company operation

Figure B.1: Data for region 1.

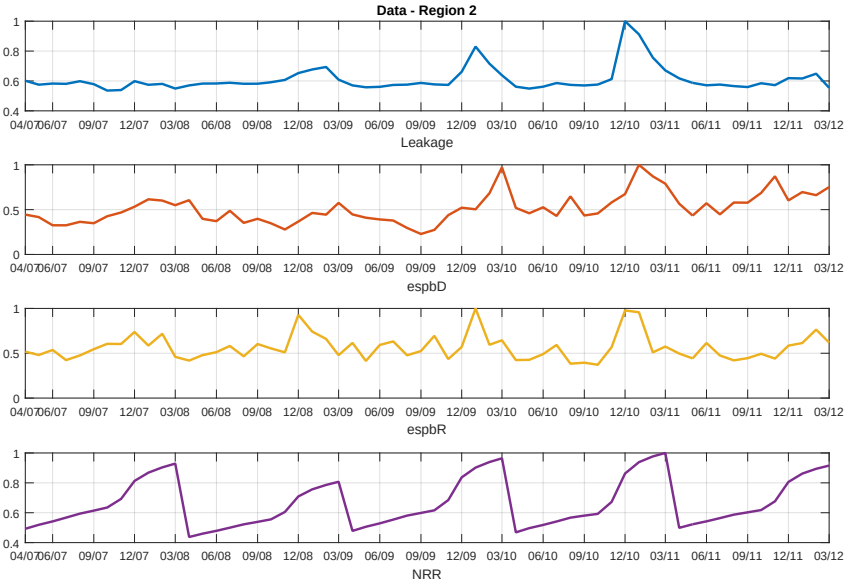

Figure B.2: Data for region 2.

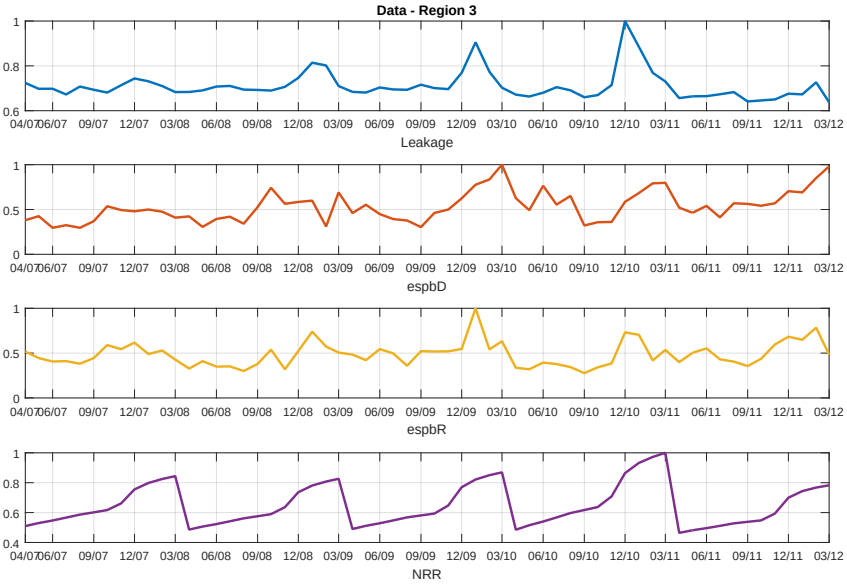

Figure B.3: Data for region 3.

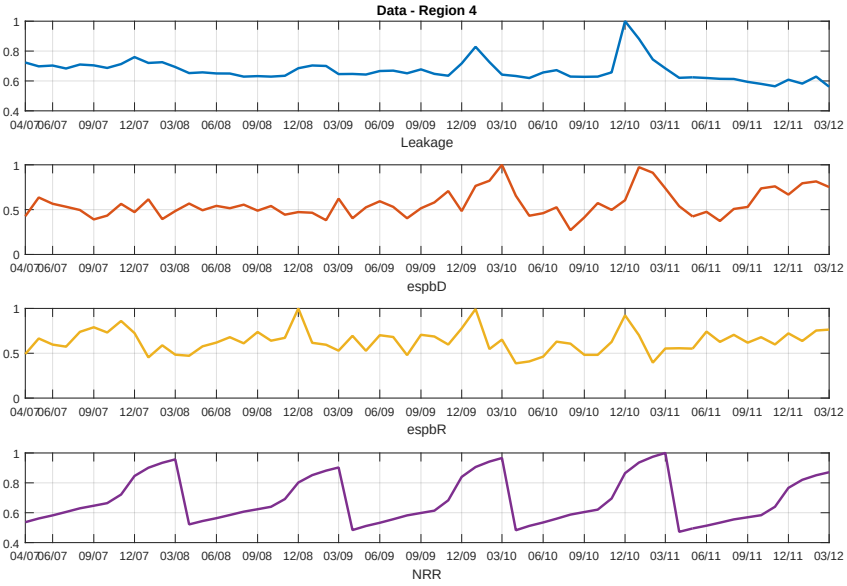

Figure B.4: Data for region 4.

