Quasi-optimization of Neuro-fuzzy Expert Systems using Asymptotic Least-squares and Modified Radial Basis Function Models

Intelligent Planning of Operational Research Problems

Samer Mohammed Jaber Mubarak

Supervisor: Andrew Crampton

Department of Computer Science
University of Huddersfield

This thesis is submitted for the degree of
Doctor of Philosophy

School of Computing and
Engineering

May 2021

And say,
"my Lord" increase me in knowledge

To the spirit of

Dr Mohammed Ali Alkailany

your advice and guidance as yet inspires me

we will never forget you . . .

Declaration

I hereby declare that the contents of this thesis are original, except where specific reference is made to the work of others. This work has not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This thesis is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. The author of this thesis (including any non reproduced figures and/or tables) owns any copyright in it (the —Copyright), and he has given the University of Huddersfield the right to use such Copyright for any Librarian, administrative, promotional, educational and/or teaching purposes.

Samer Mohammed Jaber Mubarak

May 2021

Acknowledgements

First of all, I would like to thank Allah, the Almighty, on whom ultimately we depend for sustenance and guidance, for providing me with the inspiration and strength to finish this research and introduce it as a simple contribution to the scientific effort of humanity. I am highly indebted to my parents, wife, children, brothers and sisters for their constant encouragement and support throughout this study. It would not be possible to write this thesis without support from them. From the bottom of my heart, I would like to say thank you.

With polite respect, I would like to express my gratitude and appreciation to my patient and supportive supervisor, Professor Andrew Crampton, for his guidance during this research. My words fall short to thank you, Prof. Andrew. I would also like to thank my co-supervisor Dr Jennifer Carter for her consistent assistance during the running of this research. It was a great opportunity for me to improve my ability to learn, as well as my researching capabilities. A big thanks to all academic and administrative staff that supported me during my study period. It was an elegant experience to live and interact with the lovely British society during my research time, so, thank you to all nice and friendly people that I met in the UK.

I would like to express my sincere gratitude to the Republic of Iraq/Ministry of Higher Education and Scientific Research/University of Baghdad who funded my scholarship. A profound thanks to the Iraqi Cultural Attaché in the London/UK, who provided me with all needed support during my study time in the UK. I also wish to thank the experts at the Iraqi Ministry of Oil for there cooperative, in particular, Mr. Mohammed Jasim Ajam for his support and encouragement throughout the problem analysis process.

I am also grateful to thank Dr Abdulkareem Alfaisal / Chairman of the Iraqi Prime Minister Advisory Commission for his help and support before and during my study. Also, a big thank you to my colleagues, Dr Mosttafa Alghadhi / Huddersfield University, Prof. Marwan Abdul-Hameed Ashour and Dr Omar Mohammed Naser Alashari /University of Baghdad, for their continued motivation and sincere help during my studies.

The author wishes to thank Dr Nazmul Siddique and Dr Hojjat Adeli for granting their permission to the author to adapt figures from their book [117] to be included in the third chapter of this thesis.

Abstract

The uncertainty found in many industrialization systems poses a significant challenge; particularly in modelling production planning and optimizing manufacturing flow. In aggregate production planning, a key requirement is an ability to accurately predict demand from a range of influencing factors, such as consumption for example. Accurately building such causal models can be problematic if significant uncertainties are present, such as when the data are fuzzy, uncertain, fluctuate and are non-linear. AI models, such as Adaptive Neuro-Fuzzy Inference Systems (ANFIS), can cope with this better than most but even these well-established approaches fail if the data is scarce, poorly scaled and noisy.

ANFIS is a combination of two approaches; Sugeno-type Fuzzy Inference System (FIS) and Artificial Neural Networks (ANN). Two sets of parameters are required to define the model: premise parameters and consequent parameters. Together, they ensure that the correct number and shape of membership functions are used and combined to produce reliable outputs. However, optimally determining values for these parameters can only happen if there are enough data samples representing the problem space to ensure that the method can converge. Mitigation strategies are suggested in the literature, such as fixing the premise parameters to avoid over-fitting, but, for many practitioners, this is not an adequate solution, as their expertise lies in the application domain, not in the AI domain.

The work presented here is motivated by a real-world challenge in modelling and predicting demand for the gasoline industry in Iraq, an application where both the quality and quantity of the training data can significantly affect prediction accuracy. To overcome data scarcity, we propose novel data expansion algorithms that are able to augment the original data with new samples drawn from the same distribution. By using a combination of carefully chosen and suitably modified radial basis function models, we show how robust methods can overcome problems of over-smoothing at boundary values and turning points. We further show how transformed least-squares (TLS) approximation of the data can be constructed to asymptotically bound the effect of outliers to enable accurate data expansion to take place. Though the problem of scaling/normalization is well understood in some AI applications, we assess the impact on model accuracy for two specific scaling techniques. By comparing and contrasting a range of data scaling and data expansion methods, we can evaluate their effectiveness in reducing prediction error.

Throughout this work, the various methods are explained and expanded upon using the case study drawn from the oil and gas industry in Iraq which focuses on the accurate prediction of yearly gasoline consumption. This case study, and others are used to demonstrate, empirically, the effectiveness of the approaches presented when compared to current state of the art. Finally, we present a tool developed in Matlab to allow practitioners to experiment with all methods and options presented in this work.

Table of contents

List of figures

List of tables

Nomenclature

Greek Symbols

η	Learning Rate
μ	Membership Value
α	Neuron's Parameters

Other Symbols

$A.B$	Algebraic product of two sets
$A \ominus B$	The bounded difference of two sets
$A \oplus B$	The bounded summation of two sets
$A \times B$	Cartesian product of two sets
$\overline{A}$	Complement of a set A
$\overline{\overline{A}}$	Complement of Complement of a set A
$A - B$	Difference of a set A with respect to set B
$\emptyset$	Empty set
$A \cap B$	Intersection of two sets A and B
$A \neq B$	A is not equal to B
$A \subset B$	A is proper subset of B
$A \cup B$	Union of two sets A and B
NOT	Complement of a set

AND Intersection of two sets

OR Union of two fuzzy sets

$y_A(x)$ Characteristic Function in set A for the element x

$\mu_A(x)$ Degree of membership of the element x in set A

$f(x)$ Function of x

CON Concentration of a fuzzy set

DIL Dilation of a fuzzy set

X Universal set

Acronyms / Abbreviations

AI Artificial Intelligence

ANFIES Adaptive Neuro-Fuzzy Inference Expert System

ANFIS Adaptive Neuro-Fuzzy Inference System

ANN Artificial Neural Networks

APP Aggregate Production Planning

ARIMA Autoregressive Integrated Moving Average

CF Characteristic Function

Ch Chebyshev Scaling Method

COA Centroid of Area Method

CON Concentration

COS Centre of Sums Method

Cub Cubic Radial Basis Function

DENFISs Dynamic Evolving Neuro-Fuzzy Inference Systems

DIL Dilation

FES Fuzzy Expert System

FIS Fuzzy Inference System

FKB Fuzzy Knowledge Base

FK Fuzzy Knowledge base

FL Fuzzy Logic

FLS Fuzzy Logic System

GA Genetic algorithm

GD Gradient Descent Method

GRNN General Regression Neural Network

HANFIS Hierarchical Adaptive Neuro-Fuzzy Inference System

KB Knowledge Base

Lin Linear Radial Basis Function

LL Linguistic Labels

LSE Least-Squares Estimator

LS Least-Square Method

LV Linguistic Variable

LVQ Learning Vector Quantization Neural Networks

MA Moving Average

MD Membership Degrees

MF Membership Function

MLP Multi-layer Perceptron Neural Network

MLR Multiple Linear Regression

MOM Mean of Maximum Method

Mul Modified Multiquadric Radial Basis Function

NRMSE Normalized Root Mean Square Error

OR Operational Research

PSO Particle Swarm Optimization Algorithm

R^2 Coefficient of Determination

RBFNN Radial Basis Function Neural Networks

RBF Radial Basis Function

RENFSM Recurrent Error-Based Neuro-Fuzzy System with Momentum

RMSE Root Mean Square Error

SLS Standard Least-Square Method

TLS Transformed Least-Square Method

TSK Takagi–Sugeno model

WANFIS Wavelet-Adaptive Neuro-Fuzzy Inference System

Z-s Zero-mean or Z-scores Scaling Method

dsigmf Difference Sigmoidal Membership Function

gaussmf Gaussian Membership Function

gbellmf Generalized Bell Membership Function

trabmf Trapezoidal Membership Function

trimf Triangular Membership Function

PART 1

Introduction & Literature Review

Chapter 1

Introduction

1.1 Introduction

Planning can be considered as one of the most important factors in developing enterprises. Production planning and optimising the manufacturing flow represents an essential component in the planning process. It can balance the available resources and capacities from one side, with the demand or consumption from the other side; to give an idea to the management as to what quantity of materials and other resources are to be procured and when. Operational Research (OR) techniques can be considered as one of the effective tools in solving production planning problems [51, 138].

Rivett [110], classified seven basic structures of OR models. He showed that, in most cases, solving a real-life problem can not be achieved by using these seven standard OR approaches singly. Taha [128], stated that there is no one comprehensive approach that can solve all mathematical models. He said, "The complexity and type of any mathematical model dictates the nature of the solution method". Therefore, modelling real-world problems represents a considerable challenge. The modellers have to deal with a process that may contain various measurements. The main challenge is that when the problem is affected by human behaviours. In this case, the explicit (standard) mathematical models may fail to solve the real-life problems. In other words, if the standard OR mathematical models are apt to solve the resulting model, then the available algorithms can be used. Otherwise, if the mathematical relationships of the resulting model are too complex, then the modellers may have to use the simulation and inference approaches to solve the problem. In fact, the majority of real-life problems usually contain varying degrees of approximation [129]. In most cases, modelling a real-life problem needs to combine various approaches, rather than use one specific technique alone. This means obtaining a good optimized model will be more likely to use a combination of OR algorithms as well as other approaches. In this case, it can be called as combined optimization techniques.

As a matter of fact, most real-world problems usually contain several types of decision variables. Some variables can be measured, and others need to be estimated. However, building optimized planning models requires accurate input data. Therefore, the model builder has to take into account how to find the best method of predicting and estimating these input variables if needed. For instance, aggregate planning is one of the production planning methods that can be solved using the OR's mathematical models, such as linear programming, to provide decision-makers with an overall plan. It has been applied mainly to the production field, called Aggregate Production Planning (APP). The APP shows how the capacity (production, and inventory), resources (workforce, subcontracting, and facilities), and policies (hiring/firing, back-orders, and overtime) can match the required demand/consumption at the lowest cost [93]. Figure (1.1) shows how the capacities, resources,

policies and forecasting demand/consumption should be processed into operational research mathematical tools to provide the APP. It can be noticed that one of the input variables, i.e. demand/consumption, has to be estimated. The prediction of demand/consumption can be considered as one of the most important affecting factors in different fields of industrial and production planning. However, this can only be possible if the accuracy of the predicting model is reliable. Which means, the efficacy of a production planning model depends on the accuracy of the prediction model [88].

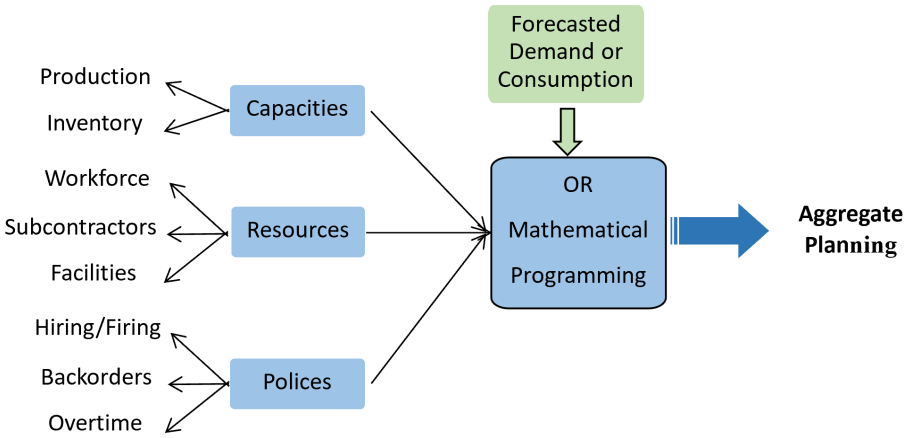


Fig. 1.1 Aggregate production planning operational activities.

Finding a suitable forecasting method that gives an accurate consumption prediction represents an important factor in planning. Conventionally, there are several reliable statistical forecasting approaches that are available. Generally, they are classified into qualitative and quantitative techniques [13]. Qualitative forecasting methods are used when relying on expert judgement and opinion to develop forecasts. It can be considered as an appropriate approach when the historical data, for the forecasted variables are rare or not applicable. Quantitative methods can be applied when the historical data is available, quantified, and reflective of the future [11]. The main concern, at this stage, is represented by the rise in complexity of the prediction model. For example, if the data contains fuzziness, uncertainty, and is fluctuating in its nature; furthermore, there is the possibility of scarcity and significant noise in the data. Other issues arise concerning the factors that affect consumption. Some factors can be specified using fuzzy or vague concepts. All these can affect the model and make it difficult for the traditional forecasting approaches to provide the desired accuracy.

Artificial Intelligence (AI) techniques, such as Adaptive Neuro-Fuzzy Inference System (ANFIS), can be used as an effective prediction technique, which can provide significant improvement when compared to the traditional forecasting methods. ANFIS can be specified as one of the best prediction approaches that use a combination of neural networks and the

fuzzy inference system algorithms. There are two main advantages to using ANFIS. Firstly, the ability to deal with fuzziness and uncertainty. The second is the capability of this system for performing parallel computations as well as its ability to simulate a nonlinear system that is hard to characterize using traditional forecasting models [166].

The motivation for the current research has come from a problem posed by the Iraqi Oil Industry. Here, they have a need to accurately predict oil consumption as part of an aggregate production planning (APP) process. However, they are faced with a number of challenges. The data that they use to guide their understanding of demand is uncertain. For example, they are aware that consumption is affected by the number of cars on the road. They are also aware that as the weather changes throughout the year, so does demand for gasoline. Consumption, in this context, is a complex value dependent not only on those factors affecting demand, but also on the capacity of gasoline that can be physically produced. Further exploration of this problem shows that there is limited data available on which to develop accurate models for prediction. additionally, as data is often recorded manually, there are questions raised regarding the accuracy of the data and it is not uncommon to see outliers present in the data set that can easily impact on the accuracy of the prediction models. Finding a robust solution to this problem has focused the research that will follow and has motivated the various approaches and techniques developed. However, further case studies have also been used to attest to the wider applicability of the proposed solutions.

In this work, we are proposing a range of models that can help to overcome the problems of forecasting within fuzzy environments, as well as dealing with the effects of scarce and noisy data. This research will follow the form of non-standardized optimized models, formed from a combination of ANFIS that works in parallel with the OR's mathematical models in order to solve the APP problem. The thesis structure includes 9 chapters, classified into four parts. Figure (1.2) demonstrate a summary diagram of this structure.

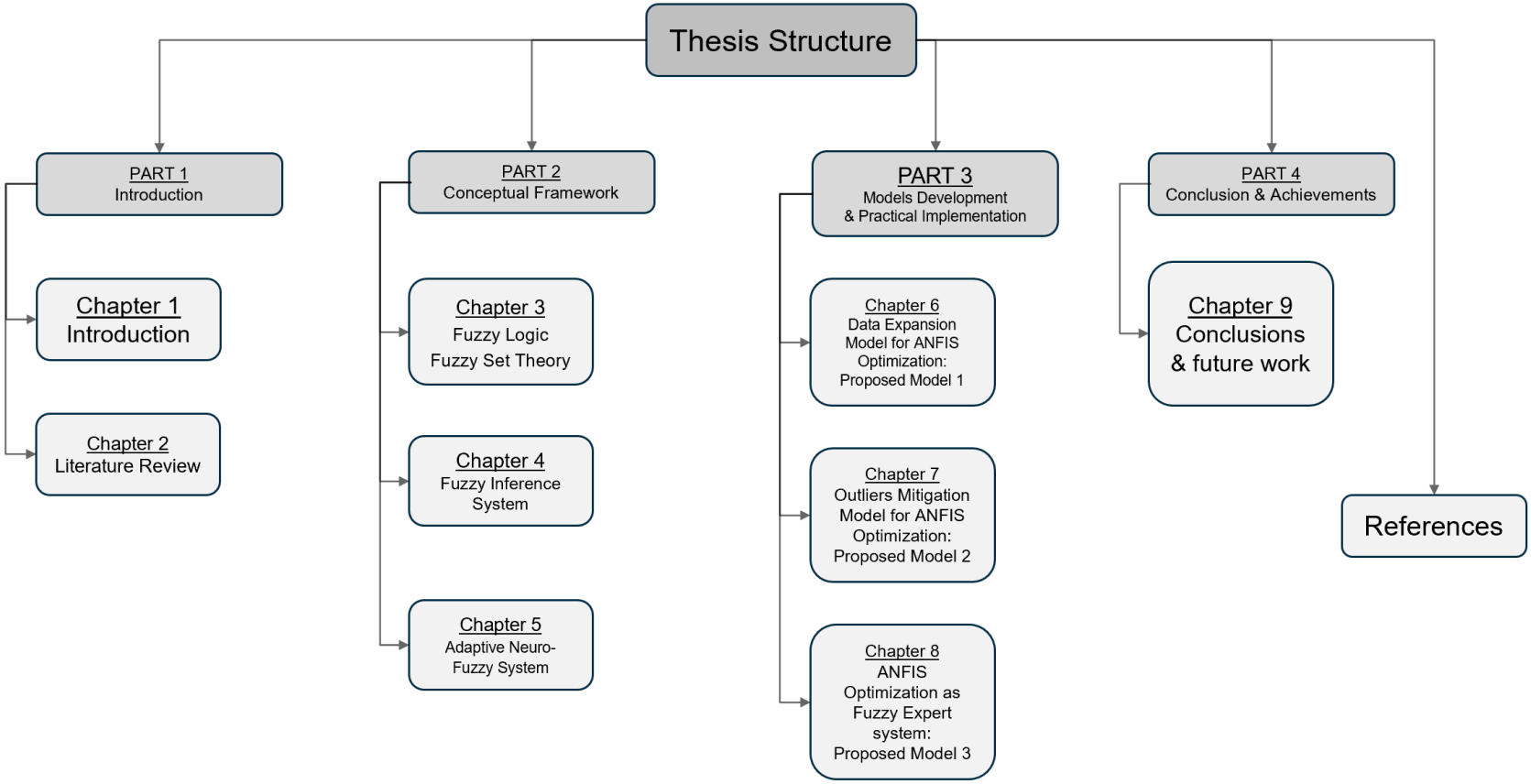


Fig. 1.2 Thesis structure.

1.2 Research Questions

This work will contribute both theoretically and empirically to an enhanced understanding of the solution methods that can be combined to optimize ANFIS as a prediction model. The following questions have been raised as a result of notable gaps in the literature relating to this problem:

1. What is the impact of data scarcity and outliers on the Adaptive Neuro-Fuzzy Inference System (ANFIS) optimization?
2. To what extent does the model complexity versus data sample size pose a significant, practical problem?
3. How can reliable data augmentation methods, such as Radial Basis Functions (Linear, Cubic, and modified Multiquadric), be constructed and used as robust continuous reliable data expansion models.
4. How can the modified Multiquadric RBF be tuned to provide robust and accurate data expansion?
5. How can Transformed Least Squares (TLS) be used to mitigate the effect of outliers when optimizing the model?

1.3 Research Statement

This work focuses on proposing robust mathematical models as a solution method for optimizing the ANFIS parameters when the data are scarce, significantly noisy, and poorly scaled. To demonstrate our novel approach, we look at the problem of consumption prediction as it represents one of the most important input variables of building the APP. Often, consumption is significantly affected by human behaviours. As such, we are often required to accept that there is a need to deal with fuzziness and uncertainty when working with this type of environment. When we are modelling data sets that contain fuzziness and uncertainty in its nature, then it is appropriate to construct prediction models using fuzzy inference systems (FIS). In addition, many *real-world* problems have limited data to work with and yet require models with high prediction accuracy. Nonetheless, the scarcity and noise of the data raise concerns when attempting to construct models of this type.

As has been mentioned in the previous section, ANFIS can be defined as a an expert system that uses a combination of artificial neural networks (ANN) and FIS. The solution method is based on a Takagi–Sugeno fuzzy inference system. The ANN component is

a supervised multi-layer feed-forward adaptive network. Whereas the main body of the fuzzy inference system (FIS) consists of four main components, i.e., fuzzification, fuzzy knowledge base, inference engine, and the defuzzification process. The fuzzy knowledge base, in turn, contains the types and numbers of Membership Functions (MFs) that can be utilized within the solving process. Each MF includes a specified number of function-specific parameters. The initial values of these parameters determine the initial shape of The MFs. These parameters are to be optimised using a hybrid learning algorithm in order to optimize ANFIS [61].

The main body of ANFIS networks consists of five layers. Thus, the numerical information represented by the historical data (input-output data pairs) is to be processed through these layers to fine-tune the network parameters. A hybrid learning algorithm, based on supervised learning from example data, must be applied to determine the parameters. The solution algorithm consists of two passes, i.e. forward and backward passes. In the forward-pass, the solution algorithm processes the so-called premise parameters. These are the Membership Functions (MFs) built-in parameters which are located at layer one. Whereas, in the backwards-pass, the network updates the linear parameters, named consequent parameters at layer four.

There are many applications where ANFIS (Sugeno-type fuzzy models) provides better models of an underlying problem than artificial neural networks alone. Nonetheless, there are instances when obtaining a reliable model for either approach can be difficult. However, this can only happen if there are enough data samples representing the problem space to ensure that the method can converge. When the data samples are greater than the number of total parameters (premise and consequent); then, there is enough coverage of the problem space to optimize the parameters. In contrast, if the data samples are less than the total number of parameters the data may not capture the problem well and, as a result, over-fitting can occur. Although ANFIS is theoretically known to be a universal approximator [61], training them accurately on small data sets is a significant problem in practice. When data is scarce, the literature recommends that the premise parameters should be kept fixed and set prior to training [163]. This will reduce the likelihood of over-fitting. Thus, only the consequent parameters should be optimised by the learning algorithm. However, forcing the premise parameters to stay fixed will limit the ability to optimize well when dealing with the fuzzy nature of the problem.