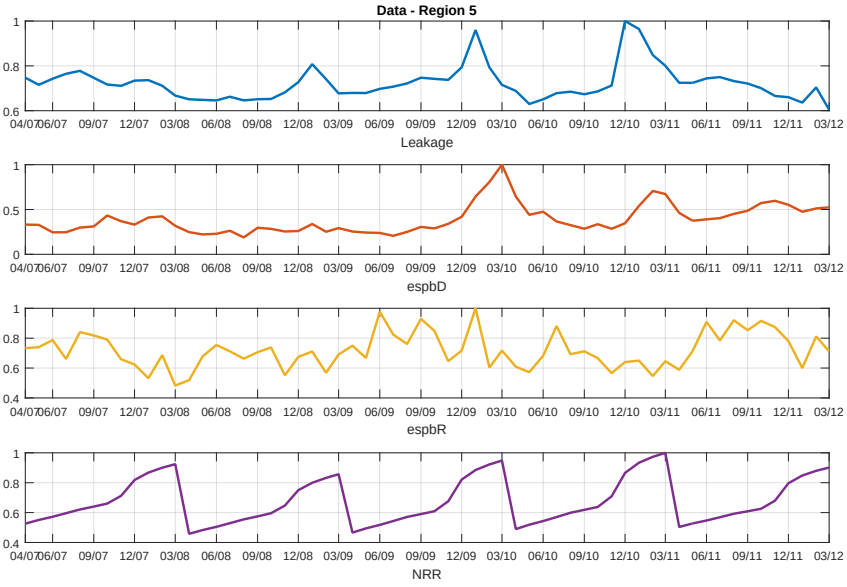

Figure B.5: Data for region 5.

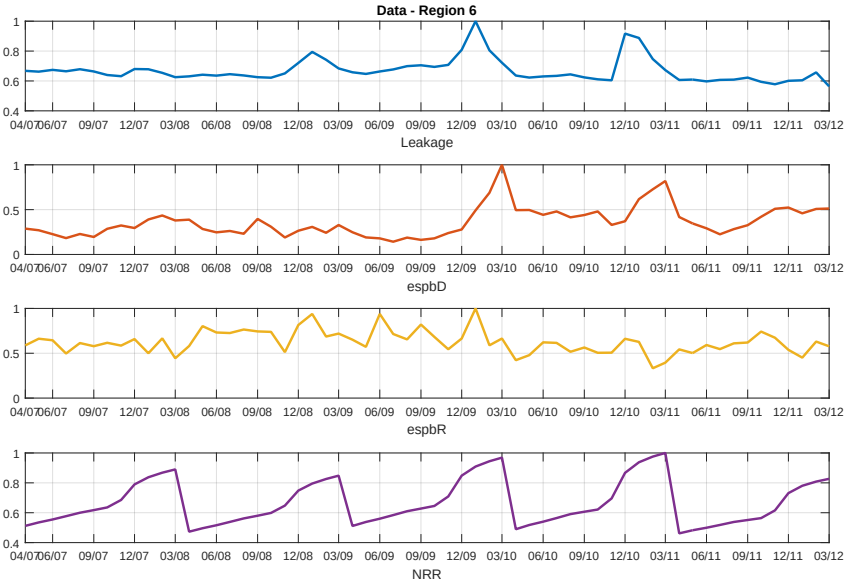

Figure B.6: Data for region 6.

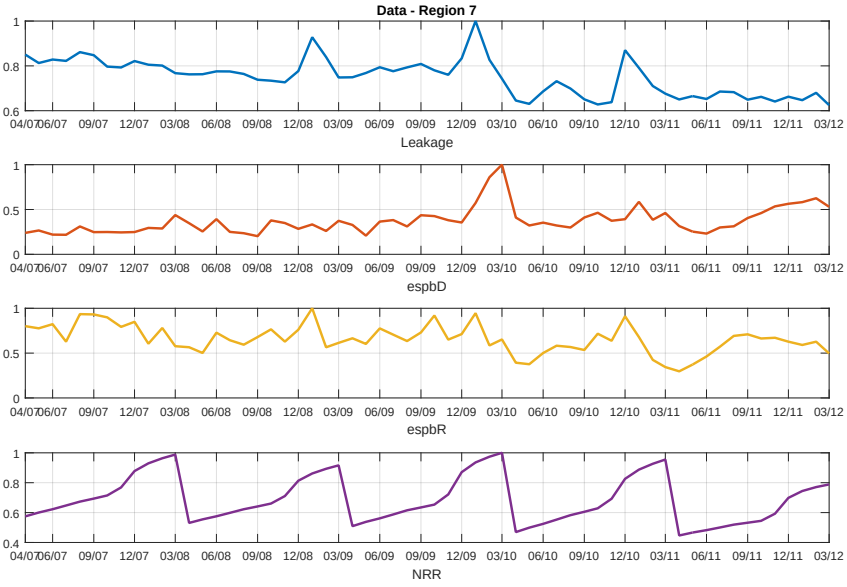

Figure B.7: Data for region 7.

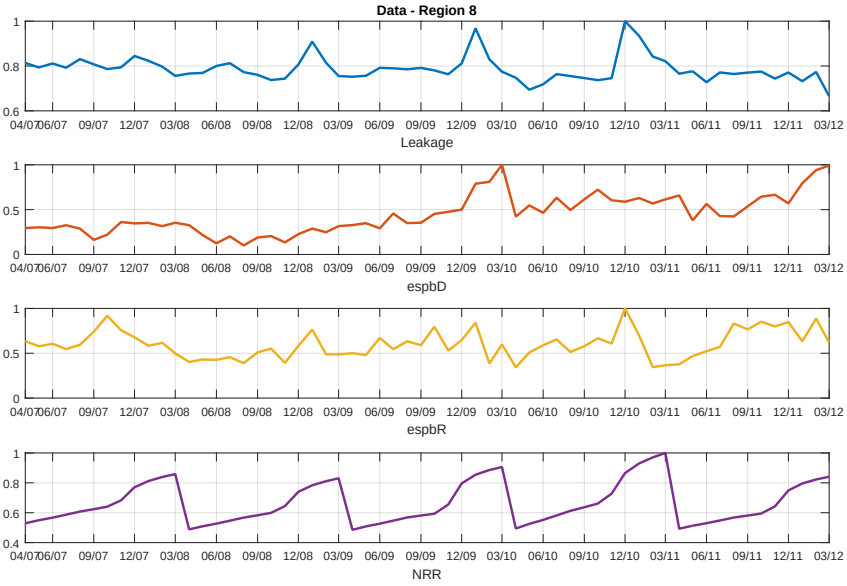

Figure B.8: Data for region 8.

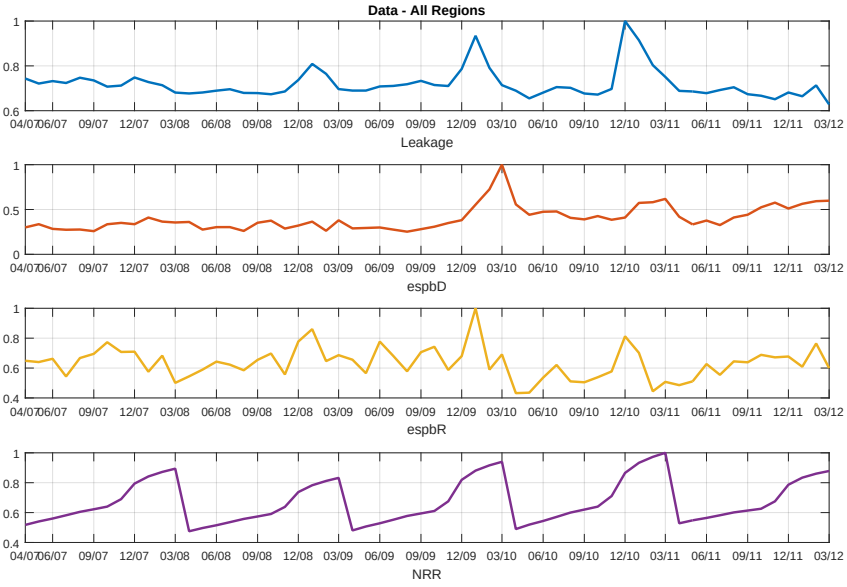

Figure B.9: Data for all regions combined.

Bibliography

Almeida, A. B. & Ramos, H. M. (2010), 'Water supply operation: diagnosis and reliability analysis in a Lisbon pumping system', *Journal of Water Supply: Research and Technology - AQUA* 59(1), 66.

Alvisi, S. & Franchini, M. (2011), 'Fuzzy neural networks for water level and discharge forecasting with uncertainty', *Environmental Modelling & Software* 26(4), 523–537.