A search of both recent and historical studies has not revealed any work presenting an integrated solution to deal with this problem. One way to overcome this problem is to expand the rare data into a larger, but representative, data set. Therefore, we are introducing our proposed expansion models that can re-sample (augment) the original data with a

larger number of data samples. Moreover, the proposed model can mitigate the effect of outliers, where they exist, and produce a shape-preserving fitting curve that can keep the underlying trend of the original data. We will show that this can be achieved by proposing two mathematical models. The Radial Basis Functions (RBFs), such as Linear, Cubic, and modified Multiquadric, are to be used as the basis functions for both models.

In the first proposed model, we intend to apply a direct radial basis function (DRBF) as an interpolation model using the three types of (Linear, Cubic, and modified Multiquadric). This model will be used as an expansion model when the raw data sets has no noise. We will show that by replacing the discrete data with a carefully chosen and carefully optimised continuous model, we can re-sample from it at a finer granularity and use this to optimise prediction accuracy for an ANFIS model. As we construct our continuous prediction models, we will highlight some dangers of over smoothing that we have encountered and justify why shape-preserving models might be preferred in some cases. We will show how a modified Multiquadric radial basis function (RBF) approach yields a family of shape-preserving models that afford an amount of smoothness not found in other shape-preserving models such as the linear RBFs.

In the second proposed model, we are addressing the problem of modelling data containing noise classified as outliers. In order to overcome this problem, we intend to use the modified Least Squares approach, so-called Transformed Least Squares (TLS) as an approximation model. Again, RBFs (Linear, Cubic, and modified Multiquadric) will be utilized as the basis approximation functions under the form of TLS. We choose a TLS approach because traditional models, such as standard least squares, tend to cope poorly due to the influence of these points on the resultant model. We are particularly interested in finding ways to mitigate the effect of outliers on the model parameters as part of the data expansion process, rather than through the application of any additional pre-processing operations on the raw data. We will show that it is possible to asymptotically bound the contribution of any outliers to the error function being minimised and so produce good models for data expansion. Furthermore, we will show that it is possible to do this by employing a straightforward iteratively re-weighted least-squares approach.

Furthermore, we intend to develop an application tool that makes dealing with all of our proposed models straightforward and more manageable. This application will allow the user to transact with a friendly user interface. This application will give users the ability to choose from various options for each proposed model. Moreover, it will allow the user to select different types and numbers of MFs. This means we are developing an **expert system** using a tool that can simulate all possible fuzzy knowledge base combinations (MFs types and numbers). An Adaptive Neuro-Fuzzy Inference Expert System (ANFIES) can be employed

as a fuzzy expert tool which mimics expert knowledge, and human behaviour in order to provide an accurate prediction model. Often, domain experts have a better understanding of the data that describes the problem than they do of the mathematical models. From a practical point of view, it can be both difficult and time-consuming to try many pre-fixed strategies in pursuit of a good model. Therefore, creating a tool that can take the lead to examine all the available fuzzy knowledge bases and extract the best model is proposed. This best model (best fuzzy knowledge base), in its turn, must provide the optimum solution of the ANFIS. Depending on that, we have proposed our third model. This model will provide a tool to optimize ANFIES as a fuzzy expert system that can simulate the expert's knowledge. This proposed expert system application can provide the researchers with enough flexibility to compare and contrast between ANFIS models and select the best one. We will demonstrate the effectiveness of this approach through data taken from the oil/fuel industry; where data is scarce.

1.4 Research Objectives

Major objectives of this research are to:

1. Develop an optimization model to solve complex prediction causal problems that contain fuzziness, uncertainty, fluctuatiion, and non-linearity in their nature; particularly when it is accompany with the problems of scarcity in data as well as outliers.

2. Investigate the effectiveness of using combined mathematical tools and techniques such as expansion (interpolation and approximation), normalization, and scaling the data in optimizing the prediction accuracy of ANFIS.

3. Explore the reliability of using the radial basis functions (Linear, Cubic, and modified Multiquadric) as basis functions of interpolation and approximation models.

4. Explore the robustness of the transformed least squares (TLS) approach as an outlier mitigation model.

5. Develop a fuzzy expert system application tool that can mimic the expert's knowledge, and provide an appropriate environment for processing the data and solving ANFIS models; as well as an expert system that can be used with a novel application area, such as health care services, education etc.

Chapter 2

Literature Review

In this chapter, we present relevant literature related to our work. We have three main aims. One aim is to provide examples of research that demonstrate the move from traditional statistical methods to machine learning (ML) methods in prediction and forecasting. Many of these papers compare and contrast ML with traditional techniques for a range of problems. The next aim is to present research comparing ANFIS systems with other ML approaches, which also highlights the diverse areas of applications where ANFIS approaches are considered best. Finally we present the state-of-the-art for dealing with the sparse data and outlier mitigation problems.

2.1 Forecasting Techniques and ANFIS

Economically, forecasting represents an important tool that allows enterprises to predict into the future in order to plan their demand, consumption, sales, and production etc. Therefore, finding reliable and efficient forecasting techniques are in high demand. The literature shows that many forecasting approaches have been developed. Conventionally, statistical techniques have been the most commonly used methods. In general, forecasting techniques are classified into two groups; *qualitative methods* and *quantitative methods*. Qualitative methods of forecasting rely on human expertise and judgement, whereas quantitative methods rely on the use of historical data [49, 81]. Broadly speaking, quantitative methods are based on two types of technique - i.e., time-series and causal models. Time-series (e.g., moving average, exponential smoothing, and Box-Jenkins) are considered powerful tools in forecasting and are used widely in a range of different applications [26, 89]. Causal models (e.g., regression and econometric models) have been used for solving complex prediction problems using the methodology of cause and effect and influencing factors [80]. However, dealing with complex models that may contain non-linearity in its nature represents a significant challenge. There is evidence to show that the classical forecasting methods may not provide the best-desired performance when predicting with these type of models. Therefore, finding alternative solutions has become a high demand. Reviewing the literature shows that artificial intelligence has been adopted as an effective forecasting technique for many models. In this section, we intend to introduce and discuss the use of ANN and ANFIS as prediction models as an alternative to the traditional forecasting methods.

2.1.1 Artificial Intelligence in Forecasting

There are abundant studies in the literature claiming that Artificial Intelligence techniques, such as artificial neural networks (ANN), can be more effective than conventional forecasting

methods for a large number of diverse application areas. Recently, Pao [103] adopted an ANN and multiple linear regression models for analyzing the determinants of capital structures of the conventional and high-tech manufacturing in Taiwan. He indicated that ANNs produced the lowest forecasting error (i.e. Root Mean Square Error - RMSE) and a better fit than the multiple linear regression model; mainly when dealing with non-linear models. Mitrea *et al.* [87] investigated the performance of traditional forecasting techniques for the inventory management problem. They compared the Moving Average (MA), and Autoregressive Integrated Moving Average (ARIMA) with two ANN approaches represented by a Feed-forward NN and Non-linear Autoregressive Network containing exogenous inputs (NARX). The results showed that the ANN prediction was more accurate than both the (MA) and (ARIMA) approaches. Noori *et al.* [100] compared an ANN model to multivariate linear regression models as a forecasting method for the river stream-flow problem. They found that the ANN model offers a better prediction performance than the MLR model.

In the following year, Gosasang *et al.* [49] employed the Multi-layer Perceptron (MLP) neural network technique and Linear Regression as a prediction model for the containerization problem. They concluded that the MLP technique produced more accurate forecasting results than using linear regression. Yip *et al.* [154] presented a comparison of two forecasting techniques, the General Regression Neural Network (GRNN) and the traditional Box-Jenkins time series models, to predict the cost of equipment's maintenance in the construction field. The results showed that the use of (GRNN) made significant improvements in forecasting results compared with traditional Box-Jenkins time series models. In 2017, Laptev *et al.* [72] utilized an end-to-end neural network to forecast the Uber trips completion time and its effectiveness on reducing the waiting time of the riders. The researchers demonstrated that ANN forecasting models could produce better results than some classical methods if the number, length, and correlation of the time-series under consideration is high.

In 2019, further studies compared the multiple linear regression (MLR) to the ANN as prediction approaches. Abdipour *et al.* [1] evaluated the performance of five ANN models along with the MLR model as a seed production predictor. They found that the MLP neural network provided the best results out of the five ANNs tested and outperformed the multiple linear regression model. This study concluded that multiple linear regression failed in explaining the non-linearity of the problem. In contrast, the MLP can overcome this problem and provide a better prediction model. In another study, Kadam [64] used the MLR and ANN as forecasting models of the groundwater quality and its suitability for drinking. The results showed that the accuracy of prediction of the ANN model was higher than the MLR model. Later, Matyjaszek *et al.* [84] investigated the performance of three conventional forecasting models, i.e., time series, ARIMA, and ROBUST models (robust regression and

robust multivariate analysis) versus two ANN models, i.e., multi-layer feed-forward network (MLFN) and GRNN. The study was conducted on the prediction of the price in financial markets. The researchers indicated that applying these models on the full time-series showed that the GRNN outperformed the traditional forecasting models. In contrast, using transgenic time series showed that ARIMA offers satisfactory forecasting performance when compared to the other models.

2.1.2 ANFIS as a Forecasting Model

There is a class of problems, for which data can be uncertain. Amongst all the artificial intelligence techniques, a hybrid system such as neuro-fuzzy has the potential to give better performance in forecasting compared to standard ANNs and other conventional methods. A neuro-fuzzy system is a combination of an ANN and a fuzzy inference system (FIS); therefore, it has the advantages of both methods [33]. Adaptive Neuro-Fuzzy Inference Systems (ANFIS) can be considered as one of the most superior intelligence and effective prediction techniques, which is capable of dealing with fuzziness, complexity, uncertainty, non-linearity and ambiguity - particularly when high precision and reliability in prediction is required [92]. In this section, a large and growing body of literature has been investigated to give a comprehensive knowledge of ANFIS models and the range of problems on which they have been applied.

2.1.2.1 The Performance of ANFIS in Forecasting

Compared to the ANN and other traditional forecasting methods, ANFIS has been shown to offer better prediction performance in most cases. Efendigil *et al.* [40] developed a comparison of prediction mechanisms for analyzing the effectiveness of using ANN and ANFIS approaches in dealing with real-world fuzzy demand forecasting problems, as part of a multi-level supply chain process. The results showed that ANFIS provided better estimation and outperformed the ANN in forecasting accuracy. Azadeh and colleagues [15] compared the use of conventional time series approaches and artificial intelligent approaches, such as ANN and ANFIS, in solving the short-term natural gas demand problem. The overall results showed that ANFIS provided significant improvements and more accurate outcomes over ANN and standard time-series prediction. Later, Lohani *et al.* [75] investigated the ability of auto-regression (AR) ANNs and ANFIS in providing better prediction of the hydrological time series modelling for the river flow problem. The results showed that the ANFIS model performed more accurately than both AR and ANN approaches in predicting the extreme river inflow. Mahdavi and Khademi [77] forecasted oil consumption in Canada. The researchers

undertook a comparative analysis of ANFIS against AR models. The results showed that ANFIS provides significantly better estimation accuracy when compared with AR.

In 2016, Khademi *et al.* [68] used three different data-driven models; ANNs, ANFIS, and multiple linear regression in order to forecast the compressive strength of recycled aggregate concrete. The researchers found that ANFIS was more efficient than the ANN and that both outperform the multiple linear regression approach. Yaïci and Entchev [148], compared the prediction performance of ANFIS and ANNs in the solar thermal energy system. The outcomes indicated that ANFIS had provided the highest accuracy and better reliability than ANNs. However, in term of the efficiency of the processing speed and implementation, the ANNs showed more flexibility than ANFIS.

In 2018, Mashaly *et al.* [83] concluded that ANFIS gave better forecasts than multiple linear regression in predicting solar productivity. In another study in 2018, Aengchuan *et al.* [3] investigated the prediction performance of both ANFIS and ANNs for uncertain supply and demand as part of an inventory system. The results showed that ANFIS provided a better fit and achieved the best performance when compared to the ANN model. Later, Okwu and Adetunji [101] adopted the ANN and ANFIS models in solving the trans-shipment problem. The research aimed to model and optimize the distribution costs in a multi-level trans-shipment system. The results showed that the use of ANN and ANFIS reduced the optimal total cost of distribution by 36% and 34%, respectively, when compared to the classical model.

In 2020, Wong *et al.* [146], estimated the efficiency of biochar adsorption for the Cu (II) ions removal within the water. They compared and evaluated the performance of the ANN, ANFIS and MLR as prediction models. They employed eleven various algorithms to train the ANN models, and eight different MFs as knowledge base for ANFIS models. The outcomes showed that both the ANN and ANFIS outperformed the MLR significantly. However, the researchers concluded that ANFIS was found to provide the best performance. Following that, Nanda *et al.* [95], adopted the ANN and ANFIS to estimate the fundamental vibration frequencies that can be used to determine the fractures of the crosswise fixed shaft (construction problem). The outcomes showed that the ANFIS model outperformed ANN with an evident rate of average error value by 1.33%.

Despite prior evidence, some studies have claimed that ANNs provides better estimation than ANFIS, albeit findings are somewhat contradictory. As a matter of fact, each problem has its structure, such as properties, characteristics, and hypotheses. Therefore, finding the right solution methods that are suitable for solving fuzzy problems, and applying them appropriately, plays an important role when comparing these methods to each other. The

literature showed that comparing the ANFIS models with ANN was incommensurate in particular problems due to some determinants.

A study by Amid and Mesri [10] measured the prediction performance of the linear regression, radial basis function neural networks (RBFNN), MLP, and ANFIS for broiler production problem. The results showed that the RBFNN had provided the best prediction performance, followed by ANFIS, linear regression, then MLP. Technically, the body structure of ANFIS network determinant to have only one output variable that can be processed in a single ANFIS model. However, in this study, Amid and Mesri proposed their ANFIS model to include three input variables and **two output variables**. From our viewpoint, the use of ANFSI as an estimation method for this type of problem was not felicitous. A Co-active Neuro-fuzzy (CANFIS) can be a better choice as a solution method for this type of problem [61]. Consequently, comparing the results of ANFIS with ANN may not be right for this particular problem.

In another study, Parvizi *et al.* [104], modelled a natural gas prediction problem using ANN and ANFIS. Comparing the results showed that the ANN model was marginally better than ANFIS. However, the researchers have limited the knowledge base of the ANFIS model to use six Gaussian MFs for their proposed input variables. In our view, we would suggest solving ANFIS using more types of MFs which may lead to better results. Using various types of MFs will produce a wider ANFIS's knowledge base, which can provide more flexibility for the fuzzification process when dealing with fuzzy data. Therefore, the performance of ANFIS model might be improved by using more MF types. Consequently, the comparison of the two models' results will be changed.

From our perspective, comparing ANFIS performance with other prediction methods needs to be built on using valid hypotheses and correct assumptions, as well as different types and numbers of MFs in order to obtain the best solution.

2.1.2.2 ANFIS in Different Applications

Neuro-fuzzy systems have been used widely in many applications and various fields such as technical diagnostics and measurement [137], business [107], educational fields, medical systems, economic systems, traffic control, forecasting and prediction [57], electrical and electronics systems [16] [113], manufacturing and system modelling, and so forth [65][66]. More recent studies have confirmed that ANFIS has been vastly used in different applications. For instance, Table 2.1 shows the use of ANFIS in various areas during the last decade.

Forecasting demand and consumption plays a vital role in an aggregate production planning process [93]. In the last decade, ANFIS has been successfully adopted as a prediction technique in this area. There are many studies professing the use of ANFIS and

Table 2.1 Several instances of ANFIS applications

Year	Authors	Domain	Methodology
2010	Boyacioglu and Avci [27]	Stock market return prediction	ANFIS
	Talei *et al.* [133]	Rainfall–runoff prediction modeling	ANFIS
	Firat and Güngör [45]	Sediment level prediction	ANFIS, ANN, MLR
2011	Alizadeh *et al.* [7]	Stock portfolio return prediction	ANFIS
2012	SaberIraji *et al.* [112]	Students academic performance prediction	ANFIS, LVQ
	Al-Hmouz *et al.* [6]	Mobile learning adaptation	ANFIS
	Najah *et al.* [94]	Water quality prediction	ANFIS, Wavelet de-noising
2013	Mahdavi and Khademi [77]	Oil production forecasting	ANFIS, AR, Data mining
	Hosseinpour *et al.* [55]	Road accident prediction	ANFIS
	Svalina *et al.* [127]	Stock Exchange prediction	ANFIS
	Talei *et al.* [132]	Rainfall–runoff prediction modeling	ANFIS, DENFIS
2014	Khoshnevisan *et al.* [69]	Agricultural, potato yield prediction	ANFIS, ANN
	Chen and Do [34]	Students academic performance prediction	HANFIS, GA
	Emamgholizadeh *et al.* [42]	Groundwater level prediction	ANFIS, ANN
2015	Wang and Ning [141]	Bank cash flow optimization	ANFIS, PSO
	Vasileva-Stojanovska *et al.* [136]	Educational quality of experience prediction	ANFIS
	Dragomir *et al.* [41]	Renewable energy performance	ANFIS, ANN
	Wen *et al.* [144]	Groundwater level prediction	WANFIS
2016	Hsu [56]	E-Commerce cash flow	ANFIS
	Yaïci and Entchev [148]	Solar thermal energy prediction	ANFIS, ANN
	Su and Cheng [124]	Stock forecasting	ANFIS
	Mahmud and Meesad [78]	Stock market price prediction	RENFSM, ANFIS
	Mekanik *et al.* [85]	Seasonal rainfall forecasting	ANFIS, ANN
	Atsalakis *et al.* [14]	Stock market forecasting	ANFIS
2017	Đokic and Jović [167]	GDP health and growth analysis	ANFIS
	Yaseen *et al.* [152]	Stream flow forecasting	ANFIS
	Rezakazemi *et al.* [109]	Hydrogen separation evaluation	ANFIS, GA, PSO
	Amid and Mesri [10]	Broiler production predictive models	ANFIS, MLP, RBF
2018	Mashaly *et al.* [83]	Solar energy prediction	ANFIS
	Yaseen *et al.* [153]	Rainfall forecasting	ANFIS
	Stojcic *et al.* [123]	Queuing systems time optimization	ANFIS
	Jones *et al.* [63]	Population growth	ANFIS
	Aengchuan *et al.* [3]	Inventory control	ANFIS, FIS, ANN
2019	Bonakdari *et al.* [25]	Soil temperature	WANFIS, MLP, PSO
	Benmouiza and Cheknane [22]	Solar radiation forecasting	ANFIS
	Ahmadlou *et al.* [5]	Flood susceptibility modelling	ANFIS
	Zhou *et al.* [165]	Flood forecasting	ANFIS, GA
	Parvizi *et al.* [104]	Natural gas reforming modelling	ANFIS, ANN
2020	Sirabahenda *et al.* [119]	River's sediment concentrations and loads	ANFIS
	Ahanger *et al.* [4]	Education quality assessment	ANFIS
	Wong *et al.* [146]	Water quality	ANN, ANFIS, MLR
	Nanda *et al.* [95]	Construction	ANN, ANFIS

demonstrating its performance as a demand and consumption forecasting technique. In 2010, Azadeh *et al.* [15] employed ANFIS to predict a short-term natural gas demand. The researchers indicated that ANFIS provided better results than the ANN and the traditional time series approach. They stated that ANFIS was the right choice as an intelligent model for solving demand and consumption prediction problems. Mainly, when dealing with causal modelling that contains multiple inputs, non-linearity, uncertainty, complexity, and ambiguity in its nature. Later, Nadimi *et al.* [92] presented ANFIS as a prediction model for long-term electricity consumption. The proposed ANFIS model was combined with classical auto-regression (AR) in order to produce sufficient input data. The performance of the proposed ANFIS-AR model was compared to an ANN model. ANFIS outperformed the ANN and provided accurate results that were very close to the actual consumption values. They concluded that ANFIS algorithm is one of the superior approaches that have the capability to deal with fuzzy and complex consumption problems.

Mordjaoui and Boudjema [90], used ANFIS to estimate short-term electricity load demand as part of power system planning procedures. The outcomes showed that the ANFIS model provided high prediction accuracy when compared to other models such as ANNs. The researchers concluded that the ANFIS model has the ability to handle the rapid fluctuations in power demand, unlike ANN models. Azadeh *et al.* [17] proposed a combined long-term natural gas consumption prediction model based on ANFIS and computer simulation model (CS). The proposed model has been compared to ANN-MLP models as well as traditional regression models. The outcomes showed that ANFIS-CS model outperformed the other models and provided an applicable model with better performance. The researchers claimed that their proposed model presented a unique and flexible ANFIS model in solving gas consumption problems, which can be applied to estimate gas demand in the future.

In 2016, Yang *et al.* [150] adopted a combined prediction model which consisted of three methods, i.e., ANFIS, Back Propagation(BP) neural networks, and ARIMA model to forecast short-term electricity demand. The proposed model was working on using the ANFIS and BP model to deal with the non-linearity of the data. Whereas, the ARIMA model dealt with the linearity and seasonality. The results showed that using the proposed combined model had a high prediction accuracy. The work presented by Panapakidis & Dagoumas [102] also addresses the problem of natural gas demand. Here, they present an approach combining ANFIS with genetic algorithms, wavelet transforms and feed-forward neural networks. The results indicated that the proposed model was distinguished by a high level of flexibility and comprehensive operation. An important conclusion made by the authors is the need for robust data pre-processing to lower forecasting error.

One year later, Kaveh *et al.* [67] employed ANFIS and ANN models to predict the consumption of the drying energy for four crops as part of an agricultural problem. The structure of the proposed model consisted of four input variables (i.e., air temperature, air velocity, drying time, and produced type), and one output variable. The results of both models showed that the ANFIS model had outperformed ANN and provided high capability in evaluating all output. In 2020, Adedeji *et al.* [2] adopted ANFIS models to predict the energy consumption of a multi-campus institution. They proposed two models in order to forecast the consumption of four different campuses. First, ANFIS was used as a standalone model. The second proposed model employed particle swarm optimisation (PSO) alongside ANFIS to produce a combined hybrid prediction model. The performance of the two models was compared. Based on the overall results, the researchers concluded that the hybrid ANFIS-PSO model outperformed the straightforward ANFIS model.

2.1.2.3 Pre-Processing Data for ANFIS Problems

As defined in the previous sections, an ANFIS structure is based on two main methods, i.e., the fuzzy inference system (FIS) and artificial neural network (ANN). It takes the advantages of both approaches; the FIS describes the uncertain phenomena, and the ANN provides a self-learning ability. This means ANFIS uses a set of data samples in order to build the fuzzy knowledge base and process the input-output mappings to fine-tune the MFs at the fuzzy knowledge base. A significant concern is that the quantity and scale of the real data may not be sufficient to train a reliable model. In many cases, the collected input data can be either scarce or contain significantly different scales. Therefore, it is essential to find a robust data processing model that can help to improve the prediction performance and reduce the error measures.