Andreu, J. & Angelov, P. (2010), Forecasting time-series for NN GC1 using Evolving Takagi-Sugeno (eTS) Fuzzy Systems with on-line inputs selection, *in* 'International Conference on Fuzzy Systems', IEEE, Barcelona, 1–5.

Angelov, P. (2004*a*), 'A fuzzy controller with evolving structure', *Information Sciences* 161(1-2), 21–35.

Angelov, P. (2004*b*), 'An approach for fuzzy rule-base adaptation using on-line clustering', *International Journal of Approximate Reasoning* 35(3), 275–289.

Angelov, P. (2011), 'Fuzzily Connected Multimodel Systems Evolving Autonomously From Data Streams.', *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society* 41(4), 898–910.

Angelov, P. & Filev, D. P. (2004), 'An approach to online identification of Takagi-Sugeno fuzzy models.', *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society* 34(1), 484–98.

Angelov, P. & Filev, D. P. (2005), Simpl_eTS: a simplified method for learning evolving Takagi-Sugeno fuzzy models, *in* 'The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ '05.', IEEE, 1068–1073.

Angelov, P., Filev, D. P. & Kasabov, N., eds (2010), *Evolving Intelligent Systems*, John Wiley & Sons, Inc., Hoboken, NJ, USA.

Angelov, P. & Kordon, A. (2010), 'Adaptive inferential sensors based on evolving fuzzy models', *Systems, Man, and Cybernetics, Part B: ...* 40(2), 529–539.

Angelov, P., Kordon, A. & Zhou, X. (2008), Evolving fuzzy inferential sensors for process industry, *in* '2008 3rd International Workshop on Genetic and Evolving Systems', Vol. 44, IEEE, 41–46.

Angelov, P. & Lughofer, E. (2008), 'Data-driven evolving fuzzy systems using eTS and FLEXFIS: comparative analysis', *International Journal of General Systems* 37(1), 45–67.

Angelov, P., Lughofer, E. & Zhou, X. (2008), 'Evolving fuzzy classifiers using different model architectures', *Fuzzy Sets and Systems* 159(23), 3160–3182.

Angelov, P., Victor, J., Dourado, A. & Filev, D. P. (2004), On-line evolution of Takagi-Sugeno fuzzy models, *in* '2nd IFAC Workshop on Advanced Fuzzy/Neural Control'.

Angelov, P. & Zhou, X. (2006), Evolving Fuzzy Systems from Data Streams in Real-Time, *in* '2006 International Symposium on Evolving Fuzzy Systems', Vol. 00, IEEE, 29–35.

Armstrong, J. S. (1998), Integration of statistical methods and judgment for time series forecasting: Principles from empirical research, *in* 'Forecasting with Judgment', John Wiley & Sons Ltd., 269–293.

Assimakopoulos, V. & Nikolopoulos, K. (2000), 'The theta model: a decomposition approach to forecasting.', *International journal of forecasting* 16(4), 521–530.

Babovic, V., Drécourt, J.-P., Keijzer, M. & Friss Hansen, P. (2002), 'A data mining approach to modelling of water supply assets', *Urban Water* 4(4), 401–414.

Ben Ghalia, M. & Wang, P. P. (2000), 'Intelligent system to support judgmental business forecasting: the case of estimating hotel room demand', *IEEE Transactions on Fuzzy Systems* 8(4), 380–397.

Bennett, C., Stewart, R. A. & Beal, C. D. (2013), 'ANN-based residential water end-use demand forecasting model', *Expert Systems with Applications* 40(4), 1014–1023.

Bezdek, J. C., Ehrlich, R. & Full, W. (1984), 'FCM: The fuzzy c-means clustering algorithm', *Computers & Geosciences* 10(2), 191–203.

Birek, L., Petrovic, D. & Boylan, J. E. (2011), A fuzzy logic based approach to leakage forecasting in water industry, *in* 'International Symposium of Forecasitng 2011 Proceedings', Internationl Institute of Forecasters, Prague, 1–12.

Birek, L., Petrovic, D. & Boylan, J. E. (2014), 'Water leakage forecasting: the application of a modified fuzzy evolving algorithm', *Applied Soft Computing* 14, 305–315.

Brown, R. G. (1959), 'Statistical forecasting for inventory control', *McGraw-Hill, New York* .

Cardoso, G. & Gomide, F. (2007), 'Newspaper demand prediction and replacement model based on fuzzy clustering and rules', *Information Sciences* 177(21), 4799–4809.