Pre-/post-processing data is a well-known technique and has been widely used to improve ANFIS performance. Nadimi *et al.* [92] compared ANFIS with ANN models as prediction techniques for long-term power consumption. The structure of the proposed models consisted of two input variables (i.e., population and Gross Domestic Product (GDP)) and one output variable (i.e., electricity Consumption). The annual net consumption data of seven industrialized countries have been drawn from the world bank development indicators. Twenty-eight sets of data samples were available covering the period from 1980 to 2007. Due to the difference of data scales, the proposed ANFIS model has been equipped with pre-processing and post-processing techniques to remove any possible noise in the data. All the input and output data samples have been scaled and normalized before being fed it into ANFIS models. The researchers concluded that the use of pre and post data processing had

provided higher precision to the performance of the proposed ANFIS model. However, they did not provide any further details of the methods that had been used for data processing.

Azadeh & colleagues [15] presented ANFIS as a more superior method than ANN in short-term prediction of the natural gas demand. The proposed ANFIS model structure contained four inputs and one output. The authors adopted the daily natural gas consumption for nearly six months period in order to demonstrate the applicability and comparability of both models. They applied the Z-score technique to normalize the data over the range of [0,1]. The scaling process was used to ensure that all data entries were equal in its weight. The researchers claimed that significant improvements had been achieved when they used scaling and pre-processing to remove noise in the data set prior to training ANFIS to model demand prediction.

Najah *et al.* [94] proposed a composite forecasting model containing ANFIS and wavelet de-noising technique (WDT-ANFIS) to predict water quality parameters. The combination of these two methods was used to develop a model that can deal with the noise of data signals caused by systematic and random errors. The main goal was to enhance the prediction performance and accuracy of the water quality. A total of sixty data samples were gathered from the mainstream of Johor River for the period of 1998–2007. However, these data samples showed numerous inconsistencies in the data recorded by the relevant department. Therefore, the researchers adopted the wavelet de-noising as a pre-processing tool in order to enhance the data quality prior to being fed it into ANFIS. The results showed that the WDT had contributed effectively in improving the prediction accuracy, and the proposed composite WDT-ANFIS model outperformed the straightforward ANFIS model.

Mahdavi and Khademi [77] adopted a data mining technique as a pre-processing tool to enhance the prediction performance of oil production using an ANFIS model. They employed a data cleaning technique to enhance and integrate the data samples before feeding it to ANFIS. The authors stated that the pre-processing operation consisted of two phases. First, for integration, they take out any invalid values from the training data. Second, convert the data into static by equalising the mean and variance of the data in time duration. Although the results indicated that the proposed pre-processing model had played a part in improving ANFIS performance, this study did not provide any evident details of the methodology that had been used to process the data.

In their work, Azadeh *et al.* [17] proposed a combined prediction algorithm based on ANFIS and Computer Simulation (CS) to forecast long-term gas consumption. The structure of the model included four variables, i.e., three inputs and one output. The provided data sets were on monthly basis samples covering the period from 2000 to 2008. This work offered a new methodology of using the pre-post data processing. The pre-processing operation was

represented by applying the CS model into the historical data. First, the provided data for each year are to be examined in order to identify the best distribution function that fits it. Second, the CS model is to be applied to generate random variables for each year using the best probability distribution extracted from the first step. The outcomes of the CS model are to be fed into ANFIS model in order to estimate the long-term gas consumption. Moreover, the proposed ANFIS-CS model was compared to 12 various ANN-MLP models and 10 different types of regression models in order to evaluate the performance of the proposed model. The ANFIS-CS outperformed all the other models and offered better prediction accuracy.

Su and Cheng [124] used a hybrid stock prediction model based on the method of integrated non-linear feature selection (INFS) and ANFIS time series model. They proposed a solution algorithm which contains three main phases, i.e., data pre-processing, ANFIS modelling, and forecasting and evaluation. The first phase (pre-processing) contains two steps, firstly it works on converting the original data into technical indicators, and secondly, selecting the important indicators using the INFS method. This should produce pre-processed data that can be fed into ANFIS. The proposed model offered a specific type of pre-processing operation that can be used when the input variables need to be in the form of technical indicators. The results showed that the proposed model had provided better total profitability and accuracy than the use of explicit data.

More recently, [99] approached the problem of fine-tuning the parameters governing the shape of the ANFIS membership functions. The authors compared optimised and non-optimised ANFIS applications to the problem of the estimation of freight train energy consumption. They used a variant of the Bee Colony Optimization (BCO) algorithm, a meta-heuristic approach, for the adjustment of fuzzy logic membership functions. They concluded that the precision of the developed fuzzy reasoning model was significantly increased after tuning membership functions by the BCO.

2.1.2.4 ANFIS Models: The Data Scarcity Problem

Data scarcity represents a significant challenge nowadays. In the previous sections, several studies introduced different methodologies of pre-processing of ANFIS's entry data, such as scaling. However, a limited number of studies addressed a pre-processing solution for dealing with a small data problem. Recent evidence suggests that ANFIS can be considered as one of the preferred prediction methods in dealing with problems when the entry data are scarce. However, this is true only to some limited extent. If we refer to the Nyquist Theorem [37], which states that the more complex the underlying distribution is, more data

is needed to be able to reproduce it accurately. In other words, as the problem complexity increases, more data are needed to define it and capture that level of complexity.

Li *et al.* [73] indicated that, for building a precise and useful knowledge of any artificial learning system, a sufficient number of data samples are needed before proceeding the training process. Therefore, they proposed a comparison model to show how limited information and small data are negatively affecting the performance and accuracy of prediction. They compared the performance of traditional neural networks such as Pythia (a crisp learning approach) with the ANFIS (a fuzzy learning technique) in solving the early scheduling of the dynamic flexible manufacturing system problem. The main goal was to prove that expanding the fuzzification domain range, by adding more data, can provide a wider fuzzification area for better prediction accuracy. In other words, enlarging the training data sets will improve the fuzzification process by extending the MF's mapping area into a broader range. A set of 100 data samples were used to compare the prediction performance of both methods. The researchers examined the performance of both methods using eleven data sets of different sizes, clipped out of the original data. The smallest data set size consisted of only five samples, whereas the largest represented the entire original data set of 100 samples. The results showed that by adding more data, the performance is improved in both methods. However, the fuzzy learning model (ANFIS) had significantly outperformed the crisp learning method (Pythia). ANFIS testing accuracy increased from 79% (using the set of five data samples) to 93% (using 100 data samples), whereas using Pythia provided prediction performance of 51% (using a set of five data samples) to 78% (using 100 data samples). It is not difficult to notice that solving the ANFIS using five data samples is even better than the Pythia using 100 data sets. The authors concluded that by using fuzzy models, such as ANFIS, in modelling small data sets, learning can improve the prediction accuracy.

In a later work , Li and his colleagues [74], developed another combined model using a data trend estimation approach, mega-fuzzification, and ANFIS to overcome the rare data problem in the early scheduling of the dynamic flexible manufacturing systems. This study represents an improved version of the first model mentioned above. The results showed that the prediction accuracy had increased from 69.3% (using a set of five data samples) to 94.7% (using 100 data samples) using the proposed combined ANFIS model.

Sen *et al.* [116] examined the performance of ANFIS models and a genetic programming (GP) technique in the sampling problem, mainly when the data are rare and continuous. The k-fold cross-validation and fuzzy c-means clustering techniques were employed as a solution algorithm for these methods (i.e., ANFIS and GP). The results indicate that the fuzzy c-means is more reliable than the k-fold cross-validation for both methods, and that the ANFIS model had outperformed the GP when modelling small data. Dewan *et al.* [39] developed an ANFIS

model to predict the ultimate tensile strength of welded aluminium alloy joints. The structure of the proposed model contained three inputs and one output variable. The data, consisting of 73 samples, were obtained experimentally. The study showed that ANFIS was one of the best methods when dealing with small data problems. They found that when predictions were required using small experimental data sets, their approach yielded better results by using leave one out cross-validation with ANFIS and ANN. The results indicated that the ANFIS model offered more reliable performance and better prediction accuracy than ANN.

In another study, Barak and Sadegh [18] proposed a hybrid ARIMA-ANFIS ensemble model to predict the annual energy consumption, mainly when the data were insufficient. They presented three solution scenarios based on an ARIMA-ANFIS hybrid model as a time series forecasting technique. Initially, the ARIMA model had been used to forecast the linear part of the original data, prior processing the three scenarios. In the first scenario, ARIMA forecasts the linear data, whereas the ANFIS model is used to predict the non-linear (residuals) data. In the second scenario, the forecasting of linear data (ARIMA outputs) is used as one of the ANFIS inputs in addition to the non-linear inputs. The last scenario is similar to the second one. However, the researchers employed the AdaBoost (Adaptive Boosting) as an ensemble method to enhance the data as a proposed solution method to deal with the rarity of the data. The overall results indicated that the third scenario provided better prediction than the other models, as the MSE enhanced from 0.058 to 0.026 compared to the second scenario.

In 2019, Fachini *et al.* [43] adopted ANFIS as a voltage prediction model to determine the critical bus voltage for the IEEE 14-bus system when the data is of a limited amount. They indicated that ANFIS provides a better estimation for small data problems. The researchers showed that using ANN techniques to find predictions with limited amounts of data gave poor and inaccurate results. The authors further showed that this could be overcome by using ANFIS.

Despite prior evidence, ANFIS can be used to solve problems with small data, *if and only if*, there are sufficient data that can provide reliable learning knowledge to optimize all ANFIS parameters (i.e. premise and consequent). In other words, performing ANFIS's hybrid learning algorithm requires the entry data sets to be greater than the number of total parameters. Otherwise, the modellers are advised to either fix the MFs parameters (premise), which means no further optimization for the fuzzification process will be made [61] [163]; or to obtain a reliable pre-processing data model that can expand the data samples up to a sufficient number in which to satisfy this condition.

Delving in-depth into the literature, few related kinds of research was found in which to deal with this problem. In 2009, Efendigil *et al.* [40], used the fuzzy and neural approaches

to predict consumer demands within a multi-level supply chain, particularly when it is under uncertainty. They employed ANNs and ANFIS as a bipartite methodology to propose their forecasting models. A comparison of the two proposed models indicated that ANFIS had provided better performance than ANN. The ANFIS structure of the proposed demand forecasting system has been built with four input variables (i.e., unit sale price, product quality, customer satisfaction level, and effect of promotion holidays), and one output (i.e. demand quantity). Various input knowledge bases have been employed to investigate the impact of using different types and numbers of MFs on the prediction accuracy. This structure has been applied to three retailers. Real-world data have been obtained using a questionnaire issued to retailing experts and pre-determined factors extracted from the literature. The questionnaire provided only twenty-four data samples. The collected data was in the form of a monthly basis covering a two-year period. Looking at the number of total parameters (premise and consequent) accompanying each ANFIS model, the researchers indicated that the number of data samples was insufficient. Therefore, they intended to expand the data into 96 monthly periods by generating more samples using Monte Carlo simulation. However, the researchers did not provide any evidence to show the reliability and/or robustness of the data generating model. Nor did they comment on the extent to which it can explain the characteristics of the original 24 samples. Moreover, they did not supply enough information about the original and expanded data for all input variables. Therefore, it is difficult to form an opinion of the performance of the proposed expansion model.

To the best of our knowledge, using data expansion as a pre-processing technique to overcome the data scarcity problem (specifically for ANFIS modelling) is only applied across a limited range. We did an extensive search in the literature to cover this area. However, due to the limitation that existed in the previous technique, we did not find more studies dealing with this particular problem. With this motivation, our proposed model can be considered as a novel work as it represents the first research using robust mathematical models to expand the data for ANFIS modelling.

2.2 Sparse Data and Outliers

2.2.1 Sparse Data

Sparse data provides a challenge in many applications. There is evidence were some models can deal with this problem and provide an enhanced range of data to overcome the data sparsity problem. Here we will bring into account the most recent data sparsity modelling research.

Antholzer *et al.* [12], developed an image reconstruction model formed from a deep learning convolutional neural network (CNN) algorithm to be used in photo-acoustic tomography (PAT) from sparse data. They employed a linear reconstruction algorithm to deal with the data sparsity problem prior to the implementation of the CNN on the training data using adjusted weights (the actual image reconstruction). The results showed that the proposed model provides better quality image reconstruction compared to conventional approaches for PAT from sparse data.

Beigi *et al.* [21], looked into the problem of sparse data for modelling personality in social networks, which is caused by the small percentage of negatively-signed links compared to positive links. They investigated the possibility of mitigating the data sparsity problem by obtaining personal information. The authors proposed a signed link prediction (SLP) model that allows experimental exploration of user personality through social media data. They relied on the optimism and pessimism information obtained from the user's personality in order to establish more positive and negative links. Their research aimed to investigate the possibility of obtaining personal information, and to determine if this information could help in overcoming the data sparsity problem in this area. The results showed that the performance of the SLP and all other prediction methods improved after increasing the size of the training sample.

Chen *et al.* [32], proposed a novel method named Hierarchical Bayesian Data Augmentation (HBDA) to deal with data sparsity in fatigue S-N curves. The proposed method is to be integrated with hierarchical Bayesian modelling (HBM) and Bayesian data augmentation (BDA) to create a larger fatigue life data sample from the raw sparse samples. Four solution strategies were processed to perform the proposed model. These strategies were validated and compared using data drawn from the open literature. The results showed that the HBDA (proposed method) significantly outperformed the conventional methods when compared with the HBM/BDA alone, particularly with a small data sample. The authors have evaluated the proposed model by applying it to a real-life problem where only limited data are available for testing. Other, recent publications relating to the problem of modelling with small data sets can be found in Table (2.2).

The lack of data, which is interchangeably described as scarce data or sparse data, represents one of the big challenges to accurate data-driven model building. As a solution method for this problem, data expansion via sample interpolation and approximation is reasonably well understood and covered extensively in the literature. Wagner *et al.* [139], compare and evaluate a range of regression-based interpolation methods for modelling daily rainfall in data-scarce regions. This is particularly relevant to our work. In our work, we compare and contrast a range of approaches to try to determine the most suitable model for

Table 2.2 Sparse data most recent literature

Year	Authors	Domain	Methodology
2017	Takwoingi *et al.* [131]	Meta-analysis - sparse data	Hierarchical summary receiver operating characteristic (HSROC)
2017	Yu and Baek [155]	Sparse data in wireless sensor networks	Sparse Random Sampling (SRS)
2018	Chen *et al.* [31]	Sparse data in computed tomography (CT)	Adaptive deep learning - image reconstruction
2018	Hao *et al.* [53]	Low-sample size data in genomic medicine	Pathway-Associated Sparse Deep Neural Network (PASNet)
2019	Sinha *et al.* [118]	Sparse data - time series	The Koopman operator
2020	Feng *et al.* [44]	Sparse data in recommender systems	Fusion collaborative filtering method

accurately reproducing the trend from scarce data. In their paper, the authors take a similar approach. They recognise that where there are regions over which data density is a problem, different interpolation schemes can yield different results. Their aim is to try to determine the best approach. In his research, Coppejans [36] uses piece-wise cubic splines as a tool to support a method for estimating statistical expectation. Our choice of using cubic radial basis functions to model the data shares interesting overlaps with the theory of cubic spline interpolation.

MacAllister *et al.* [76] investigated the feasibility of employing meta-models, Kriging and Gaussian RBFs, as multivariate data approximation techniques to generate synthetic data that can overcome problems due to the scarcity of training data for Bayesian networks. Their proposed model was applied to predict customer behaviour for three companies (Amazon, Apple, and Google) as well as identifying market trends. Particle Swarm Optimization (PSO) was utilized to fine-tune the network parameters for four network structures using three small data sets. The proposed models were used to increase the small data into three different generated data sets of sizes: ten thousand, one-hundred thousand, and a million data points. The results showed that using the proposed model provides increased accuracy over small sample sets in training the Bayesian networks.

Inspired by this exciting development, we have been motivated to search for a suitable mathematical solution to solve the problem of data scarcity that we faced. Despite all the previously proposed models for dealing with the data sparsity problem, we could not find a suitable one that we can be applied directly to our case study due to the fuzziness, uncertainty and fluctuation in the original data.

It is underlined that in all the published methods, limited studies discussed the use of RBF interpolation to deal with the small data problem. In later chapters, where we introduce the data interpolation and near-interpolation approach, we will refer the reader to relevant research where more information on the forms being constructed can be found.

2.2.2 Outliers (anomaly) Mitigation

Outlier mitigation has been presented in many research areas and applications in the literature. It has been discussed and developed using different solution methodologies and approaches. Here, we bring into account some of the most recent research in this field.

Pozo-Prez *et al.* [105], presented a regression framework as an effective alternative to the Receiver Autonomous Integrity Monitoring (RAIM) in GNSS signals. The proposed framework is used for mitigating the unexpected large errors which do not meet the assumption of Gaussian noise. The outcomes indicated that the proposed model provides a significant improvement compared to other conventional approaches. Kim *et al.* [70], introduced a solution algorithm that enhances the location accuracy of indoor pedestrian dead reckoning. They compensated for the location error using the magnetic field map-matching technique. The researchers employed the roughness weighting factors to mitigate the outliers using multiple magnetic sensors as a hardware tool. The results showed that the proposed model improved the performance in all indoor situations. Nikitin and Davidchack [98], introduced the intermittently non-linear filters as an effective tool for real-time mitigation of outliers and noise in electromagnetic interference (EMI). The authors provided an overview of the tools and methodology that can be used for outlier enhancement. The results showed the improvement in the signal that can be earned after applying the proposed model. Yang *et al.* [149], showed how the Graduated Non-Convexity (GNC) approach can be used in combination with the non-minimal solvers to provide a robust solution for the outliers rejection. The authors claimed that the proposed model is significantly faster than specialized solvers and it has outperformed the classical approaches and provides more accurate results than specialized local solvers.

Talking within the time series framework, the literature presented many approaches that can be used to detect outliers in different applications. For instance, in 2016, Wang *et al.* [142], proposed an online self-learning method that can discover the outliers. This model can automatically recognise outlier time series in addition to its exact position without any prior knowledge about the data. Two years later, Wang *et al.* [143], looked at the problem of outlier detection for multivariate time series and its challenges such as variable subsets, different dimensions, and scale of the subset data. The authors introduced the multivariate outliers algorithm as a solution method for the outlier detection problem. Later on, Munir *et*

al. [91], proposed a deep learning-based outlier detection model for the cyclic and seasonality outliers that may occur usually in data streaming under the time series framework. The results showed that the proposed model has outperformed the state of the art of outliers detection approaches. Along this direction, Amarbayasgalan *et al.* [9], proposed a deep learning-based unsupervised outlier detection model which can be applied to batch and real-time outliers. The outcomes indicated that the proposed model outperformed the state-of-the-art in outlier detection approaches in most cases. Geiger *et al.* [48], also proposed an unsupervised machine learning model for time-series outlier detection based on Generative Adversarial Networks. The results showed that the proposed model was effective and outperformed the baseline methods in most cases.

In summary, the literature review has shown that when it comes to dealing with scarce/sparse data and outliers, the techniques presented are often bespoke to the application being studied. This is natural, and expected, but does make transferring knowledge and understanding to other applications difficult. It is clear too, that there is a balance to be found between effectiveness of the approach and the simplicity of its implementation. A complex and difficult to implement method is unlikely to be adopted in practice, regardless of how effective it might be. A lesson that we have clearly learnt from the literature study is that effective, robust, implementable methods are needed that generalise well to a wide range of applications.

PART 2

Conceptual Framework

Chapter 3

Fuzzy Logic & Fuzzy Set Theory

3.1 Introduction to Fuzzy Logic

Fuzzy logic (FL) can be defined as a precise logic of ambiguity and approximate reasoning. It can be considered as an endeavour of formalization/mechanization of human capabilities. As humans, we tend to learn knowledge from experiencing the world in which we live. We have a limit in our abilities to understand the world and to find reasoning. However, we use our capabilities to make reasonable decisions within an environment of ambiguity, uncertainty, lack of/conflicting information, as well as inaccuracy of measurements [162]. In other words, we use our capabilities of reasoning to make order within the heap of information (i.e., formulating human knowledge in a systematic method). The other factor which can limit our desire for exactness is the natural language that we use for sharing/describing knowledge, ideas, information, and so forth. We perceive the essential meanings of words which gives the ability to communicate precisely to an acceptable level. However, in general, we often cannot accurately agree among ourselves on a commonsense meaning for one single word or term. This leads us to the fact that language is often ambiguous [120]. Essentially, our comprehension of the real world requires us to interpret and combine many concepts (facts) that do not have clearly defined boundaries. For instance, "small", "short", "very large", "old", ... etc. These concepts can be considered as subjective terms that are true to some extent but can also be false to another extent. Consequently, a human brain can interpret the meaning of them, whereas computers might not (because it is using fixed rules and logic); therefore, these concepts can be regarded as being fuzzy.

It is widely understood that computers must strive to represent and reason with fuzzy knowledge about the real world in ways that are similar to the human brain. Lotfi Zadeh [160], took the challenge over the crisp set theory by introducing the concept of the fuzzy set as a mathematical tool for dealing with the fuzzy logic and uncertainties (such as vagueness, imprecision) problems. Before 1965, probability theory was used as a useful tool for dealing with uncertainties. The probability theory mechanism works based on two-valued logic, which represents the concepts of classical or crisp set theory [106]. Whilst, Prof. Zadeh argued that there are some uncertainties, which cannot be tackled using the probability theory because it can only handle one out of the different types of possible uncertainties. Moreover, the classical (crisp) sets describe events that either happens or not; it measures the possibility for a given event and the expectation to occur (or not) by using probability theory. Conversely, the fuzzy logic theory affords a mechanism to interpret any ambiguity that may be built in the linguistic body, such as few, many, short, tall, ... and so forth. It gives an inference structure that simulates human logical abilities. The Figure (3.1) represents the fuzzy logic system (FLS) that is processing the inputs (vague, imprecise) to provide decisions.

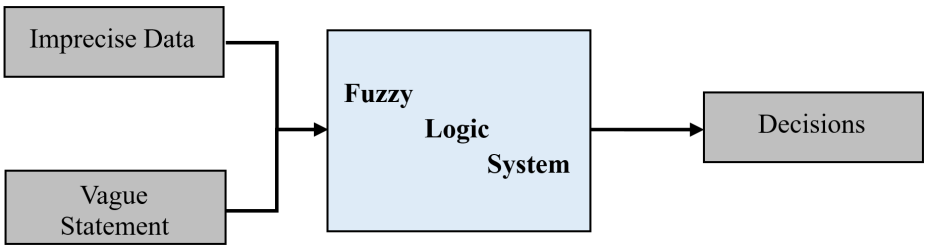


Fig. 3.1 A fuzzy logic system that takes vague statements and imprecise data such as short, medium, tall and provides decisions.

In real-world problems, complexity can be measured by the degree of uncertainty. As uncertainty rises, so does the complexity of the problem; which explains the reason why real problems are often very complex. There is an intimate relationship between complexity and fuzziness. As the complexity of a system surpasses a certain threshold, without a doubt, it should become fuzzy [120].

In this chapter, all the figures, diagrams, and examples have been created by the author, except where indicated with the word "reproduced".

3.2 Classical Sets (Crisp Sets)

The classical set is defined as the set of crisp events with definite boundaries, which means there is a certainty of the events that either do or do not occur. In other words, an individual entity is either to be a member (or not) of the set [30]. From another view, it means the membership of crisp (classical) sets is without ambiguity. Assume the set X is a particular **Universal Set** under consideration, which is composed of all possible individual elements x (also known as members) which are related to some specific context. For example, if we consider the integers 1 to 100, then all the integer numbers in the interval $[1-100]$ will be represented by our universal set X. Let us try to find the set A which represents the numbers that can be wholly divided by five in the universe X. To obtain this set, we check each integer number in the universal set in order to find out whether it belongs to A or not. Consequently, there is a specified and well-defined boundary separating the elements in the set A from the elements lying outside it.

$$A = \{x_1, x_2, \ldots, x_n\}. \tag{3.1}$$

Or, it can be represented by its element property as:

$$A = \{x | P(x)\}, \tag{3.2}$$

which means set A contains all values x that have the property P in the universe X.
The set A can be represented by the Characteristic Function (CF) as follows:

$$CF = y_A(x) = \begin{cases} 1, & \text{if } x \text{ is a member of } A, \\ 0, & \text{if } x \text{ is not a member of } A. \end{cases} \tag{3.3}$$

Where, $y_A(x)$ has any of the two values 1 (true) or 0 (false).
This Characteristic Function $y_A(x)$ represents the membership mapping in set A for the element x in the universe X as shown in Figure (3.2)

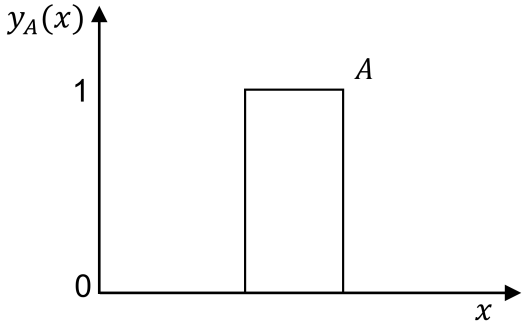


Fig. 3.2 Membership mapping for Crisp Set A

3.3 Fuzzy Sets

The fuzzy set can be defined as an adequate theory in dealing with the concept of ambiguity. It makes it possible to use quantitative methods to deal with the fuzziness. The fuzzy sets based on the concept of membership graded relatively in defining the sets [120]. In comparison to the classical set, a fuzzy set provides a method for modelling the uncertainty related with the vagueness, imprecision, and lack of information. The real-world problems are often complex and uncertain, as its complexity increases, the fuzzy logic becomes the best way to solve it [161]. Zadeh [160] generalized the assumption of a crisp set from (definitely in / out) to the interval of real values that can be called **membership degrees** (MD), using the concept of the **membership function** (MF) which expresses to what degree an entity can be judged to be part of a set. In other words, a fuzzy set is a set without a crisp boundary, which means the elements in the fuzzy set can have differing degrees of membership within the set. This means that the element can be a member of the fuzzy set according to its membership value. That is, the transition from "member of a set" to "not member of a set" is progressive and

characterized by a membership function that maps every individual element within the fuzzy set.

Definition: 3.1 *Membership Function (MF)*
The membership function is considered as the unique method to characterize and represent the fuzziness of the fuzzy set [117]. It associates each element in the universe with its membership value (degree of membership) in a particular fuzzy set. The mapping interval of the fuzzy set is to be real-numbered values within [0, 1]. Suppose that we have the fuzzy set A in the universe X, and $\mu_A(x)$ represents the degrees of membership of x related to each element in A within the universe X within the interval [0, 1] [117]. Thus, the fuzzy set A can be represented by its membership function which consists of two terms, first is the element x, and second is its degree of membership (membership value) $\mu_A(x)$. It can be denoted by:

$$A(x) = \{(x, \mu_A(x)) \mid x \in X\} \tag{3.4}$$

The previous expression (3.4) can be viewed as a simple form for representing the membership function of the fuzzy set. There is no steady rule for specifying the form of a membership function; usually, it depends on the problem type and nature. The membership function will be discussed in more details in the next sections.

Example 3.1
Assume that a fuzzy set A contains five elements $A = \{x_1, x_2, x_3, x_4, x_5\}$ characterized by $\mu_A(x)$ that maps every individual element x in the universe X to the values (0.3, 1.0, 0.6, 0.5 and 0.2) respectively. this fuzzy set can be represented as follows:

$$A(x) = \{(x_1, 0.3),\ (x_2, 1.0),\ (x_3, 0.6),\ (x_4, 0.5),\ (x_5, 0.2)\},$$

figure (3.3) shows the mapping for the above fuzzy set A.

3.3.1 Types of Fuzzy Sets

According to the nature of the universe, the fuzzy set can be either discrete or continuous. It can be denoted as follows:

- **Discrete Fuzzy Set:** Consider the universal set X that contains discrete objects, any fuzzy set within this universe will be a discrete fuzzy set. It can be formulated as:

$$A(x) = \sum_{i=1}^{n} \mu_A(x_i)/x_i,\ x \in X, \tag{3.5}$$

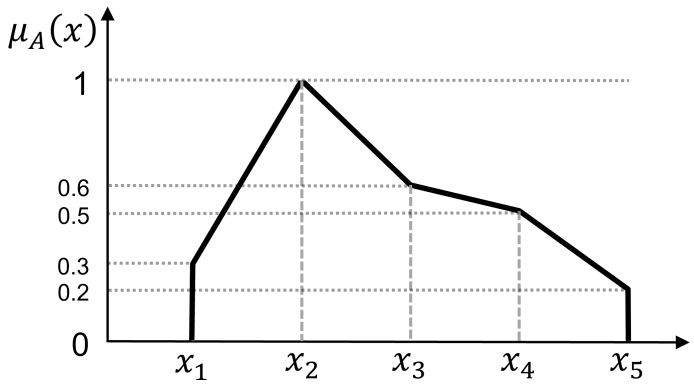


Fig. 3.3 Fuzzy set.

where n is the number of elements that belongs to the fuzzy set A.

- **Continuous Fuzzy Set:** Consider the universal set X that contains continuous objects, then the mathematical form of any fuzzy sets within this universe can be formulated as:

$$A(x) = \int_X \mu_A(x)/x \tag{3.6}$$

The symbols $\sum$ and $\int$ in the previous equations 3.5 and 3.6 does not indicate the mathematical operations of the summation or integration. However, they are utilized to point out the pairs of (element, membership value) that illustrate the discrete and continuous fuzzy sets [61].

3.3.2 Operations of Fuzzy Sets

Assume the universe X contains two fuzzy sets A and B. The following operations and relations can be defined:

1. **Union of fuzzy sets:** The union of two fuzzy sets is composed of all the elements in the universe that can appear in either set $A(x)$ or set $B(x)$ or in both, simultaneously (refer to Fig. 3.4a). Its membership function can be expressed by $\mu_{A\cup B}(x)$ as follows:

$$\mu_{A\cup B}(x) = \mu_A(x) \vee \mu_B(x), \tag{3.7}$$

where $\vee$ refers to the maximum operator i.e.:

$$\mu_{A\cup B}(x) = max\{\mu_A(x), \mu_B(x)\}.$$

Example 3.2
Suppose we have the following fuzzy sets:

$$A(x) = \{(x_1, 0.1),\ (x_2, 0.2),\ (x_3, 0.3),\ (x_4, 0.4),\ (x_5, 0.5)\}$$
$$B(x) = \{(x_1, 0.5),\ (x_2, 0.6),\ (x_3, 0.7),\ (x_4, 0.8),\ (x_5, 0.9)\}$$

Then:

$$\begin{aligned}\mu_{A\cup B}(x) &= max\{\mu_A(x),\ \mu_B(x)\} \\ &= \{(x_1, 0.5),\ (x_2, 0.6),\ (x_3, 0.7),\ (x_4, 0.8),\ (x_5, 0.9)\}.\end{aligned}$$

2. **Intersection of fuzzy sets:** The intersection of two fuzzy sets represents all the elements in the universe that belongs to both sets $A(x)$ and $B(x)$ simultaneously (refer to Fig. 3.4b). It can be denoted by $\mu_{A\cap B}(x)$ as follows:

$$\mu_{A\cap B}(x) = \mu_A(x)\ \wedge\ \mu_B(x) \tag{3.8}$$

where $\wedge$ refer to the minimum operator, i.e.:

$$\mu_{A\cap B}(x) = min\{\mu_A(x),\ \mu_B(x)\}.$$

Referring to the previous example (2), the intersection of the two fuzzy sets $A(x)$ and $B(x)$ can be given as:

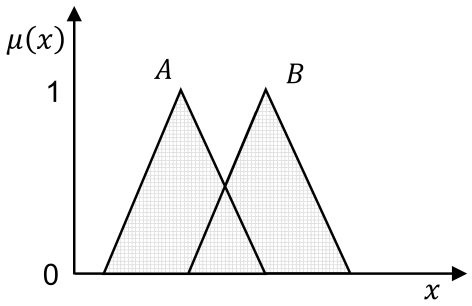
$$\begin{aligned}\mu_{A\cap B}(x) &= min\{\mu_A(x),\ \mu_B(x)\} \\ &= \{(x_1, 0.1),\ (x_2, 0.2),\ (x_3, 0.3),\ (x_4, 0.4),\ (x_5, 0.5)\}.\end{aligned}$$

3. **Complement of fuzzy sets:** The complement of fuzzy sets signifies the collection of all elements in the universe that do not belong to the fuzzy set $A(x)$ (refer to Fig. 3.4c). It can be expressed by $\overline{A}(x)$ as follows:

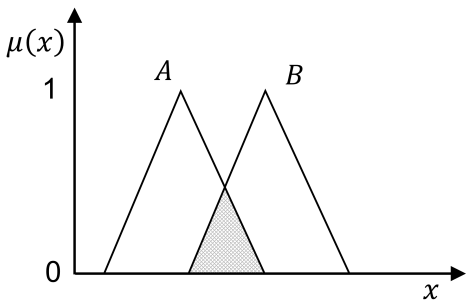
$$\overline{A}(x) = 1 - A(x),\ \text{for all } x \in X, \tag{3.9}$$

Referring to the previous example (2), the complement of the fuzzy sets $A(x)$ can be given as:

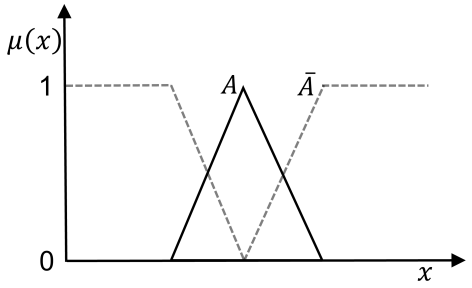
$$\overline{A}(x) = \{(x_1, 0.9),\ (x_2, 0.8),\ (x_3, 0.7),\ (x_4, 0.6),\ (x_5, 0.5)\}.$$



(a) Union of two fuzzy sets.



(b) Intersection of two fuzzy sets.



(c) Complement of a fuzzy set.

Fig. 3.4 Fuzzy set operations.

4. **Subset of fuzzy sets:** The fuzzy set $A(x)$ is a subset of $B(x)$, if the membership value $\mu_A(x)$ of each element in $A(x)$ is less than $\mu_B(x)$ of the corresponding x in $B(x)$. It can be signified as follows:

$$A(x) \subset B(x),\ if\ \mu_A(x)\ <\ \mu_B(x), \tag{3.10}$$

Referring to the previous example 2: $A(x) \subset B(x)$, because $\mu_A(x)\ <\ \mu_B(x)$ for all $x \in X$.

5. **Equality of fuzzy sets:** Two fuzzy sets $A(x)$ and $B(x)$ are to be equal, if and only if all $\mu_A(x)$ are equal to all its corresponding $\mu_B(x)$. It can be signified as follows:

$$A(x) = B(x),\ if\ \mu_A(x)\ =\ \mu_B(x), \tag{3.11}$$

Referring to the previous example 2: $A(x) \neq B(x)$, because $\mu_A(x) \neq \mu_B(x)$ for different $x \in X$.

6. **Algebraic product of fuzzy sets:** The algebraic product of two fuzzy sets $A(x)$ and $B(x)$ results from the product of the membership value $\mu_A(x)$ of each element in $A(x)$ by the $\mu_B(x)$ of the corresponding x in $B(x)$.It can be denoted by $A(x).B(x)$ as follows:

$$A(x).B(x) = \{(x,\ \mu_A(x).\mu_B(x)),\ x \in X\}, \tag{3.12}$$

Referring to the previous example 2, the algebraic product can be given as:

$$A(x).B(x) = \{(x_1,0.05),\ (x_2,0.12),\ (x_3,0.21),\ (x_4,0.32),\ (x_5,0.45)\}.$$

7. **Multiplication of a fuzzy set by a crisp number:** Assume that we have a crisp number a, the multiplication of a fuzzy set $A(x)$ by this crisp number will provide a new fuzzy set resulting from multiplying a by the membership value $\mu_A(x)$ of each element in $A(x)$. It can be expressed by $a.A(x)$ as follows:

$$a.A(x) = \{(x,\ a \times \mu_A(x)),\ x \in X\} \tag{3.13}$$

Let us consider our previous example (2), if we multiply the first fuzzy set $A(x)$ by $a = 0.3$, the result will be as follows:

$$a.A(x) = \{(x_1,0.03),\ (x_2,0.06),\ (x_3,0.09),\ (x_4,0.12),\ (x_5,0.15)\}.$$

8. **Power of fuzzy set:** Raising the fuzzy set $A(x)$ up to the p-th power, i.e. $A^p(x)$ will provide a new fuzzy set resulting from raising the membership value $\mu_A(x)$ of each element in $A(x)$ up to the p-th power, i.e. $\mu_A(x)\}^p$. Power of fuzzy set can be expressed by $A^p(x)$ as follows:

$$A^p(x) = \{(x,\ \{\mu_A(x)\}^p),\ x \in X\} \tag{3.14}$$

Referring to the previous example (2), if we raise the first fuzzy set $A(x)$ to the second power (i.e. $p = 2$) the result will be as follows:

$$A^2(x) = \{(x_1,0.01),\ (x_2,0.04),\ (x_3,0.09),\ (x_4,0.16),\ (x_5,0.25)\}.$$

9. **Arithmetic summation of fuzzy sets:** The arithmetic summation of two fuzzy sets $A(x)$ and $B(x)$ can be defined as follows:

$$A(x)+B(x)=\{(x,\ \mu_{A+B}(x)),\ x\in X\}, \tag{3.15}$$

where $\mu_{A+B}(x)=\mu_A(x)+\mu_B(x)-\mu_A(x).\mu_B(x)$.
Let us consider our previous example (2),if we add the two fuzzy sets $A(x)$ and $B(x)$ using arithmetic summation, the result will be as follows:

$$A(x)+B(x)=\{(x_1,0.55),\ (x_2,0.68),\ (x_3,0.79),\ (x_4,0.88),\ (x_5,0.95)\}.$$

10. **Bounded summation of fuzzy sets:** The bounded summation of two fuzzy sets $A(x)$ and $B(x)$ can be defined as follows:

$$A(x)\oplus B(x)=\{(x,\ \mu_{A\oplus B}(x)),\ x\in X\}, \tag{3.16}$$

where $\mu_{A\oplus B}(x)=min\{1,\ \mu_A(x)+\mu_B(x)\}$.
Referring to the previous example (2), if we add the two fuzzy sets $A(x)$ and $B(x)$ using bounded summation, the result will be as follows:

$$A(x)\oplus B(x)=\{(x_1,0.6),\ (x_2,0.8),\ (x_3,1.0),\ (x_4,1.0),\ (x_5,1.0)\}.$$

11. **Arithmetic difference of fuzzy sets:** The arithmetic difference of two fuzzy sets $A(x)$ and $B(x)$ can be defined as follows:

$$A(x)-B(x)=\{(x,\ \mu_{A-B}(x)),\ x\in X\}, \tag{3.17}$$

where $\mu_{A-B}(x)=\mu_{A\cap\overline{B}}(x)$.
Referring to the previous example (2), if we subtract the two fuzzy sets $A(x)$ and $B(x)$ using arithmetic difference, the result will be as follows:

Now, $\overline{B}(x)=\{(x_1,0.5),\ (x_2,0.4),\ (x_3,0.3),\ (x_4,0.2),\ (x_5,0.1)\}$
Therefore, $A(x)-B(x)=\{(x_1,0.1),\ (x_2,0.2),\ (x_3,0.3),\ (x_4,0.2),\ (x_5,0.1)\}$

12. **Bounded difference of fuzzy sets:** The bounded difference of two fuzzy sets $A(x)$ and $B(x)$ can be defined as follows:

$$A(x)\ominus B(x)=\{(x,\ \mu_{A\ominus B}(x)),\ x\in X\}, \tag{3.18}$$

where $\mu_{A\ominus B}(x) = max\{0,\ \mu_A(x) + \mu_B(x) - 1\}$.
Referring to the previous example (2), if we subtract the two fuzzy sets $A(x)$ and $B(x)$ using bounded difference, the result will be as follows:

$$A(x) \ominus B(x) = \{(x_1,0),\ (x_2,0),\ (x_3,0),\ (x_4,0.2),\ (x_5,0.4)\}.$$

13. **Cartesian product of fuzzy sets:** Consider the two universal sets X and Y, which include a pair of fuzzy sets, i.e. $A(x)$ and $B(y)$. The Cartesian product of these two fuzzy sets can be expressed as follows:

$$\mu_{A\times B}(x,y) = \min(\mu_A(x), \mu_B(y)). \tag{3.19}$$

Example 3.3
Suppose we have the following two fuzzy sets defined in the X and Y:

$$A(x) = \{(x_1,0.2),\ (x_2,0.4),\ (x_3,0.5),\ (x_4,0.7),\ (x_5,0.3)\},$$

$$B(y) = \{(y_1,0.3),\ (y_2,0.5),\ (y_3,0.4)\}.$$

Now, if we multiply the two fuzzy sets $A(x)$ and $B(y)$ using Cartesian product, the result will be as follows:
$min(\mu_A(x_1), \mu_B(y_1)) = 0.2, min(\mu_A(x_1), \mu_B(y_2)) = 0.2, min(\mu_A(x_1), \mu_B(y_3)) = 0.2,$
$min(\mu_A(x_2), \mu_B(y_1)) = 0.3, min(\mu_A(x_2), \mu_B(y_2)) = 0.4, min(\mu_A(x_2), \mu_B(y_3)) = 0.4,$
$min(\mu_A(x_3), \mu_B(y_1)) = 0.3, min(\mu_A(x_3), \mu_B(y_2)) = 0.5, min(\mu_A(x_3), \mu_B(y_3)) = 0.4,$
$min(\mu_A(x_4), \mu_B(y_1)) = 0.3, min(\mu_A(x_4), \mu_B(y_2)) = 0.5, min(\mu_A(x_4), \mu_B(y_3)) = 0.4,$
$min(\mu_A(x_5), \mu_B(y_1)) = 0.3, min(\mu_A(x_5), \mu_B(y_2)) = 0.3, min(\mu_A(x_5), \mu_B(y_3)) = 0.3.$
Thus, $A(x) \times B(y)$ will result the following matrix:

$$A \times B = \begin{bmatrix} 0.2 & 0.2 & 0.2 \\ 0.3 & 0.4 & 0.4 \\ 0.3 & 0.5 & 0.4 \\ 0.3 & 0.5 & 0.4 \\ 0.3 & 0.3 & 0.3 \end{bmatrix}$$

3.3.3 Classification of Fuzzy Sets

Suppose we have an element x belonging to the fuzzy set A with a membership value $\mu_A(x)$ within the universe X. According to the membership function representation, we can illustrate four classifications for the fuzzy set as follows:

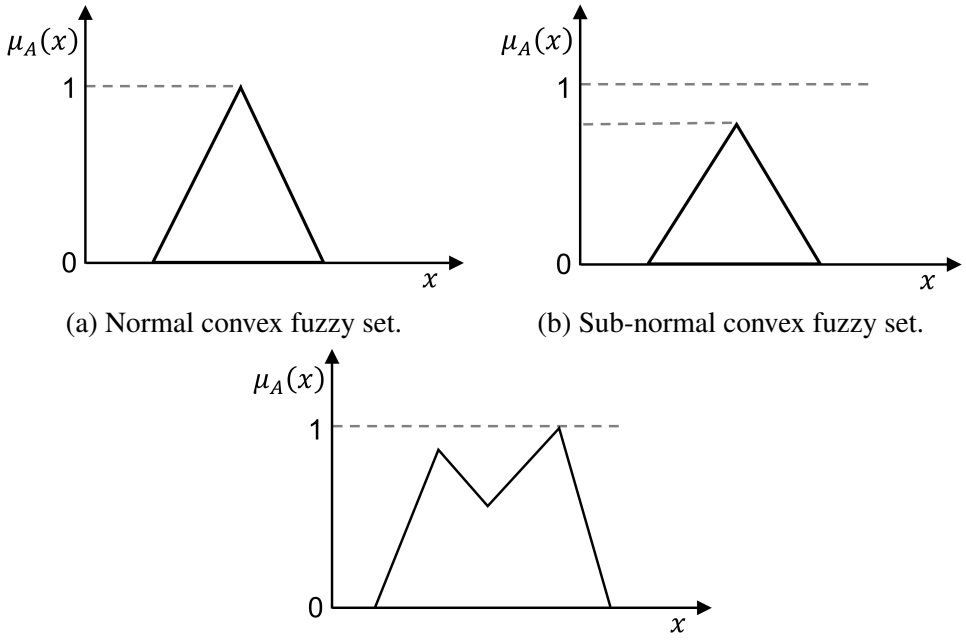
- **Normal fuzzy set:** The fuzzy set can be classified as normal if the universe contains at least one element with a membership value equal to one. In other words, the highest value (peak) of the membership function $= 1.0$, i.e., $Peak(x) = \mu_A(x) = 1.0$, (refer to Fig. 3.5a).
- **Subnormal fuzzy set:** The subnormal fuzzy set can be considered when the highest value of the membership function is less than 1.0, i.e., $Peak(x) < 1.0$, (refer to Fig. 3.5b).
- **Convex fuzzy set:** The fuzzy set can be classified as convex if the membership function contains membership values that progressively (increase and/or decrease) in simultaneity with the increasing of the elements value, (refer to Fig. 3.5a and 3.5b).
- **Non-Convex fuzzy set:** The non-convex fuzzy set can be considered when the membership function has membership values fluctuating (up and/or down) in simultaneity with the increasing of the elements value, (refer to Fig. 3.5c).