Chen, S.-M. & Chang, Y.-C. (2010), 'Multi-variable fuzzy forecasting based on fuzzy clustering and fuzzy rule interpolation techniques', *Information Sciences* 180(24), 4772–4783.

Chen, S.-M. & Tanuwijaya, K. (2011), 'Multivariate fuzzy forecasting based on fuzzy time series and automatic clustering techniques', *Expert Systems with Applications* 38(8), 10594–10605.

Chiu, S. L. (1994), 'Fuzzy model identification based on cluster estimation', *Journal of intelligent and Fuzzy systems* 22(6), 827–841.

Collopy, F. & Armstrong, J. S. (1992), 'Rule-Based Forecasting: Development and Validation of an Expert Systems Approach to Combining Time Series Extrapolations', *Management Science* 38(10), 1394–1414.

Crone, S. F., Hibon, M. & Nikolopoulos, K. (2011), 'Advances in forecasting with neural networks? Empirical evidence from the NN3 competition on time series prediction', *International Journal of Forecasting* 27(3), 635–660.

Dalkey, N. & Helmer, O. (1963), 'An Experimental Application of the DELPHI Method to the Use of Experts', *Management Science* 9(3), 458–467.

Davydenko, A. & Fildes, R. (2013), 'Measuring Forecasting Accuracy: The Case Of Judgmental Adjustments To Sku-Level Demand Forecasts', *International Journal of Forecasting* 29(3), 510–522.

Dovzan, D., Logar, V. & Skrjanc, I. (2012), Solving the sales prediction problem with fuzzy evolving methods, *in* '2012 IEEE Congress on Evolutionary Computation', IEEE, 1–8.

Duru, O. (2012), 'A multivariate model of fuzzy integrated logical forecasting method (M-FILF) and multiplicative time series clustering: A model of time-varying volatility for dry cargo freight market', *Expert Systems with Applications* 39(4), 4135–4142.

Egrioglu, E., Aladag, C. H., Yolcu, U., Uslu, V. R. & Erilli, N. A. (2011), 'Fuzzy time series forecasting method based on GustafsonKessel fuzzy clustering', *Expert Systems with Applications* 38(8), 10355–10357.

Fildes, R. A., Randall, A. & Stubbs, P. (1997), 'One day ahead demand forecasting in the utility industries: two case studies', *Journal of the Operational Research Society* 48(1), 15–24.

Fildes, R. & Goodwin, P. (2013), 'Forecasting support systems: What we know, what we need to know', *International Journal of Forecasting* 29(2), 290–294.

Fildes, R., Goodwin, P., Lawrence, M. J. & Nikolopoulos, K. (2009), 'Effective forecasting and judgmental adjustments: an empirical evaluation and strategies for improvement in supply-chain planning', *International Journal of Forecasting* 25(1), 3–23.

Francisque, A., Rodriguez, M. J., Sadiq, R., Miranda, L. F. & Proulx, F. (2009), 'Prioritizing monitoring locations in a water distribution network: a fuzzy risk approach', *Journal of Water Supply: Research and Technology AQUA* 58(7), 488.

Gardner, E. S. (1985), 'Exponential smoothing: The state of the art', *Journal of Forecasting* 4(1), 1–28.

Gardner, E. S. (2006), 'Exponential smoothing: The state of the art – Part II', *International Journal of Forecasting* 22(4), 637–666.

Goodwin, P. (2000), 'Improving the voluntary integration of statistical forecasts and judgment', *International Journal of Forecasting* 16(1), 85–99.

Goodwin, P. (2002), 'Integrating management judgment and statistical methods to improve short-term forecasts', *Omega* 30(2), 127–135.

Goodwin, P. (2010), 'The Holt-Winters Approach to Exponential Smoothing : 50 Years Old and Going Strong', *Foresight* 30–33.

Goodwin, P., Sinan Gönül, M. & Önkal, D. (2012), 'Antecedents and effects of trust in forecasting advice', *International Journal of Forecasting* 29(2), 354–366.

Hadavandi, E., Shavandi, H. & Ghanbari, A. (2011), 'An improved sales forecasting approach by the integration of genetic fuzzy systems and data clustering: Case study of printed circuit board', *Expert Systems with Applications* 38(8), 9392–9399.

Hagan, M. T. & Behr, S. M. (1987), 'The time series approach to short term load forecasting', *Power Systems, IEEE Transactions on* 2(3), 785–791.

Hamerly, G. & Elkan, C. (2002), Alternatives to the k-means algorithm that find better clusterings, *in* 'Proceedings of the Eleventh International Conference on Information and Knowledge Management', CIKM '02, ACM, New York, NY, USA, 600–607.

Hand, D. J. (2009), 'Mining the past to determine the future: Problems and possibilities', *International Journal of Forecasting* 25(3), 441–451.

Hartigan, J. A. (1975), *Clustering Algorithms*, 99th edn, John Wiley & Sons, Inc., New York, NY, USA.

Hartigan, J. A. & Wong, M. A. (1979), 'Algorithm AS 136: A k-means clustering algorithm', *Applied statistics* 28(1), 100–108.

Harvey, N. & Harries, C. (2004), 'Effects of judges' forecasting on their later combination of forecasts for the same outcomes', *International Journal of Forecasting* 20(3), 391–409.

Herrera, M., Torgo, L., Izquierdo, J. & Pérez-García, R. (2010), 'Predictive models for forecasting hourly urban water demand', *Journal of Hydrology* 387(1-2), 141–150.

Holt, C. C. (2004), 'Forecasting seasonals and trends by exponentially weighted moving averages', *International Journal of Forecasting* 20(1), 5–10.

Hoyle, T. (2010), 'District metering the front line in the fight against leaks'. http://www.abb.co.uk/cawp/seitp202/8b1bac80a0dcc280c1257707004aba9d.aspx

Hyndman, R. J. & Athanasopoulos, G. (2014), *Forecasting: principles and practice*, OTexts.

Hyndman, R. J. & Koehler, A. B. (2006), 'Another look at measures of forecast accuracy', *International Journal of Forecasting* 22(4), 679–688.

Hyndman, R. J., Koehler, A. B., Ord, J. K. & Snyder, R. D. (2008), *Forecasting with Exponential Smoothing*, Springer Series in Statistics, Springer Berlin Heidelberg, Berlin, Heidelberg.

Hyndman, R. J., Koehler, A. B., Snyder, R. D. & Grose, S. (2002), 'A state space framework for automatic forecasting using exponential smoothing methods', *International Journal of Forecasting* 18(3), 439–454.

Jain, A. K., Murty, M. N. & Flynn, P. J. (1999), 'Data clustering: a review', *ACM Computing Surveys* 31(3), 264–323.

Kasabov, N. K. & Song, Q. (2002), 'DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction', *Fuzzy Systems, IEEE Transactions on* 10(2), 144–154.

Kauko, K. & Palmroos, P. (2014), 'The Delphi method in forecasting financial markets-An experimental study', *International Journal of Forecasting* 30(2), 313–327.

Kavasseri, R. G. & Seetharaman, K. (2009), 'Day-ahead wind speed forecasting using f-arima models', *Renewable Energy* 34(5), 1388–1393.

Kelo, S. M. & Dudul, S. V. (2011), 'Short-term Maharashtra state electrical power load prediction with special emphasis on seasonal changes using a novel focused time lagged recurrent neural network based on time delay neural network model', *Expert Systems with Applications* 38(3), 1554–1564.

Kourentzes, N., Barrow, D. K. & Crone, S. F. (2014), 'Neural network ensemble operators for time series forecasting', *Expert Systems with Applications* 41(9), 4235–4244.

Lalonde, A. (2004), Assessing Water Losses: Using "Big System" concepts on "Small Systems", *in* '2004 Clean and Safe Drinking Water Workshop'.

Lawrence, M. J., Goodwin, P., O'Connor, M. & Önkal, D. (2006), 'Judgmental forecasting: A review of progress over the last 25years', *International Journal of Forecasting* 22(3), 493–518.

Lee, J. K. & Yum, C. S. (1998), 'Judgmental adjustment in time series forecasting using neural networks', *Decision Support Systems* 22(2), 135–154.

Li, Y. P. & Huang, G. H. (2009), 'Fuzzy-stochastic-based violation analysis method for planning water resources management systems with uncertain information', *Information Sciences* 179(24), 4261–4276.

Liu, H. & Motoda, H. (1998), *Feature Selection for Knowledge Discovery and Data Mining*, Springer Berlin Heidelberg.

Lughofer, E. (2008), 'FLEXFIS: A Robust Incremental Learning Approach for Evolving TakagiSugeno Fuzzy Models', *IEEE Transactions on Fuzzy Systems* 16(6), 1393–1410.

Lughofer, E. (2011*a*), *Evolving Fuzzy Systems Methodologies, Advanced Concepts and Applications*, Vol. 266 of *Studies in Fuzziness and Soft Computing*, 1st edn, Springer Berlin Heidelberg, Berlin, Heidelberg.

Lughofer, E. (2011*b*), 'On-line incremental feature weighting in evolving fuzzy classifiers', *Fuzzy Sets and Systems* 163(1), 1–23.