3.4 Membership Functions (MF)

As mentioned earlier in the membership function definition, the fuzzy set can be characterized by the MFs. Assume that we have the fuzzy set A within the universe X. The MF describes the relationship between each element x in A with its degree of membership $\mu_A(x)$. It quantifies the grade of membership of this particular element within the interval $[0,1]$. The MFs can be defined and expressed mathematically as well as graphically. There are different ways of formulating and parameterizing the MFs. The shape of MFs is also adopted as an essential criterion in representing the fuzzy sets. In this section, we intend to discuss various topics related to MFs.

3.4.1 Membership Function Structure

The structure of the membership function of a fuzzy set can be defined by the following regions (refer Fig. 3.6)



(a) Normal convex fuzzy set.

(b) Sub-normal convex fuzzy set.

(c) Non-convex fuzzy set.

Fig. 3.5 Classification of fuzzy sets.

- **Core**
  The core of MF for a fuzzy set A consists of all the elements $x \in X$ where the MF value is equal to 1 (i.e., $\mu_A(x) = 1.0$). Thus:

$$\text{Core}(A(x)) = \{x | \mu_A(x) = 1.0 \ , x \in X\}, \tag{3.20}$$

  A fuzzy set A can be classified as **normal** if we can find at least one element $x \in X$ with MF value equal to one ($\mu_A(x) = 1.0$). In other words, the core of the MF should have at least one value.

- **Crossover**
  The Crossover point of the MF for the fuzzy set A consists of all elements $x \in X$ where the MF value is equal to 0.5 (i.e., $\mu_A(x) = 0.5$). Thus:

$$\text{Crossover}(A(x)) = \{x | \mu_A(x) = 0.5, x \in X\}. \tag{3.21}$$

- **Boundary**
  The Boundary region of the MF for the fuzzy set A consists of all elements $x \in X$

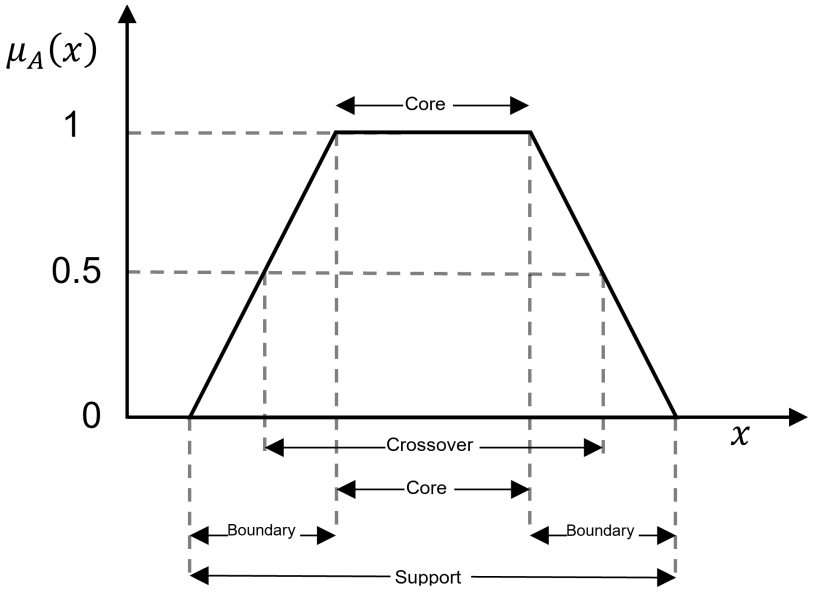


Fig. 3.6 Structure of membership function.

where the MF value is between 0 and 1 (i.e., $0 < \mu_A(x) < 1$). Thus:

$$\text{Boundary}(A(x)) = \{x | 0 < \mu_A(x) < 1, x \in X\}. \tag{3.22}$$

- **Support**
  The Support region of the MF for the fuzzy set A consists of all elements $x \in X$ where the MF value is greater than 0 (i.e., $0 < \mu_A(x) < 1$). Thus:

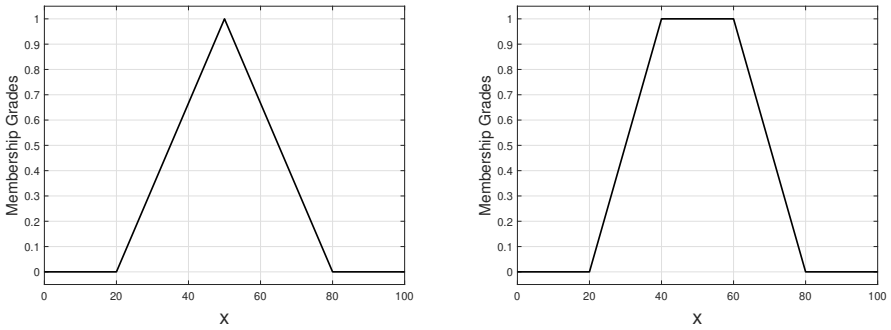
$$\text{Support}(A(x)) = \{x | \mu_A(x) > 0, x \in X\}. \tag{3.23}$$

3.4.2 Membership Function Formulation and Parameterization

The membership function can be represented both graphically and mathematically (which is considered as the most precise way). According to the number and type of the inputs variables and parameters, the MFs are classified into two categories, i.e. one or two dimensions [61]. The impacts of modifying parameters value for the one dimension's MFs and its effectiveness in fine-tuning the fuzzy inference system will be discussed in this section. As follows, we are introducing the most commonly used MFs which are classified into four types.

3.4.2.1 Piece-wise Linear Functions

These functions can be considered as the simplest MF. It can be represented by using straight lines. There are two MFs that can be classified under this type. First, is the triangular membership function, which is a collection of three points that forms a triangle (Fig. 3.7a). Whilst the second named as the trapezoidal membership function, that consists of four points resulting in a truncated triangle with a flat top (Fig. 3.7b).



(a) Triangle-Shaped MF. (b) Trapezoid-Shaped MF.

Fig. 3.7 Piece-wise linear functions.

- **Triangle Membership Function**
  The triangle membership function can be specified by a triangle curve representing a function of the x depending on three scalar parameters $\{a,b,c\}$ which represent the three edges of the triangle. The general formula of this MF can be denoted as followed:

$$\text{Triangle MF,} \quad f(x;a,b,c) = \begin{cases} 0, & x \leq a. \\ \frac{x-a}{b-a}, & a \leq x \leq b. \\ \frac{c-x}{c-b}, & b \leq x \leq c. \\ 0, & x \geq c. \end{cases} \tag{3.24}$$

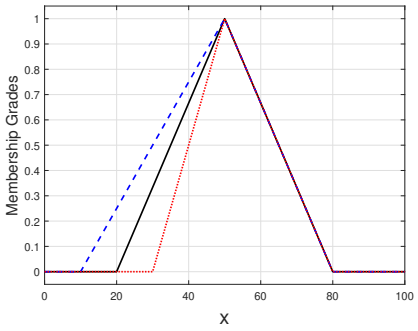
  Or, by using max-min, we can write the definition in the compact form:

$$\text{Triangle MF,} \quad f(x;a,b,c) = max\left(min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right). \tag{3.25}$$

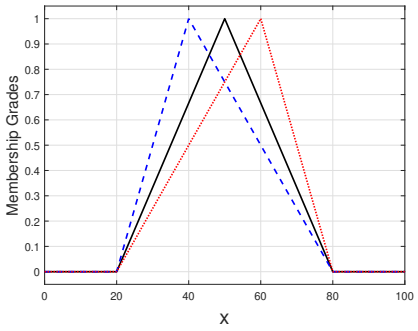
  where $a < b < c$. These parameters represent the three vertices for the x coordinates of the Triangle MF. The parameters (a,c) represent the left and right "feet"; the parameter b represent the peak of the triangle. Figure (3.7a) shows a triangle MF defined by $(x;20,50,80)$; that is $a = 20$ and $c = 80$, which represent the x coordinates

that allocated on the first and third corners of the triangle, while $b = 50$ represent the highest value of the x coordinates (the peak of triangle).

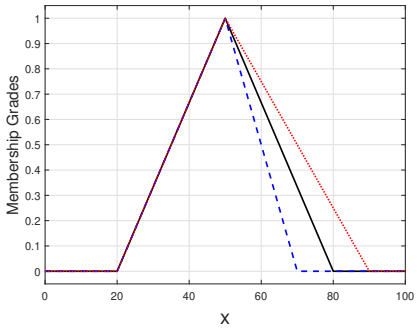
We can obtain the desired triangle MF by changing the parameter set $\{a, b, c\}$ into different values. Figure (3.8) illustrate the effects of changing the parameters sequentially. As shown in figures (3.8a, 3.8c), if we change the parameter a and c within the interval $[-10, +10]$ to modify the left or right foot values (x coordinates); the slope of the left or right side of the triangle will be changed. Moreover, if we change the value of the parameter b with the same range, the top corner (peak of triangle) and the slope of both left and right sides of the triangle will be changed simultaneously (Fig. 3.8b).



(a) Changing parameter a.



(b) Changing parameter b.



(c) Changing parameter c.

Fig. 3.8 Effects of changing Triangular MF parameters.

- **Trapezoidal Membership Function**

  The trapezoidal curve is similar (to some degree) to a triangle with a flat top (Fig. 3.7b). This means, it will have two vertices on the top instead of one, which resulting in a trapezoidal curve consisting of four points. This curve, representing the trapezoidal membership function, has four scalar parameters $\{a, b, c, d\}$ which represent the four edges of the trapezoidal shape. The general formula of this MF can be denoted as followed:

$$\text{Trapezoidal MF,} \quad f(x;a,b,c,d) = \begin{cases} 0, & x \leq a. \\ \frac{x-a}{b-a}, & a \leq x \leq b. \\ 1, & b \leq x \leq c. \\ \frac{d-x}{d-c}, & c \leq x \leq d. \\ 0, & x \geq d. \end{cases} \tag{3.26}$$

Again, by using max-min, we can write the definition in the compact form:

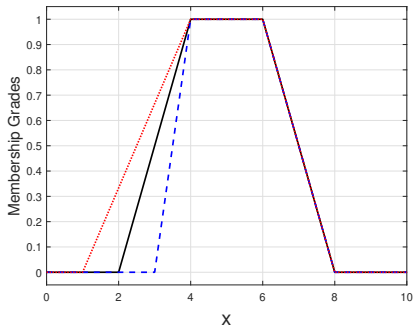
$$\text{Trapezoidal MF,} \quad f(x;a,b,c,d) = max\left(min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right). \tag{3.27}$$

where $a < b < c < d$. These parameters represent the four vertices for the x coordinates of the trapezoidal MF. The parameters (a,d) represent the left and right "feet", whilst the parameters (b,c) represent the left and right "top" of the trapezoidal MF shape. Figure (3.7b) shows a trapezoidal MF defined by $(x;20,40,60,80)$; that is $a = 20$ and $d = 80$, which represent the x coordinates allocated on the first and fourth vertices of the trapezoidal shape, while $b = 40$ and $c = 60$ represent the values of the x coordinates allocated on the second and third vertices of the shape. Figure (3.9) shows the effects of changing each of the parameters on the slope of both sides of the trapezoidal MF shape.

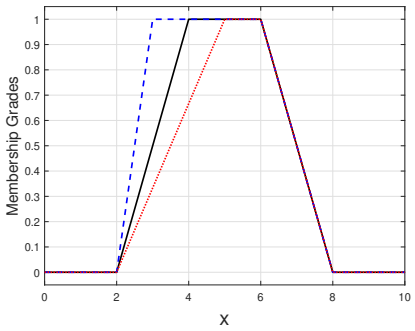
3.4.2.2 The Gaussian Distribution Functions

There are three membership functions that can be considered as Gaussian distribution functions. The first MF, called simple Gaussian MF, consists of two parameters $\{\sigma, c\}$ that represent the Gaussian curve. The second MF is the two-sided Gaussian MF which represents a composite of two different Gaussian curves. It is determined by four parameters $\{\sigma_1, c_1, \sigma_2, c_2\}$. Each pair of these parameters represents a simple Gaussian curve on one side. The third MF is the generalized bell. This MF has three parameters $\{a,b,c\}$. Figure (3.10) illustrates the plotting of the three functions. These MFs can be considered to be the most popular functions used in processing fuzzy sets. It is characterized by two major features. First, the smoothness and simplicity of its notation. Second, the nonzero values of the membership function at all points.

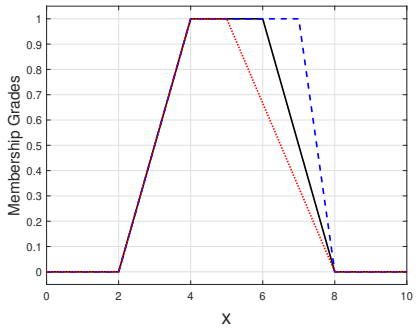
- **Simple Gaussian Curve Function**
  Two parameters specify the Gaussian MF. These parameters are σ and c, where, c determines the centroid of the curve, and σ represents the width. The plot shown



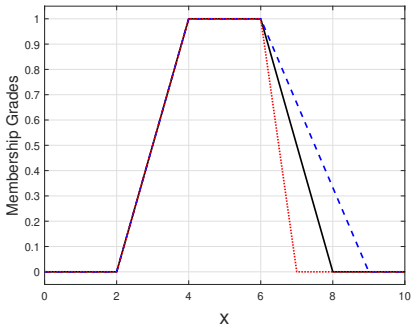
(a) Changing parameter a.



(b) Changing parameter b.



(c) Changing parameter c.



(d) Changing parameter d.

Fig. 3.9 Effects of changing Trapezoidal MF parameters.

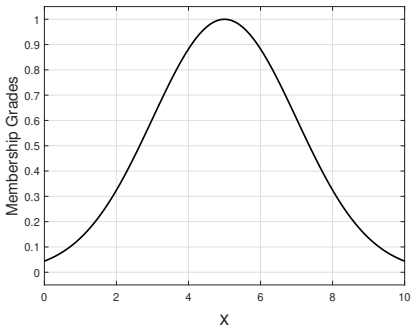
in figure (3.10a) illustrates the Gaussian MF with parameters $\sigma = 2$ and $c = 5$. The general mathematical formulation for this MF can be denoted as:

$$\text{Simple Gaussian MF,} \quad f(x;\sigma,c) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2}. \tag{3.28}$$

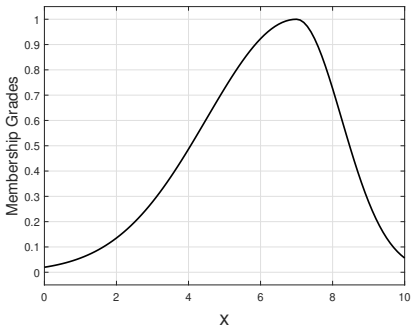
Figure (3.11) illustrates the effects of changing the Gaussian MF parameters. The first part (Fig. 3.11a) shows how the width of the Gaussian curve is varying from the centre by increasing and decreasing the value of the first parameter σ within the interval of $[1,3]$ of the original value of $\sigma = 2$. The figure (3.11b) shows the moving of the centre point of the Gaussian curve that is resulting from changing the parameter $c = 5$ within the interval of $[3,7]$.

- **Two-Sided Gaussian Curves**

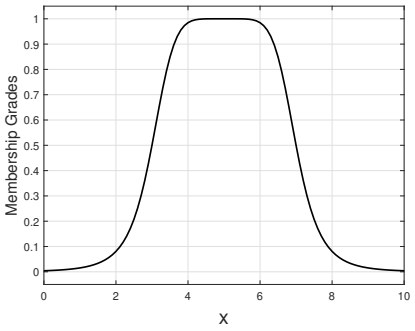
  The two-sided Gaussian MF is specified by combining two simple Gaussian MF parameters; that is $\{\sigma_1, c_1, \sigma_2, c_2\}$. The first pair σ_1 and c_1 represent the shape of the left curve of the MF, while, the second pair σ_2 and c_2 determined the right curve of the MF shape. Consequently, the two-side Gaussian MF will have two different



(a) Simple Gaussian MF.



(b) Two-sided Gaussian MF.



(c) Generalized Bell MF.

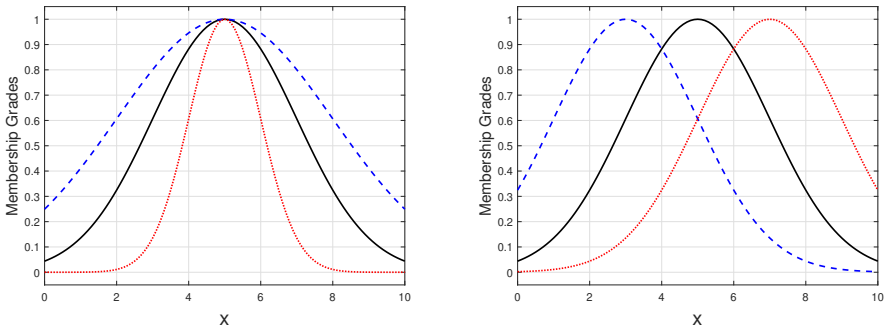
Fig. 3.10 Gaussian distribution functions.

curves from the left and right sides of the centroid of the MF shape (Fig. 3.10b). Mathematically, this MF can be denoted by:

$$\text{Two-Sided Gaussian MF,} \quad f_i(x;\sigma_i,c_i) = e^{\frac{-(x-c_i)^2}{2{\sigma_i}^2}}. \tag{3.29}$$

Where $i = 1,2$, represents the left or right MF according to its parameters that represents the two sides of the function.

- **The Generalized Bell Function**
  A generalized bell MF is determined by three parameters $\{a,b,c\}$. The parameter a represents the width of the MF around the centre point. Whereas the parameter b represents the slope of the sides around the fixed crossover point of the MF. The value of this parameter is usually positive, as a negative value will result in flipping the shape of the MF upside-down. The parameter c determines the centroid value of the MF. The plot shown in figure (3.10c) illustrates the generalized bell MF with parameters $a = 2$, $b = 3$ and $c = 5$. The general formula of this MF can be denoted as follows:



(a) Changing parameter σ. (b) Changing parameter c.

Fig. 3.11 Effects of changing simple Gaussian MF parameters.

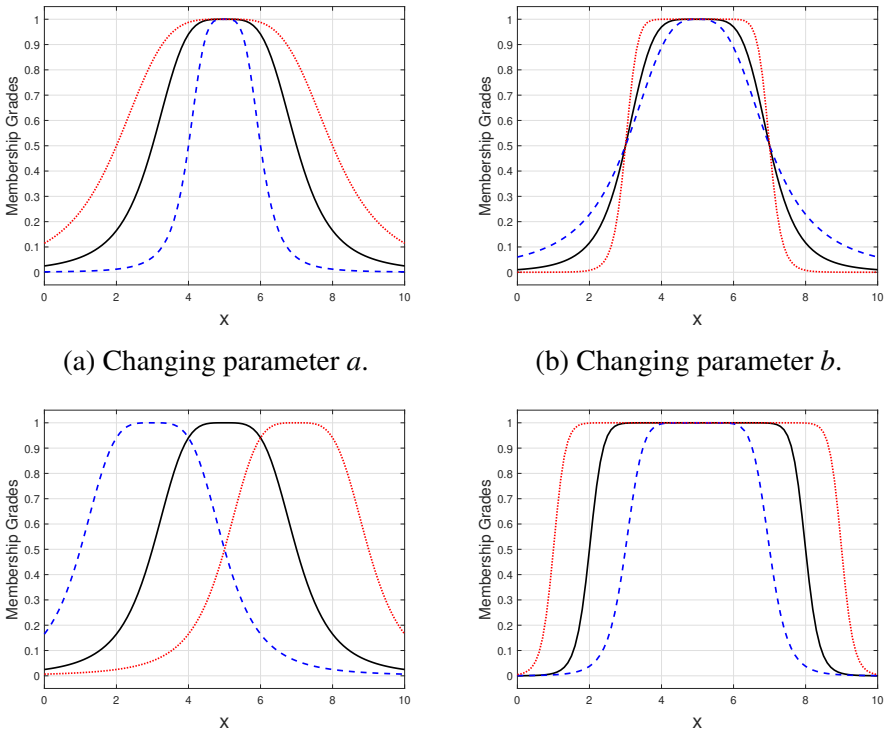
$$\text{The Generalized Bell MF,} \quad f(x;a,b,c) = \frac{1}{1+\left|\frac{x-c}{a}\right|^{2b}}. \tag{3.30}$$

The generalized bell function can be considered as a direct generalization of the Cauchy distribution in the probability distribution, sometimes so-called as the Cauchy MF. Figure (3.12) shows the effects of changing each parameter in a generalized bell MF of $(x;2,3,5)$. The first part (Fig. 3.12a) shows the effectiveness of changing the value of parameter $a = 2$ within the interval of $[1,3]$ on the width of the MF curve from the centroid. Secondly, figure (3.12b) illustrates the varying of the side's slope caused by increasing and decreasing the values of the second parameter b within the interval of $[1.5,6]$. The third figure (3.12c), shows the moving of the centroid of the MF curve resulting from changing the parameter $c = 5$ within the interval of $[3,7]$. The figure (3.12d) demonstrates the effect of changing two parameters a and b at the same time.

3.4.2.3 The Sigmoid Curve Functions

The sigmoid curve functions contain three types of membership functions; the sigmoidal MF, difference sigmoidal MF and product sigmoidal MF. The difference between the first one and the other two is that the sigmoidal MF is either open to the left or right; whilst the difference and product sigmoidal MF are closed and asymmetric, which results from combining two sigmoidal MF. Figure (3.13) shows the plot of these three types.

- **Sigmoidal Membership Function**
  This membership function is determined by two parameters $\{a,c\}$. The parameter a represents the slope of the curve, while the parameter c represents the x coordinate value of the crossover point. The shape of this MF is to be open to the left and the right



(a) Changing parameter a.