Lughofer, E. & Angelov, P. (2011), 'Handling drifts and shifts in on-line data streams with evolving fuzzy systems', *Applied Soft Computing* 11(2), 2057–2068.

Lughofer, E., Bouchot, J.-L. & Shaker, A. (2011), 'On-line elimination of local redundancies in evolving fuzzy systems', *Evolving Systems* 2(3), 165–187.

Luna, I. & Ballini, R. (2011), 'Top-down strategies based on adaptive fuzzy rule-based systems for daily time series forecasting', *International Journal of Forecasting* 27(3), 708–724.

Luna, I., Lopes, J. E. G., Ballini, R. & Soares, S. (2009), Verifying the Use of Evolving Fuzzy Systems for Multi-Step Ahead Daily Inflow Forecasting, *in* '2009 15th International Conference on Intelligent System Applications to Power Systems', IEEE, 1–6.

Luna, I., Soares, S. & Ballini, R. (2007), An Adaptive Hybrid Model for Monthly Streamflow Forecasting, *in* '2007 IEEE International Fuzzy Systems Conference', IEEE, 1–6.

Maciel, L., Gomide, F. & Ballini, R. (2011), 'Evolving fuzzy systems for pricing fixed income options', *2011 IEEE Workshop on Evolving and Adaptive Intelligent Systems (EAIS)* 54–61.

Mackey, M. C. & Glass, L. (1977), 'Oscillation and chaos in physiological control systems', *Science* 197(4300), 287–289.

Makinde, O. D., Moitsheki, R. J., Jana, R. N., Bradshaw-Hajek, B. H. & Khan, W. A. (2014), 'Nonlinear Fluid Flow and Heat Transfer', *Advances in Mathematical Physics* 2014, 1–2.

Makridakis, S. & Hibon, M. (2000), 'The M3-Competition: results, conclusions and implications', *International Journal of Forecasting* 16(4), 451–476.

Mamdani, E. H. & Assilian, S. (1975), 'An experiment in linguistic synthesis with a fuzzy logic controller', *International Journal of Man-Machine Studies* 7(1), 1–13.

Mangalova, E. & Agafonov, E. (2014), 'Wind power forecasting using the k-nearest neighbors algorithm', *International Journal of Forecasting* 30(2), 402–406.

Market Transformation Programme (2008), BNWAT22 : Domestic water consumption in domestic and non-domestic properties, Technical report, Market Transformation Programme.

Marshallsay, D. (2005), Understanding Domestic Water Use In The Context Of Demand Management and Forecasting, Technical report, WRc plc.

Morais, D. C. & de Almeida, A. T. (2007), 'Group decision-making for leakage management strategy of water network', *Resources, Conservation and Recycling* 52(2), 441–459.

Nasseri, M., Moeini, A. & Tabesh, M. (2011), 'Forecasting monthly urban water demand using Extended Kalman Filter and Genetic Programming', *Expert Systems with Applications* 38(6), 7387–7395.

Obradović, D. (2000), 'Modelling of demand and losses in real-life water distribution systems', *Urban Water* 2(2), 131–139.

Omron Healthcare (2016), 'Omron Healthcare 100 million units sold — Toward Accurate Blood Pressure Monitoring at Home'.
http://www.healthcare.omron.co.jp/bpm/english/history/03.html

Ord, J. K., Koehler, A. B. & Snyder, R. D. (1997), 'Estimation and Prediction for a Class of Dynamic Nonlinear Statistical Models', *Journal of the American Statistical Association* 92(440), 1621.

Ord, K. & Fildes, R. (2013), *Principles of business forecasting.*, 1st edn, Cengage Learning.

Pai, P.-F. & Lin, C.-S. (2005), 'A hybrid ARIMA and support vector machines model in stock price forecasting', *Omega* 33(6), 497–505.

Pearson, D. & Trow, S. W. (2005), Calculating the Economic Levels of Leakage, *in* 'Leakage 2005 Conference Proceedings', 1–16.

Pedrycz, W. (1994), 'Why triangular membership functions?', *Fuzzy Sets and Systems* 64(1), 21–30.

Poulakis, Z., Valougeorgis, D. & Papadimitriou, C. (2003), 'Leakage detection in water pipe networks using a Bayesian probabilistic framework', *Probabilistic Engineering Mechanics* 18(4), 315–327.

Ramos, J. V. & Dourado, A. (2003), Evolving Takagi-Sugeno Fuzzy Models, Technical report, University of Coimbra, Centre for Informatics and Systems.

Rowe, G. & Wright, G. (1999), 'The Delphi technique as a forecasting tool: issues and analysis', *International Journal of Forecasting* 15(4), 353–375.