(b) Changing parameter b.

(c) Changing parameter c.

(d) Changing parameter $a\&b$.

Fig. 3.12 Effects of changing Generalized Bell MF parameters.

(Fig. 3.13a). This function can be represented by:

$$\text{Sigmoidal MF,} \quad f(x;a,c) = \frac{1}{1+e^{-a(x-c)}}. \tag{3.31}$$

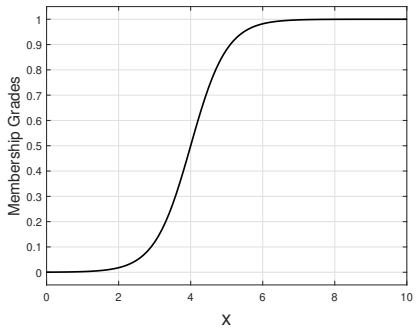
- **Difference Sigmoidal function**

  This membership function represents two different asymmetric sigmoidal curves. That means, it will depend on two pairs of parameters $\{a_1,c_1,a_2,c_2\}$, each pair a_1, c_1 or a_2 ,c_2 will represent one sigmoidal function for one of the two sides (fig 3.13b). This MF can be denoted by:

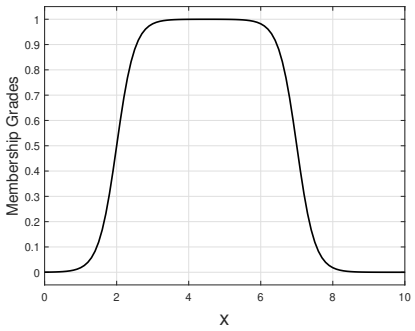
$$\text{Difference Sigmoidal MF,} \quad f_i(x;a_i,c_i) = \frac{1}{1+e^{-a_i(x-c_i)}}. \tag{3.32}$$

  Where $i = 1,2$, draws the left or right of the MF. The difference of the two functions can be denoted as:

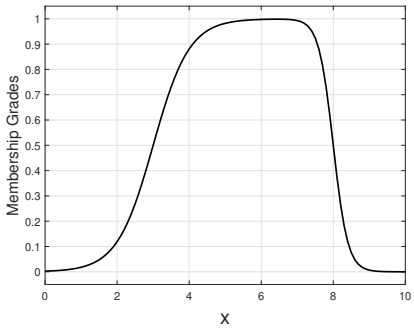
$$f(x;a_1,c_1,a_2,c_2) = f_1(x;a_1,c_1) - f_2(x;a_2,c_2). \tag{3.33}$$



(a) Sigmoidal MF.



(b) Difference Sigmoidal MF.



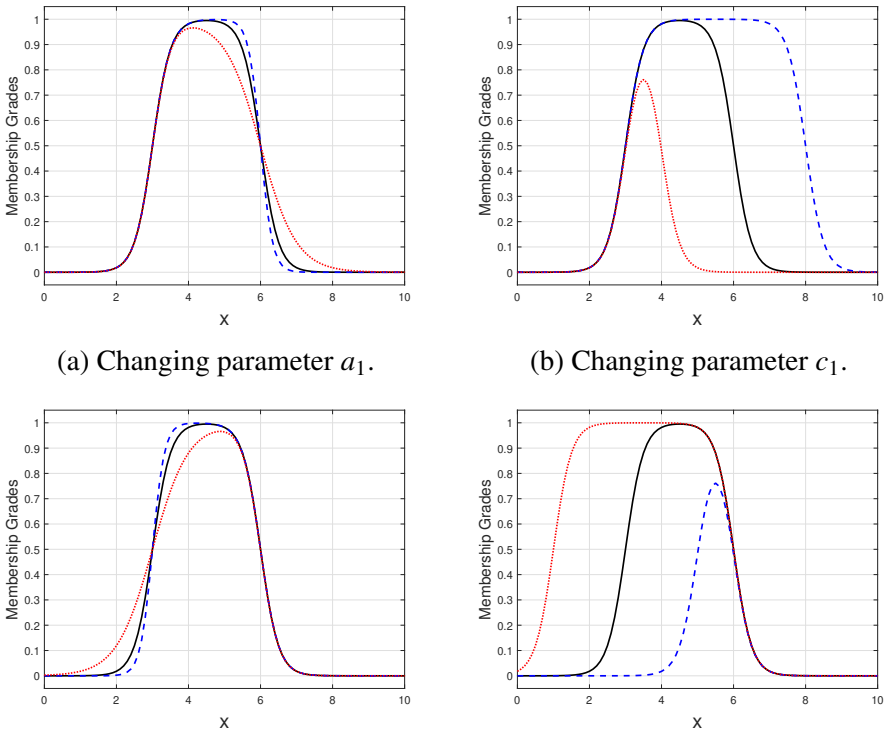
(c) Product Sigmoidal MF.

Fig. 3.13 Sigmoid curve functions.

Figure (3.14) illustrates the effects of changing the difference sigmoidal MF parameters of $a_1 = 4$, $c_1 = 6$, $a_2 = 4$ and $c_2 = 3$. Figures (3.14a, 3.14c) shows the effect of changing the parameters a_1 and a_2 within the interval of $[2,6]$, while, figures (3.14b, 3.14d) demonstrates the effect of changing the parameters c_1 within the interval $[4,8]$ and c_2 within the interval of $[2,5]$.

- **Product Sigmoidal function**
  Similar to the difference sigmoidal function, the product sigmoidal MF represents the product of two asymmetric sigmoidal curves (fig. 3.13c). It has the same parameters that were listed in the order of $\{a_1, c_1, a_2, c_2\}$. This means, the function will have the same mathematical formulation as the previous one. However, the relation between the two functions will be as followed:

$$f(x; a_1, c_1, a_2, c_2) = f_1(x; a_1, c_1) \times f_2(x; a_2, c_2). \tag{3.34}$$



(a) Changing parameter a_1.

(b) Changing parameter c_1.

(c) Changing parameter a_2.

(d) Changing parameter c_2.

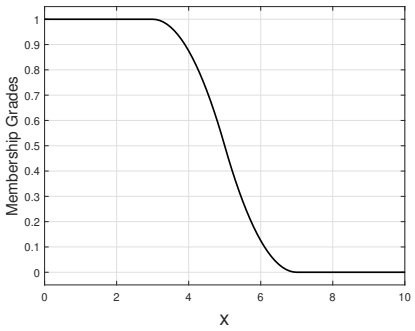
Fig. 3.14 Effects of changing Difference Sigmoidal MF parameters.

3.4.2.4 Quadratic and Cubic Polynomial Curves Functions

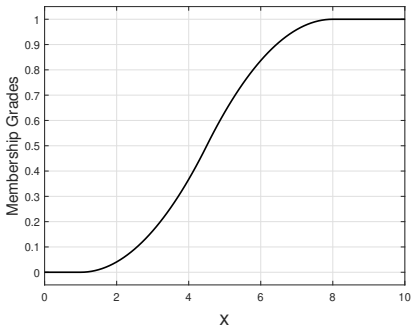
There are three membership functions included under the Quadratic and Cubic Polynomial group of functions. The asymmetrical polynomial curve function, the mirror-image function, and the zero extremes function (refer to fig. 3.15).

- **Asymmetrical polynomial membership function**
  The asymmetrical polynomial curve function can be specified by two parameters $\{a,b\}$. The shape of this MF curve is open to the left. According to the shape of this function, it also called Z-shape MF (fig. 3.15a). The mathematical formulation for this function can be denoted as follows:

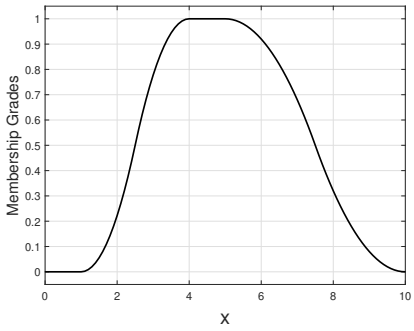
$$f(x;a,b) = \begin{cases} 1, & x \leq a. \\ 1-2\left(\frac{x-a}{b-a}\right)^2, & a \leq x \leq \frac{a+b}{2}. \\ 2\left(\frac{x-a}{b-a}\right)^2, & \frac{a+b}{2} \leq x \leq b. \\ 0, & x \geq b. \end{cases} \tag{3.35}$$



(a) Asymmetrical Polynomial MF.



(b) Mirror-Image MF.



(c) Zero Extremes MF.

Fig. 3.15 Quadratic and Cubic Polynomial curves functions.

- **Mirror-Image membership function**
  The mirror-image function depends on two parameters $\{a,b\}$. In contrast to the asymmetrical polynomial curve function, the curve of this function is open to the right, which gives an S-shape and accordingly it so-called S-membership function (fig. 3.15b). The mathematical formulation for this function can be denoted as follows:

$$f(x;a,b) = \begin{cases} 0, & x \leq a. \\ 2\left(\frac{x-a}{b-a}\right)^2, & a \leq x \leq \frac{a+b}{2}. \\ 1-2\left(\frac{x-a}{b-a}\right)^2, & \frac{a+b}{2} \leq x \leq b. \\ 1, & x \geq b. \end{cases} \tag{3.36}$$

- **Zero Extremes membership function**
  The zero extremes MF is specified by four parameters $\{a,b,c,d\}$ as shown in (fig.

3.15c). The mathematical formulation for this function can be denoted as follows:

$$f(x;a,b,c,d) = \begin{cases} 0, & x \leq a. \\ 2\left(\frac{x-a}{b-a}\right)^2, & a \leq x \leq \frac{a+b}{2}. \\ 1-2\left(\frac{x-a}{b-a}\right)^2, & \frac{a+b}{2} \leq x \leq b. \\ 1, & b \leq x \leq c \\ 1-2\left(\frac{x-c}{d-c}\right)^2, & c \leq x \leq \frac{c+d}{2}. \\ 2\left(\frac{x-c}{d-c}\right)^2, & \frac{c+d}{2} \leq x \leq d. \\ 0, & x \geq d. \end{cases} \tag{3.37}$$

3.5 Linguistic Variables

According to Zadeh [161], dealing with system analysis problems using the traditional methods has become inadequate, particularly in dealing with humanistic systems. The reason behind this is because the humanistic systems are highly affected by human judgement, thinking, and emotions. Zadeh [156] pointed out the need for finding an alternative method to achieve the modelling of human thinking. However, using natural language in describing ideas and sharing knowledge will resulting in vagueness. We often use words and sentences to express and give a meaningful explanation for particular processes. Thus, any universe can be represented linguistically, For example, if we take X = **"age", "speed"** or **"tallness"** as a universe under consideration. These words or sentences are predominantly called fuzzy variables or **Linguistic Variables** [157] [158] [159]. The words age, speed, and tallness are linguistic variables if they contain **Linguistic Labels** (or in other words, linguistic terms or linguistic values) instead of numerical values. For instance, the linguistic variable **age** can be separated into the linguistic labels (**"young"**, **"middle aged"** or **"old"**).

Example 3.4

Assume that X = "tallness" represents the universe (i.e., linguistic variable) under consideration. We can identify some points as a milestone for the tallness (i.e., $x_1 = 160cm$, $x_2 = 170cm$, $x_3 = 180cm$, $x_4 = 190cm$, $x_5 = 200cm$) in height. Then we can define two fuzzy sets A_1 and A_2 within X, to have linguistic labels "tall" and "very tall" respectively. Let A_1 be associated with the grade of membership $\mu_{A_1}(x)$ of (0, 0.6, 1.0, 0.5, 0) and A_2 with $\mu_{A_2}(x)$ of (0, 0, 0.1, 0.3, 0.9) respectively. The MF of the fuzzy sets for the two linguistic labels "tall"

and "very tall" can be written as:

$$A_1(x) = \mu_{tall}(x) = \{(x_1,0),\ (x_2,0.6),\ (x_3,1.0),\ (x_4,0.55),\ (x_5,0)\}$$
$$A_2(x) = \mu_{verytall}(x) = \{(x_1,0),\ (x_2,0),\ (x_3,0.1),\ (x_4,0.3),\ (x_5,0.9)\}$$

Figure (3.16) shows the graphical representation of the two fuzzy sets "tall" and "very tall" within the universe "tallness". For instance, it can be clearly noted that the element x_4 is a member of the fuzzy set "tall" with grade of membership 0.55 and to the fuzzy set "very tall" with grade of membership 0.3.

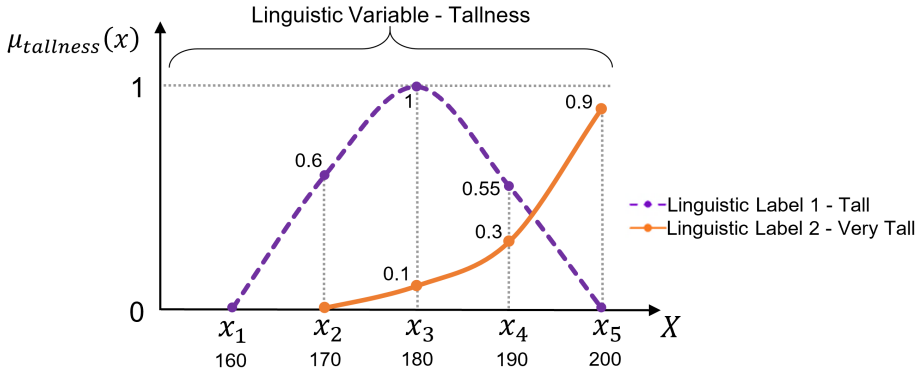


Fig. 3.16 Plotting of linguistic labels (fuzzy sets) of the linguistic variable (universe) "Tallness".

3.5.1 Linguistic Variables and its Related Terminology

We commonly separate the universe into several fuzzy sets associated with membership functions in order to cover the universe X (i.e., the whole set) in a comprehensive manner. As has been mentioned earlier, each universe of discourse (linguistic variable) consists of some fuzzy sets (linguistic label). In this section, we will define three levels of linguistic terminology corresponding with the hierarchies of the fuzzy sets as follows:

- **Linguistic variable:** The linguistic variable represents the terminology of the universe of discourse or in other words, **Whole set** (the collection of all elements in the universe). Such as, X = **"age", "speed", "size"**, . . . , etc.

- **Linguistic label:** It is also called the primary terms or linguistic values. It can be derived from the linguistic variable to represent the terminology of the **fuzzy sets** (i.e., the collection of particular elements and its membership grades in the universe). For

example, consider the linguistic variable **"speed"** which can be derived into some linguistic label such as $A(x)$ = **"slow"**, **"medium"** and **"fast"**.

- **Linguistic hedges:** the linguistic hedges can be derived from the primary terms (i.e., the linguistic labels). It represents the terminology of the **modified fuzzy sets** (the collection of particular elements and its membership grades within the fuzzy set). For example, consider the linguistic hedges (**"very slow", "more or less slow", "quite slow", "extremely slow"**, ...,etc) which can be derived from the linguistic label **"slow"**.

Figure (3.17) shows the linguistic variable **"speed"** and its linguistic labels & hedges.

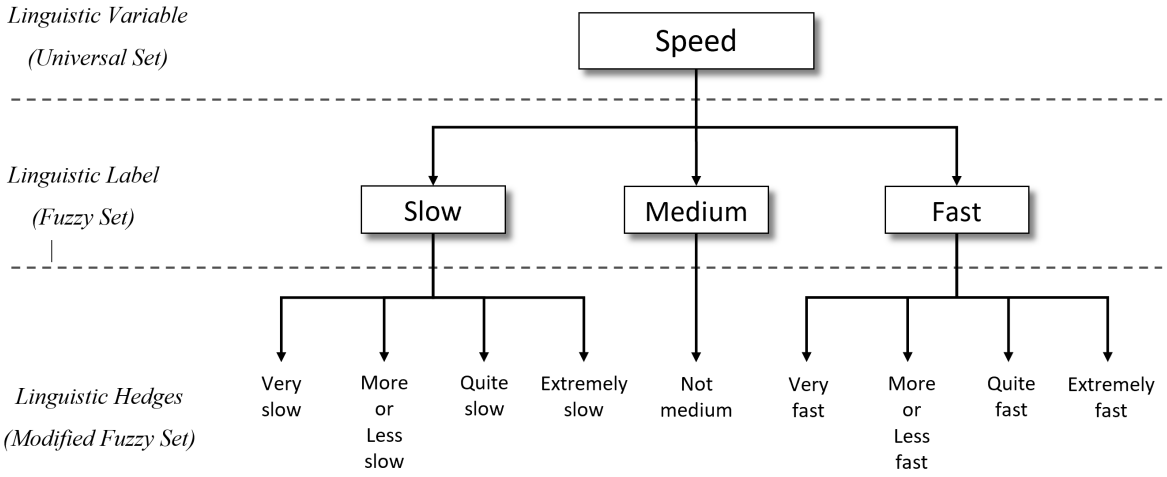


Fig. 3.17 Linguistic variable "Speed" and its linguistic labels and hedges.

Generally, the characterization of a linguistic variable can be denoted by the quinary $(x, T(x), X, G, M)$ where x represent the variable name (e.g., Speed); $T(x)$ is a fuzzy set (term set) of x that represent the linguistic labels (e.g., slow, medium, fast, ..., and so forth); X is the universe; G is the syntactic rule that creates the labels in $T(x)$, and M is the semantic rule of every linguistic label [117, 61, 86].

Example 3.5

Assume that we have the speed as a linguistic variable, then we can define its linguistic labels (terms) $T(speed)$ as follows:

$$\begin{aligned} T(speed) = \{&slow,\ not\ slow,\ very\ slow,\ not\ very\ slow,\ more\ or\ less\ slow,\ \ldots,\\ &medium\ speed \text{ (i.e., not slow \& not fast)},\ not\ medium\ speed,\ \ldots,\\ &fast,\ not\ fast,\ very\ fast,\ more\ or\ less\ fast,\ not\ very\ fast,\ \ldots,\\ &not\ very\ slow\ and\ not\ very\ fast,\ \ldots\} \end{aligned} \tag{3.38}$$

Commonly we use "speed is fast" to assign the linguistic label "fast" to the linguistic variable "speed". On the contrary, if we use the numerical variable to interpret the speed, we say "speed = 120km" to assign a digital value to the numerical variable. Whereas, we used the syntactic rule to generate the linguistic label "fast" for the linguistic variable $T(speed)$. On the other side, the semantic rule gives the membership function of each linguistic label. If we use the typical membership function (e.g., Gaussian) to represent the linguistic variable $T(speed)$. Each linguistic label is to be characterized by a fuzzy set in the universe speed (i.e., $X = [0, 200]$). All linguistic labels (terms) and its hedges can be plotted using the Gaussian membership function with different modified parameters (σ, c), as shown in figure (3.17).

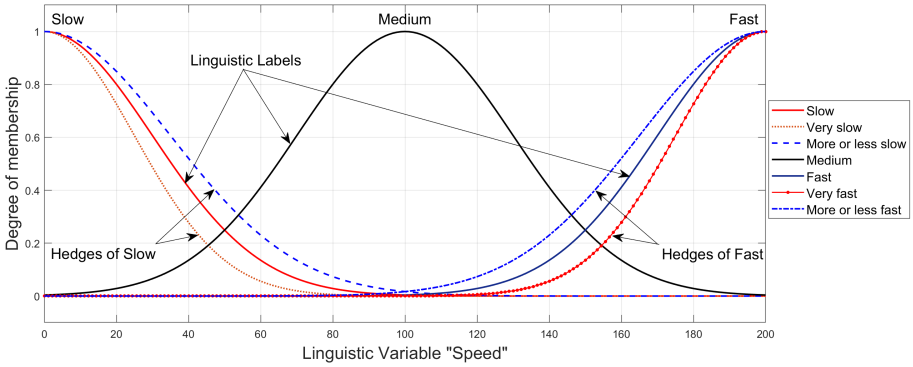


Fig. 3.18 Typical Gaussian MF of linguistic Variable "Speed" and its linguistic labels & hedges.

Referring to the previous example (5), it can be noted that the **linguistic variable** $T(speed)$ composed of some **linguistic labels** (primary terms) such as *("slow", "medium speed", and "fast")* modified by the **linguistic hedges** *(e.g., "very", "more or less", "quite", "extremely", ..., etc.)* in addition to the **negation** (Not), then linked by **connectives** such as *(and, or, either, neither, ..., etc.)*.

3.5.2 Concentration and Dilation of Linguistic Labels

As mentioned earlier, the linguistic hedges can be derived by modifying the linguistic label. Referring to the previous example (5), assume that the linguistic label $(A = slow)$ is a fuzzy set with a membership function $\mu_{slow}(x)$. Then, we can derive linguistic hedges by modifying A into A^k as follows:

$$A^K = (slow)^k = \int_X [\mu_{slow}(x)]^2 / x. \tag{3.39}$$

Accordingly, we can define the **concentration** as the operation that can modify the linguistic label by squaring its membership values , it can be denoted by:

$$\mathrm{CON}(slow) = (slow)^2 = [\mu_{slow}(x)]^2 \implies very\ slow, \tag{3.40}$$

and the **dilation** as:

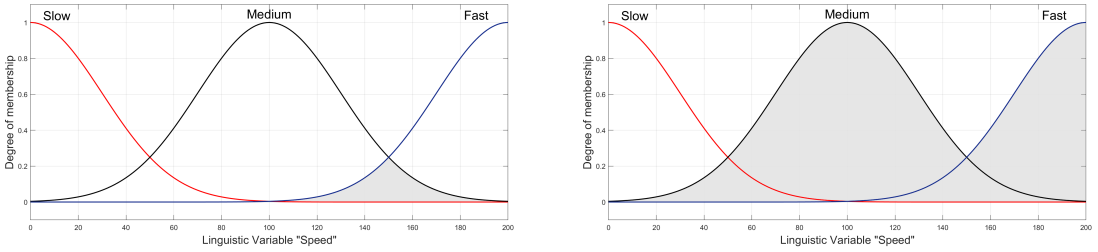
$$\mathrm{DIL}(slow) = (slow)^{0.5} = [\mu_{slow}(x)]^{0.5} \implies more\ or\ less\ slow, \tag{3.41}$$

Where, $\mathrm{CON}(slow)$ and $\mathrm{DIL}(slow)$ represent the linguistic hedges "very" and "more or less" respectively (refer to figure 3.18).

Also, the negation (Not) and the connectives AND and OR of two linguistic labels $(medium)$ and $(fast)$ can be denoted as:

$$\begin{aligned}
&\mathrm{NOT}(medium) = -(medium) = \int_X [1 - \mu_{medium}(x)]/x, \\
&medium\ \mathrm{AND}\ fast = medium \cap fast = \int_X [\mu_{medium}(x) \wedge \mu_{fast}(x)]/x, \\
&medium\ \mathrm{OR}\ fast = medium \cup fast = \int_X [\mu_{medium}(x) \vee \mu_{fast}(x)]/x.
\end{aligned} \tag{3.42}$$

The linguistic terms "medium AND fast" and "medium OR fast" are shown in figures (3.19a & 3.19b). The highlighted area representing the intersection and union of the two membership functions respectively.



(a) Medium AND fast.

(b) Medium OR fast.

Fig. 3.19 Connectives AND & OR of two linguistic labels.

By using all the previous terms of $\mathrm{CON}(x)$ and $\mathrm{DIL}(x)$, the hedges (*very* and *more or less*), the negation (NOT), and the connectives (AND, OR), all can be driven in the preceding equation (3.42). Accordingly, we can create a composite linguistic label, for example, "slow

but not very slow" and "not very fast and not very slow" ..., etc; these composites can be represented mathematically.

Example 3.6

Referring to example 5, presume that we have the linguistic labels "slow" and "fast", if we use the Gaussian membership function to define these two terms each linguistic label is to be characterized by a Gaussian membership function with different parameters (σ, c) in the universe "Speed", i.e., $X = [0, 200]$ i.e. :

$$\begin{aligned} \mu_{slow} = &\quad gauss(x; 30, 0) = e^{-\frac{1}{2}(\frac{x-0}{30})^2}, \\ \mu_{fast} = &\quad gauss(x; 30, 200) = e^{-\frac{1}{2}(\frac{x-200}{30})^2}, \end{aligned} \tag{3.43}$$

where x represents the the speed of a given car. Consequently, we can create the membership functions of the following composite linguistic labels:

- Very (fast) = CON (fast) = $[\mu_{fast}(x)]^2$

$$= \int_X \left[e^{-\frac{1}{2}(\frac{x-200}{30})^2} \right]^2 \Big/ x.$$

- More or less (fast) = DIL (fast) = $[\mu_{fast}(x)]^{0.5}$

$$= \int_X \sqrt{e^{-\frac{1}{2}(\frac{x-200}{30})^2}} \Big/ x.$$

- Not slow and not fast = $-slow \cap -fast$

$$= \int_X \left[1 - e^{-\frac{1}{2}(\frac{x-0}{30})^2} \right] \wedge \left[1 - e^{-\frac{1}{2}(\frac{x-200}{30})^2} \right] \Big/ x.$$

- Slow but not very slow = $slow \cap -slow^2$

$$= \int_X \left[e^{-\frac{1}{2}(\frac{x-0}{30})^2} \right] \wedge \left[1 - \left(e^{-\frac{1}{2}(\frac{x-0}{30})^2} \right)^2 \right] \Big/ x.$$

- Extremely fast = very very very fast

$$\text{CON}(\text{CON}(\text{CON}(fast) = ((fast^2)^2)^2 = \int_X \left[e^{-\frac{1}{2}(\frac{x-200}{30})^2} \right]^8 \Big/ x.$$

3.6 Fuzzy Relations

Following the earlier description of the fuzzy sets operations, in this section, we are considering the mathematical representation of the linguistic statements using the concept of fuzzy relations [97]. Assume that we have two universes under consideration, labelled X and Y. Then we can define a **binary fuzzy relation** R from the fuzzy set A in X to the fuzzy set B in Y using the Cartesian product of $A \times B$ in the universal space $X \times Y$ as follows:

$$R = A \times B = \{(\ (x,y),\ \mu_R(x,y))|(x,y) \in X \times Y\}, \tag{3.44}$$

where $\mu_R(x,y) = \mu_{A\times B}(x,y)$ is a **two-dimensional** fuzzy set, which can also be called a fuzzy relation matrix.

Recalling the fuzzy operations (Cartesian product and Algebraic product) of two fuzzy sets that has been introduced in section (3.3.2), the binary fuzzy relation can be provided by using the following:

- **Cartesian product:**
  According to equation (3.19), we have the ability to compose a binary fuzzy relation for the preceding equation (3.44) using the Cartesian product, which can be denoted as follows:
  $$R = A \times B = \sum \mu_R(x,y)|(x,y) = \sum min(\mu_A(x), \mu_B(y)), \tag{3.45}$$

- **Algebraic product:**
  According to equation (3.12), we can compose a binary fuzzy relation for the preceding equation (3.44) using the Algebraic product, which can be denoted as follows:
  $$R = A \times B = \sum \mu_R(x,y)|(x,y) = \sum min(\mu_A(x).\mu_B(y)), \tag{3.46}$$

where the symbol ($\sum$) does not refer to the numerical summation, but instead refers to all possible combinations of all elements in both fuzzy sets.

Example 3.7

Suppose we have the following two fuzzy sets defined in the universes X and Y:

$$A(x) = \{(x_1,1.0),\ (x_2,0.7),\ (x_3,0.5),\ (x_4,0.4),\ (x_5,0.2)\},$$

$$B(y) = \{(y_1,0.3),\ (y_2,1.0),\ (y_3,0.4)\}.$$

Now, the binary fuzzy relation R can be obtained by multiplying the two fuzzy sets $A(x)$ and $B(y)$ using the Cartesian product. The result will be as follows:

$$R = A \times B = \begin{bmatrix} \{1,.3\} & \{1,1\} & \{1,.4\} \\ \{.7,.3\} & \{.7,1\} & \{.7,.4\} \\ \{.5,.3\} & \{.5,1\} & \{.5,.4\} \\ \{.4,.3\} & \{.4,1\} & \{.4,.4\} \\ \{.2,.3\} & \{.2,1\} & \{.2,.4\} \end{bmatrix} = \begin{bmatrix} 0.3 & 1 & 0.4 \\ 0.3 & 0.7 & 0.4 \\ 0.3 & 0.5 & 0.4 \\ 0.3 & 0.4 & 0.4 \\ 0.2 & 0.2 & 0.2 \end{bmatrix}.$$

The result of using the Algebraic product to obtain the binary fuzzy relation R will be:

$$R = A \times B = \begin{bmatrix} \{1,.3\} & \{1,1\} & \{1,.4\} \\ \{.7,.3\} & \{.7,1\} & \{.7,.4\} \\ \{.5,.3\} & \{.5,1\} & \{.5,.4\} \\ \{.4,.3\} & \{.4,1\} & \{.4,.4\} \\ \{.2,.3\} & \{.2,1\} & \{.2,.4\} \end{bmatrix} = \begin{bmatrix} 0.3 & 1 & 0.4 \\ 0.21 & 0.7 & 0.28 \\ 0.15 & 0.5 & 0.2 \\ 0.12 & 0.4 & 0.16 \\ 0.06 & 0.2 & 0.08 \end{bmatrix}.$$

So far, we know that the binary fuzzy relations are resulting from combining two fuzzy sets by using either Cartesian product or Algebraic product fuzzy operations. However, we can find the composition of two relations by using the same previous fuzzy operations (Fig. 3.20).

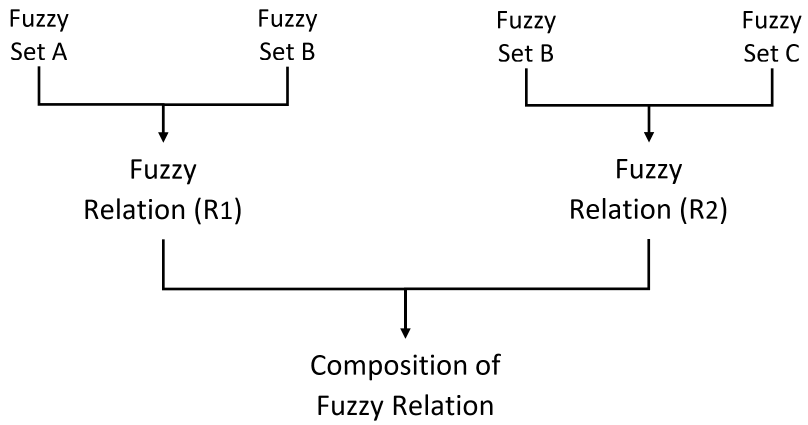


Fig. 3.20 Composition of two fuzzy relations.

Accordingly, we will establish the best known composition operations as follows:

3.6.1 Max-Min Composition

Assume that there are two fuzzy relations R_1 and R_2 defined on the space of the universes $X \times Y$ and $Y \times Z$, respectively. Consequently, the max-min composition of these two fuzzy relations can be defined as:

$$\begin{aligned} C_{max-min} =& R_1 \circ R_2 \\ =& \{(x,z),\ max\ [min(\mu_{R_1}(x,y), \mu_{R_2}(y,z))] \mid x \in X, y \in Y, z \in Z\}, \end{aligned} \tag{3.47}$$

which is equivalent to,

$$\mu_{R_1 \circ R_2}(x,z) = max[min(\mu_{R_1}(x,y), \mu_{R_2}(y,z))], \tag{3.48}$$

and by using matrix form,

$$[c_{ik}] = [r_{1_{ij}}] \circ [r_{2_{jk}}], \tag{3.49}$$

where,

$$c_{ik} = max[min(r_{1_{ij}}, r_{2_{jk}})]\ . \tag{3.50}$$

3.6.2 Max-Product Composition

Considering the same notations we used in max-min composition, then:

$$\text{max-product} = \mu_{R_1 \circ R_2}(x,z) = max[min(\mu_{R_1}(x,y).\mu_{R_2}(y,z))], \tag{3.51}$$

and by using matrix form,

$$[c_{ik}] = [r_{1_{ij}}] \circ [r_{2_{jk}}], \tag{3.52}$$

where,

$$c_{ik} = max[min(r_{1_{ij}}.r_{2_{jk}})]\ . \tag{3.53}$$

Example 3.8

Presume that we have the following relation matrices which represents two fuzzy relations R_1 and R_2. Such as,

$$R_1 = [r_{1_{ij}}] = \begin{bmatrix} 0.2 & 0.6 & 0.2 \\ 0.1 & 0.8 & 0.6 \end{bmatrix},$$

$$R_2 = [r_{2_{jk}}] = \begin{bmatrix} 0.3 & 0.5 \\ 0.2 & 0.7 \\ 0.6 & 0.9 \end{bmatrix}.$$

Now we want to find the composition $R_1 \circ R_2$ between of the two fuzzy relation. Firstly, by using max-min composition as follows:

$$\begin{aligned} c_{11} &= \max[\min(r_{1_{11}}, r_{2_{11}}), \min(r_{1_{12}}, r_{2_{21}}), \min(r_{1_{13}}, r_{2_{31}}),] \\ &= \max[\min(0.2, 0.3), \min(0.6, 0.2), \min(0.2, 0.6),] \\ &= 0.2 \end{aligned}$$

$$\begin{aligned} c_{12} &= \max[\min(r_{1_{11}}, r_{2_{12}}), \min(r_{1_{12}}, r_{2_{22}}), \min(r_{1_{13}}, r_{2_{32}}),] \\ &= \max[\min(0.2, 0.5), \min(0.6, 0.7), \min(0.2, 0.9),] \\ &= 0.6 \end{aligned}$$

$$\begin{aligned} c_{21} &= \max[\min(r_{1_{21}}, r_{2_{11}}), \min(r_{1_{22}}, r_{2_{21}}), \min(r_{1_{23}}, r_{2_{31}}),] \\ &= \max[\min(0.1, 0.3), \min(0.8, 0.2), \min(0.6, 0.6),] \\ &= 0.6 \end{aligned}$$

$$\begin{aligned} c_{22} &= \max[\min(r_{1_{21}}, r_{2_{12}}), \min(r_{1_{22}}, r_{2_{22}}), \min(r_{1_{23}}, r_{2_{32}}),] \\ &= \max[\min(0.1, 0.5), \min(0.8, 0.7), \min(0.6, 0.9),] \\ &= 0.7 \end{aligned}$$

$$C_{max-min} = \begin{bmatrix} 0.2 & 0.6 \\ 0.6 & 0.7 \end{bmatrix}.$$

Secondly, by using the max-product composition and following the same procedures we will have the following relation matrix:

$$C_{max-product} = \begin{bmatrix} 0.12 & 0.18 \\ 0.36 & 0.56 \end{bmatrix}.$$

3.7 Fuzzy If-Then Rules

Fuzzy if-then rules can be defined as the conditional statement of fuzzy logic; also called the **fuzzy rule**. It takes the following form:

$$\text{If} < fuzzy\ proposition > \text{Then} < fuzzy\ proposition >$$

such as,

$$\text{if } x \text{ is } A \text{ then } y \text{ is } B, \text{ or } A \rightarrow B \tag{3.54}$$

where A and B represents the linguistic labels (i.e. fuzzy sets) on the universes X and Y, respectively. Predominantly, the term (x is A) refers to **antecedent** or **premise**, whereas (y is B) refers to **consequence** or **conclusion**. There are some common examples of the fuzzy rules such as:

- **If** *Speed is fast* **Then** *Pressure is low*.
- **If** *Service is good* **Then** *Tip is average*.
- **If** *Road is wet* **Then** *Safety distance must be long*.

Generally, there are three forms represents the linguistic variables [117]:

i. Assignment form: A common example of this type is, x is *not fast* AND *not very slow*.

ii. Conditional form: e.g., IF x is *fast* THEN y is *slow*.

iii. Unconditional form: e.g., the speed is *fast*.

Usually, the fuzzy rule (if then rule) is defined as a fuzzy relation R that is expected to be a binary relation for x and y within the product space $X \times Y$. Thus, the fuzzy rule $A \rightarrow B$ can be interpreted in two ways:

1. $A \rightarrow B$ interpreted as A **coupled with** B then:

$$R = A \rightarrow B = A \times B = \int_{X \times Y} \mu_A(x) \hat{*} \mu_B(y) | (x,y), \tag{3.55}$$

where $\hat{*}$ represents the T-norm operator, and R represent the fuzzy relation (i.e. $A \rightarrow B$).

2. $A \rightarrow B$ interpreted as A **entails** B then:

   - Material implication:

$$R = A \rightarrow B = \neg A \cup B. \tag{3.56}$$

   - Propositional calculus:

$$R = A \rightarrow B = \neg A \cup (A \cap B). \tag{3.57}$$

   - Extended propositional calculus:

$$R = A \rightarrow B = (\neg A \cap \neg B) \cup B. \tag{3.58}$$

   - Generalization of Modus Ponens:

$$\mu_R(x,y) = \sup\{c \mid \mu_A(x) \hat{*} c \leq \mu_B(y) \text{ and } 0 \leq c \leq 1\} \tag{3.59}$$

   Again, $\hat{*}$ represents the T-norm operator, and R represents the fuzzy relation (i.e. $A \rightarrow B$). Also, if A and B are two logic propositions then all four equations (3.56 - 3.59) can be reduced to the form ($A \rightarrow B \equiv \neg A \cup B$).

By combining the previous two interpretations with the two operators (T-norm and S-norm), we can derive a number of formulations to calculate $R = A \rightarrow B$. Where R is the fuzzy set that represents the fuzzy relation with a two-dimensional membership function, that is:

$$\mu_R(x,y) = f(\mu_A(x), \mu_B(y))) = f(a,b), \tag{3.60}$$

where $f(a,b)$ represent the **fuzzy implication function** transforming the membership grade of x in A and y in B into (x,y) in $A \rightarrow B$.

3.7.1 Compound Rules

In general, a linguistic expression might include a compound rule structure. Applying the properties and operations of the fuzzy set can reduce the compound rule into a simplified

rule. Therefore, the two compound rules (Conjunctive and Disjunctive) can be expressed as follows:

(i) **Conjunctive antecedents:** A multiple conjunctive antecedent can be expressed as:

$$\text{IF } x \text{ is } A_1 \text{ AND } x \text{ is } A_2 \ldots \text{AND } x \text{ is } A_n \text{ THEN } y \text{ is } B_s \tag{3.61}$$

Equation 3.61 can be simplified as

$$\text{IF } x \text{ is } A_s \text{ THEN } y \text{ is } B_s, \tag{3.62}$$

where $A_s = A_1 \cap A_2 \cap \cdots \cap A_n$ and A_s can be expressed by using membership function form based on intersection operation as

$$\mu_{A_s}(x) = \min\left[\mu_{A_1}(x),\ \mu_{A_2}(x),\ \ldots,\ \mu_{A_n}(x)\right] \tag{3.63}$$

(ii) **Disjunctive antecedents:** In a similar way, the multiple disjunctive antecedent can be expressed as:

$$\text{IF } x \text{ is } A_1 \text{ OR } x \text{ is } A_2 \ldots \text{OR } x \text{ is } A_n \text{ THEN } y \text{ is } B_s \tag{3.64}$$

Equation 3.64 can be simplified as

$$\text{IF } x \text{ is } A_s \text{ THEN } y \text{ is } B_s \tag{3.65}$$

where $A_s = A_1 \cup A_2 \cup \cdots \cup A_n$ and A_s can be expressed by using membership function form based on union operation as

$$\mu_{A_s}(x) = \max\left[\mu_{A_1}(x),\ \mu_{A_2}(x),\ \ldots,\ \mu_{A_n}(x)\right] \tag{3.66}$$

3.7.2 Aggregation of Rules

Ordinarily, the structure of the rule-based systems contain more than one rule. The aggregation of rules can be defined as the process of obtaining the overall consequent. It results from the accumulation of the contribution of all individual rules in the system. If a system of rules is jointly satisfied, then the rules are connected by (AND) connectives. In this case, the fuzzy intersection of the entire rule's consequents are to be used to obtain the aggregated output y_i ,

where $i = 1,2,3,\ldots,r$:

$$y = y_1 \; AND \; y_2 \; AND \; \cdots AND \; y_r \tag{3.67}$$

or equivalently,

$$y = y_1 \cap y_2 \cap \cdots \cap y_r \tag{3.68}$$

or, by using the membership function form

$$\mu_y(y) = \min\left[\mu_{y_1}(y),\ \mu_{y_2}(y),\ \ldots,\ \mu_{y_r}(y)\right] \quad \text{for } y \in Y. \tag{3.69}$$

Furthermore, the aggregated output can be found by using fuzzy union if there is at least one rule that is satisfied. In this case, a disjunctive system of rules can be used and OR connectives connecting the rules for all of the rule's consequents y_i , where $i = 1,2,3,\ldots,r$:

$$y = y_1 \; OR \; y_2 \; OR \; \cdots OR \; y_r \tag{3.70}$$

or equivalently,

$$y = y_1 \cup y_2 \cup \cdots \cup y_r \tag{3.71}$$

or, by using the membership function form

$$\mu_y(y) = \max\left[\mu_{y_1}(y),\ \mu_{y_2}(y),\ \ldots,\ \mu_{y_r}(y)\right] \quad \text{for } y \in Y \tag{3.72}$$

3.8 Fuzzy Reasoning

Fuzzy reasoning, or **approximate reasoning**, can be defined using conditional rule-forms to explain the relationship between logic values. In other words, we can infer the truth of proposition B depending on A (i.e. $A \rightarrow B$). Since the traditional basic rule of inference uses Modus Ponens as a particular case, here we use the so-called **generalized Modus Ponens** (GMP) [46]. For example, assume that A defined as "the apple is red", and B as "the apple is ripe", if A is true, then B it is also true. The general **GMP** of this example can be expressed as follows:

premise 1 (fact):	x is A,
premise 2 (rule):	if x is A then y is B,
consequence (conclusion):	y is B.

However, if we know that "the apple is more or less red" then we can infer that "the apple is more or less ripe" which can be denoted as:

premise 1 (fact):	x is A',
premise 2 (rule):	if x is A then y is B,
consequence (conclusion):	y is B',

where A and A' are fuzzy sets of appropriate universe X and B and B' are fuzzy sets of Y and $A \rightarrow B$ can be expressed as fuzzy relation R on $X \times Y$. Then the fuzzy set B inferred by "x is A" and the fuzzy rule "if x is A then y is B" can be defined as:

$$\begin{aligned} \mu_{B'}(y) &= max_x \; min[\mu_{A'}(x), \; \mu_R(x,y)] \\ &= \vee_x \; [\mu_{A'}(x) \; \wedge \; \mu_R(x,y)], \end{aligned} \tag{3.73}$$

or equivalently,

$$B' = A' \circ R = A' \circ (A \rightarrow B). \tag{3.74}$$

The previous definitions represent the inference procedure of fuzzy reasoning that can be used to derive the consequences, as long as it implies the fuzzy relation $A \rightarrow B$, as a binary fuzzy relation. Further discussion follows in order to cover different cases where single or multiple fuzzy rules are combined with single or multiple antecedents; which can affect in explaining the system behaviour. However, this will be presented by using Mamdani's fuzzy inference functions and max-min composition as they have more intuitive, easier to understand rule bases, and broad applicability.

3.8.1 Single Rule with Single Antecedent

This type can be considered as the simplest case, the equation (3.73) represents the formula of this case. However, simplifying this equation yields

$$\begin{aligned} \mu_{B'}(y) &= \underbrace{[\vee_x(\mu_{A'}(x) \wedge \mu_A(x)]}_{w} \wedge \mu_B(y) \\ &= w \; \wedge \; \mu_B(y). \end{aligned} \tag{3.75}$$

Figure (3.21) shows the inference mechanism of this case and how to conclude the consequent of a fuzzy reasoning problem of single rule and single antecedent. The shaded field of the antecedent part of this figure represents the degree of match w resulted from calculating the maximum of $\mu_{A'}(x) \wedge \mu_A(x)$. Consequently, the membership function of B' is equal to B snipped by w, shown as the consequent part of the same figure.

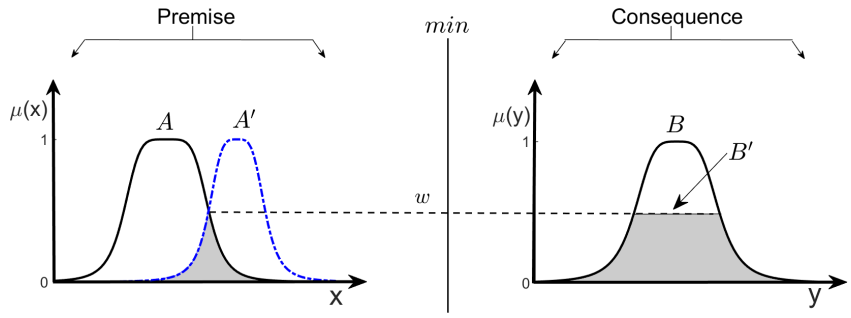


Fig. 3.21 Single rule with single antecedent.