Savic, D., Walters, G., Ashcroft, P. G. & Arscott, A. (1997), 'Hydroinformatics technology and maintenance of UK water networks', *Journal of Quality in Maintenance Engineering* 3(4), 289–301.

Severn Trent Water (2013), 'Our story : About us : Severn Trent Water'. http://www.stwater.co.uk/about-us/our-story/

Song, Q. & Chissom, B. S. (1993*a*), 'Forecasting enrollments with fuzzy time series Part I', *Fuzzy Sets and Systems* 54(1), 1–9.

Song, Q. & Chissom, B. S. (1993*b*), 'Fuzzy time series and its models', *Fuzzy Sets and Systems* 54(3), 269–277.

Song, Q. & Chissom, B. S. (1994), 'Forecasting enrollments with fuzzy time series Part II', *Fuzzy Sets and Systems* 62(1), 1–8.

Spithourakis, G. P., Petropoulos, F., Nikolopoulos, K. & Assimakopoulos, V. (2015), 'Amplifying the learning effects via a Forecasting and Foresight Support System', *International Journal of Forecasting* 31(1), 20–32.

Stepnicka, M., Cortez, P., Donate, J. P. & Štěpničková, L. (2012), 'Forecasting seasonal time series with computational intelligence: On recent methods and the potential of their combinations', *Expert Systems with Applications* .

Syntetos, A. A., Nikolopoulos, K. & Boylan, J. E. (2010), 'Judging the judges through accuracy-implication metrics: The case of inventory forecasting', *International Journal of Forecasting* 26(1), 134–143.

Syntetos, A. A., Nikolopoulos, K., Boylan, J. E., Fildes, R. & Goodwin, P. (2009), 'The effects of integrating management judgement into intermittent demand forecasts', *International Journal of Production Economics* 118(1), 72–81.

Takagi, T. & Sugeno, M. (1985), 'Fuzzy identification of systems and its applications to modeling and control', *IEEE Transactions on Systems, Man, and Cybernetics* SMC-15(1), 116–132.

Taylor, J. W. (2003), 'Exponential smoothing with a damped multiplicative trend', *International Journal of Forecasting* 19(4), 715–725.

Taylor, J. W. & Buizza, R. (2003), 'Using weather ensemble predictions in electricity demand forecasting', *International Journal of Forecasting* 19(1), 57–70.

Trapero, J. R., Pedregal, D. J., Fildes, R. & Kourentzes, N. (2013), 'Analysis of judgmental adjustments in the presence of promotions', *International Journal of Forecasting* 29(2), 234–243.

UK, M. O. (2015), 'UK climate - Historic station data'. http://www.metoffice.gov.uk/public/weather/climate-historic/

UKWIR (2005), *Natural Rate of Rise in Leakage*, United Kingdom Water Industry Research.

Vrbanek, J. & Wang, W. (2009), An Evolving Fuzzy Scheme for Dynamic System Forecasting, *in* J. Zhu, ed., 'Forecasting Models Methods & Applications', 1st edn, iConcept Press Ltd., chapter 3.

Wang, L.-X. & Mendel, J. M. (1992), 'Fuzzy basis functions, universal approximation, and orthogonal least-squares learning.', *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council* 3(5), 807–14.

Webby, R., O'Connor, M. & Edmundson, B. (2005), 'Forecasting support systems for the incorporation of event information: An empirical investigation', *International Journal of Forecasting* 21(3), 411–423.

Wellstead, P. E. & Zarrop, M. B. (1991), *Self-tuning systems: control and signal processing*, Wiley-Blackwell.

Weron, R. (2014), 'Electricity price forecasting: A review of the state-of-the-art with a look into the future', *International Journal of Forecasting* 30(4), 1030–1081.

Wilamowski, B. M. & Irwin, J. D., eds (2011), *The Industrial Electronics Handbook, Second Edition*, CRC Press, Alabama, USA.

Winters, P. R. (1960), 'Forecasting Sales by Exponentially Weighted Moving Averages', *Management Science* 6(3), 324–342.

Yager, R. R. (1984), 'Measuring the quality of linguistic forecasts', *International Journal of Man-Machine Studies* 21(3), 253–257.

Yager, R. R. & Filev, D. P. (1994), 'Approximate clustering via the mountain method', *IEEE Transactions on Systems, Man, and Cybernetics* 24(8), 1279–1284.

Zadeh, L. A. (1965), 'Fuzzy sets', *Information and control* 8(3), 338–353.

Zhang, G., Eddy Patuwo, B. & Y. Hu, M. (1998), 'Forecasting with artificial neural networks:', *International Journal of Forecasting* 14(1), 35–62.