3.8.2 Single Rule with Multiple Antecedents

This type using a fuzzy rule which contains two antecedents, the following expression represents the corresponding GMP for this case:

premise 1 (fact):	x is A' and y is B',
premise 2 (rule):	if x is A and y is B then z is C,
consequence (conclusion):	z is C'.

A more straightforward form of $A \times B \rightarrow C$, can be used for the fuzzy rule shown in premise 2. By using Mamdani's fuzzy inference function, we can transfer this rule into a ternary fuzzy relation R_m as follows:

$$\begin{aligned} R_m(A,B,C) &= (A \times B) \times C \\ &= \int_{X \times Y \times Z} \mu_A(x) \wedge \mu_B(x) \wedge \mu_C(z)/(x,y,z). \end{aligned}$$

Consequently;

$$C' = (A' \times B') \circ (A \times B \rightarrow C).$$

$$\begin{aligned} \mu_{C'}(z) &= \vee_{x,y}[\mu_{A'}(x) \wedge \mu_{B'}(y)] \wedge [\mu_A(x) \wedge \mu_B(y) \wedge \mu_C(z)] \\ &= \vee_{x,y}\{[\mu_{A'}(x) \wedge \mu_{B'}(y) \wedge \mu_A(x) \wedge \mu_B(y)]\} \wedge \mu_C(z) \\ &= \{\underbrace{\vee_x[\mu_{A'}(x) \wedge \mu_A(x)]}_{w_1}\} \wedge \{\underbrace{\vee_y[\mu_{B'}(y) \wedge \mu_B(y)]}_{w_2}\} \wedge \mu_C(z) \\ &= \underbrace{(w_1 \wedge w_2)}_{\text{firing strength}} \wedge \mu_C(z), \end{aligned} \tag{3.76}$$

where w_1 and w_2 represents the **degrees of compatibility** between (A,A') and (B,B') respectively. In other words, w_1 is the maximum of $\mu_A(x) \wedge \mu_{A'}(x)$ (i.e., $A \cap A'$); similarly for w_2 (i.e., $B \cap B'$). The term $(w_1 \wedge w_2)$ expresses the degree of satisfaction of the rule's antecedent, it is also called **degree of fulfilment** or **firing strength** of the fuzzy rule. The shaded area of figure (3.22) shows the graphic interpretation of this case, where $\mu_{C'}(z)$ is the resulting output that is equal to $\mu_C(z)$ snipped by the firing strength w, where $w = w_1 \wedge w_2$.

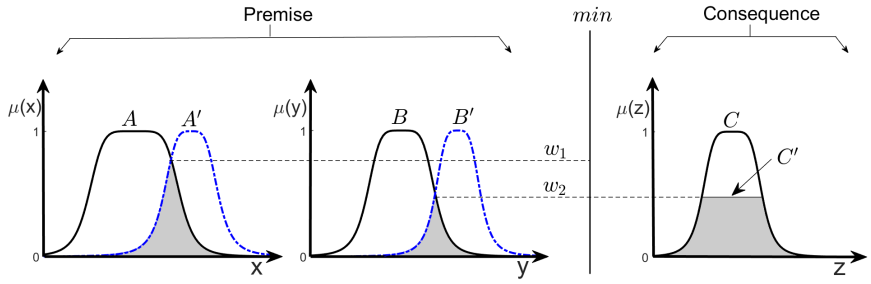


Fig. 3.22 Single rule with multiple antecedent.

3.8.3 Multiple Rules with Multiple Antecedents

This type is applying at least two fuzzy rules and antecedents. Usually, it is using the union of the fuzzy relation corresponding to the fuzzy rules. Thus, the following expression represents the GMP form for this case:

premise 1 (fact):	x is A' and y is B',
premise 2 (rule 1):	if x is A_1 and y is B_1 then z is C_1,
premise 2 (rule 2):	if x is A_2 and y is B_2 then z is C_2,
consequence (conclusion):	z is C',

we can use the GMP form above in addition to the drawing of the fuzzy reasoning of this case shown in figure (3.23) as an inference scheme to determine the fuzzy set C'. Let $R_1 = A_1 \times B_1 \rightarrow C1$ and $R_2 = A_2 \times B_2 \rightarrow C2$. Since the max-min composition operator, $\circ$ is distributive over the $\cup$ operator, then:

$$\begin{aligned} C' &= (A' \times B') \circ (R_1 \cup R_2) \\ &= [(A' \times B') \circ R_1] \cup [(A' \times B') \circ R_2] \\ &= C'_1 \cup C'_2, \end{aligned} \tag{3.77}$$

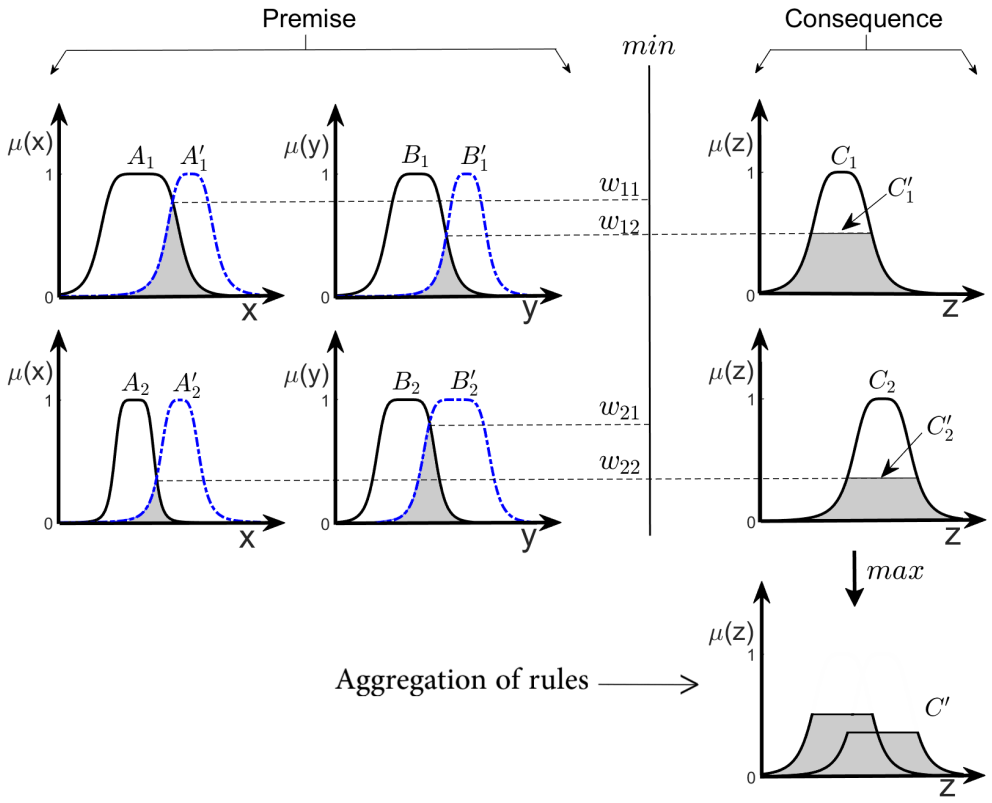


Fig. 3.23 Multiple rule with multiple antecedent

where C_1' and C_2' represents the inferred fuzzy sets for rules 1 and 2, respectively. The shaded area of figure (3.23) shows the graphic interpretation of the operation of fuzzy approximation multiple rules with multiple antecedents.

Chapter 4

Fuzzy Inference System

4.1 Introduction

This chapter will present the implementation methodology of the fuzzy logic and fuzzy theory to formulate and solve real-world problems using fuzzy systems. The **fuzzy inference system** (FIS) can be defined as a computing framework using fuzzy if-then rules combined with the fuzzy reasoning based on the fuzzy set theory in order to describe the mapping from input to output by fuzzy variables and fuzzy relations. It can be considered as one of the successful applications in different areas, such as classification of data, expert system, decision analysis, and time-series forecasting. The multidisciplinary character of the FIS provides several nomenclature, such as **fuzzy rule base system, fuzzy-expert system, fuzzy-model, fuzzy-associative memory, fuzzy-logic controller** [61]. The process of FIS contains all the pieces of **fuzzy logic operations, membership functions, linguistic variables, fuzzy-relations, fuzzy rules (if-then), and fuzzy-reasoning** that we have discussed in the previous section.

In general, a system structure often contains three stages (i.e., input-processing-output). As a fuzzy inference system, the inputs can be either fuzzy or crisp values. However, the produced outputs are often fuzzy. In some cases, if the FIS is used as a controller, then the output can be crisp values. The processing stage includes three components i.e., **fuzzification, inference engine, and defuzzification**; supported by the **fuzzy knowledge base** (FKB). The fuzzy knowledge base is to be constructed of two parts. First, the rule-base, which includes the fuzzy if-then rules; second, the database, which consists of the membership functions definitions (types and numbers). The inference engine represents the mechanism of reasoning for the inference procedure. This can be processed using different inference methods, such as Mamdani fuzzy models and Sugeno fuzzy models. Figure 4.1 interprets the overall structure of the fuzzy inference system.

In this chapter, all the figures, diagrams, and examples have been created by the author, except where indicated with the word "reproduced".

4.2 Fuzzy Inference Control

4.2.1 Fuzzification

In general, implementing the fuzzy logic controller is often accompanied by using crisp data. Therefore, fuzzification plays a vital role in converting crisp (real) input values into fuzzy sets. The fuzzification can be defined as the process of mapping the observed input data into fuzzy sets. In other words, it is providing a membership grade of the crisp (or numeric) values using membership functions stored in the fuzzy knowledge base. It determines the

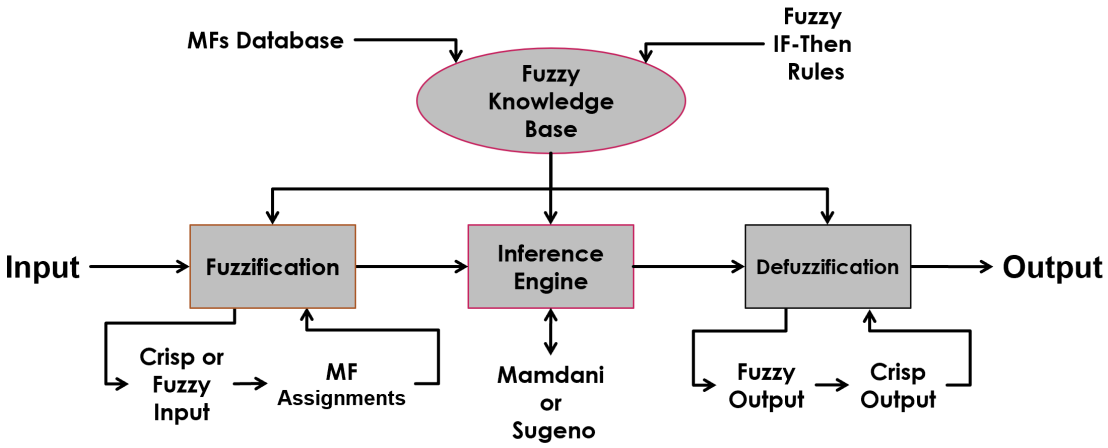


Fig. 4.1 Fuzzy inference system.

intersection value between the numeric input and the membership function. Since the FIS can take either fuzzy or crisp inputs, fuzzification can be handled by two methods:

- Singleton fuzzification: Assume we have the universe of discourse X where $x_i \in X$. A fuzzy Singleton A_{x_i} will represent the membership value that is mapping the real value of x_i into a fuzzy Singleton

$$\mu_{A_{x_i}}(x) = \begin{cases} 1, & \text{if } x = x_i \\ 0, & \text{otherwise.} \end{cases} \tag{4.1}$$

  This type of fuzzification is simplifying the computation, and it can be applied when there is no noise.

- A_{x_i} is fuzzy: This type is mapping the real values of x_i where $x_i \in X$, into a fuzzy set A_{x_i}, and it can be described by a membership function

$$\mu_{A_{x_i}}(x) = \begin{cases} 1, & \text{if } x = x_i \\ [0,1], & x_i < 1. \end{cases} \tag{4.2}$$

Thus, the fuzzification process may involve assigning membership values for the given crisp quantities which interpret the extent to which it belongs to the fuzzy set $\mu_{A_{x_i}}(x)$. The fuzzy set can be characterised by various types of MFs introduced in Section (3.4). Figure 4.2 shows the use of three different types of MF to process the fuzzification of $x_i \in X$. It elucidates that $x = x_i \in X$ has different membership values (i.e., fuzzified grade) when using several types of MF such as, A_1, trapezoidal with membership grade of $\mu_{A_1}(x_i)$; A_2, triangular

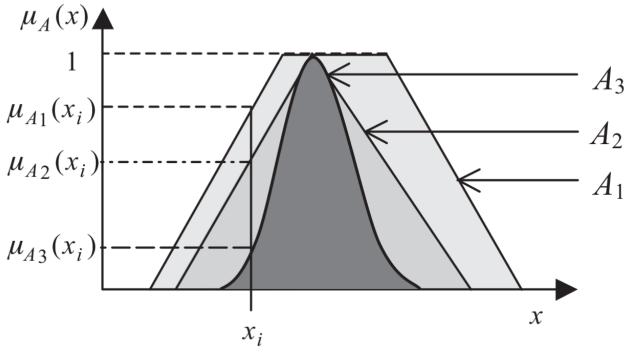


Fig. 4.2 Fuzzification in different types of MFs. Reproduced from [117]

with membership grade of $\mu_{A_2}(x_i)$; and A_3, Gaussian with membership grade of $\mu_{A_3}(x_i)$ respectively.

4.2.2 Inference Engine

The inference engine can be considered as the decision-making unit, which implements the inference operations on the rules. In other words, it performs the mapping of inputs to outputs by using a number of fuzzy if-then rules. The inference engine uses the max-min and max-product compositions to combine the membership values and calculate the *firing strengths (weight)* of each rule, then aggregate qualified consequent membership functions to obtain an overall output. The T-norm and T-conorm operators are used to perform the combining and aggregation of rules. Figure 4.3 (surrounded by a dashed line) shows the inference engine block diagram.

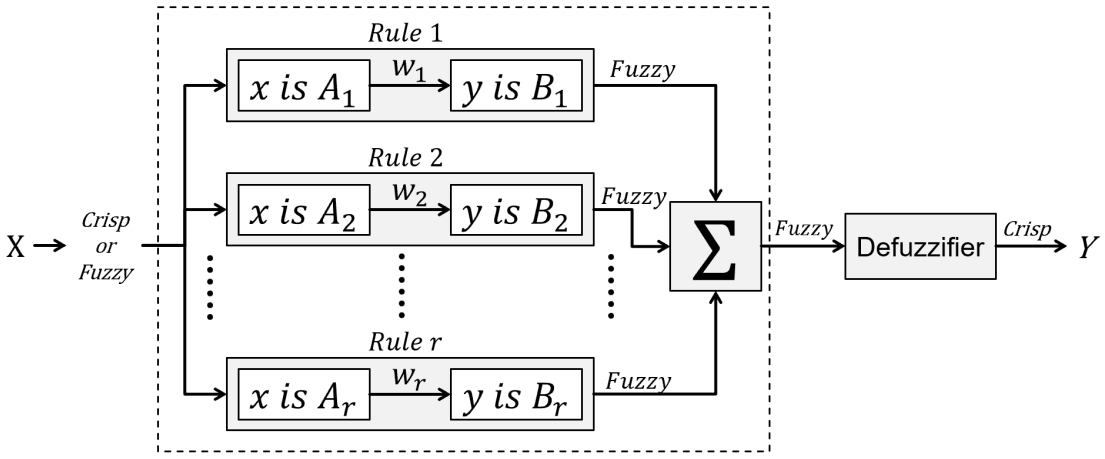


Fig. 4.3 Fuzzy Inference Engine. Reproduced from [61].

4.2.3 Defuzzification

We can define the defuzzification as the process of converting a fuzzy value into a numeric (crisp) value. It can be considered as the inverse of the fuzzification process. In other words, it represents the method of extracting real, numeric values from a fuzzy set. The literature presents many defuzzification methods. The choice of the most appropriate defuzzification method for a particular application depends on the application we use. Runkler [111] presented good research to identify the appropriate method. Broadly, there are some methods used for processing the defuzzification of the fuzzy sets. A brief explanation of the most used methods follows.

4.2.3.1 Centroid of Area Method (COA):

It is also referred to as the centre of gravity or centre of area method. It can be considered as one of the most used methods. This method works by dividing the total area of the membership function distribution into several levelled sub-areas. This is similar to the calculation of expected values of probability distributions [61]. Thus, the defuzzified value is extracted by calculating the centre of gravity of each sub-area and then finding the summation. Mathematically, the general form of this method for the continuous universe can be expressed as follows:

$$z_{COA} = \frac{\int_Z \mu_{C'}(z) z \, dz}{\int_Z \mu_{C'}(z) \, dz}, \tag{4.3}$$

and for m quantization levels in the output of a discrete universe, it is given by

$$z_{COA} = \frac{\sum_{i=1}^{m} \mu_{C'}(z_i).z_i}{\sum_{i=1}^{m} \mu_{C'}(z_i)}, \tag{4.4}$$

where $\mu_{C'}(z)$ represents the aggregated output membership function shown in figure 3.23 (aggregation of rules).

4.2.3.2 Centre of Sums Method (COS):

This method also represents one of the most commonly used methods. It is using the algebraic sum of each output fuzzy set to calculate the defuzzified value. In other words, it is using the summation instead of the union of the output fuzzy sets. Formally, the mathematical expression for the discrete case can be given as follows.

$$z_{COS} = \frac{\sum_{i=1}^{M} z_i \cdot \sum_{i=1}^{m} \mu_{C'_k}(z_i)}{\sum_{i=1}^{M} \cdot \sum_{i=1}^{m} \mu_{C'_k}(z_i)}, \tag{4.5}$$

where M represents the number of fuzzy variables, m is the number of fuzzy sets, and $\mu_{C'_k}(z_i)$ is the MF for the k-th fuzzy set.

4.2.3.3 Mean of Maximum Method (MOM):

It is also referred to as the **middle of max method**. This method identifies the output range of the fuzzy variable corresponding to the elements with the highest value of membership. The mean of all the local maxima is considered to be a single defuzzified output (crisp output). Mathematically, it can be denoted as

$$z_{MOM} = \frac{\sum_{i=1}^{M} \mu_{maxC'}(z_i)}{M}, \tag{4.6}$$

where $\mu_{maxC'}(z_i)$ is the maximum values of membership in the fuzzy output variable C' and M, which represents the number of times when the membership function reached the local-maximum value.

Figure 4.4 shows the difference between the defuzzified values for each method used for obtaining the crisp output of the aggregated fuzzy output variable C', shown in figure 3.23.

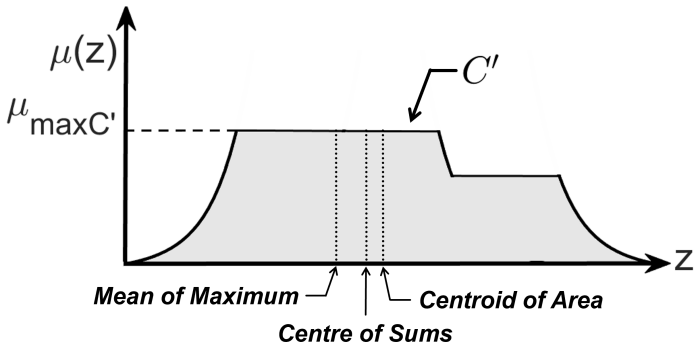


Fig. 4.4 Different defuzzification methods for obtaining crisp output. Adapted from [61]

4.3 Fuzzy Inference Methods (Inference Mechanism)

There are two main methods of modelling a FIS. The first one is known as a Mamdani FIS, which is one of the most commonly used inference methods. Another well-known inference approach is the Takagi-Sugeno [130] FIS; often only referred to as *Sugeno-type*. The main difference between the two approaches lies in the consequent part of the fuzzy rules. The Mamdani fuzzy system's output is a variable with corresponding membership functions. In contrast, T-S fuzzy systems employ linear functions of input variables as rule consequents.

Moreover, when using Mamdani fuzzy systems, the resulting output needs to be defuzzified in such models. The T-S approach is where the output is a function, thus not requiring defuzzification. Also, Mamdani is usually requires prior expert knowledge to develop the FIS (i.e., determining the membership functions and the associated parameters). Whilst T-S can be generated automatically when historical data are available, enabling supervised learning.

4.3.1 Mamdani Fuzzy Inference Method

According to the literature, the Mamdani fuzzy inference method can be considered as one of the first control systems built using fuzzy set theory [82]. Mamdani's system was improved depending on Zadeh's effort [161] on fuzzy algorithms for complex systems and decision processes. As defined, the output of a Mamdani-type inference is expected to be fuzzy sets in the form of membership functions. Each output variable should have its fuzzy set. All output fuzzy sets have to be combined using the aggregation process and be defuzzified in order to produce a real number (crisp value) as the final inference result.

Figure 4.5 shows a Mamdani-FIS, which consists of two inputs x and y (premises) and one output represented by z (consequent). Every input x, y and output z contain two membership functions, i.e., $\{A1, A2\}, \{B1, B2\}$ and $\{C1, C2\}$, respectively. A collection of R rules characterises a typical form of Mamdani-FIS rule, thus

$$k \ : \ \text{if } x \text{ is } A_i^k \text{ and } y \text{ is } B_j^k \text{ then } z \text{ is } C_l^k, \tag{4.7}$$

where $k = 1, 2, \ldots, R$, $i = 1, 2, \ldots, N$, $j = 1, 2, \ldots, M$ and $l = 1, 2, \ldots, L$. R represents the maximum number of rules. Whereas M symbolises the numbers of MFs for inputs, and L expresses the number of MFs for the output. Consider the inference mechanism in figure 4.5, there are two rules to be used,

$$\text{Rule1: IF } x \text{ is } A_1 \text{ AND } y \text{ is } B_1 \text{ , THEN } z \text{ is } C_1 \text{ ,}$$
$$\text{Rule2: IF } x \text{ is } A_2 \text{ AND } y \text{ is } B_2 \text{ , THEN } z \text{ is } C_2 \text{ .}$$

Consider x_1 and y_1 as crisp values specified for inputs x and y, respectively. Adopting max-min, and max-product as the composition rules methods can demonstrate the process of the previous two rules. In other words, we are performing the max and algebraic product using T-norm and T-conorm operators, respectively, to process the fuzzification and inferencing. Figure 4.5(a) shows that the inferred output for each rule is a fuzzy set clipped down by the firing strength with AND operator (usually T-norm) from the antecedent part via minimum or product rule. Figure 4.5(b, d) illustrates the consequent portion of every rule utilising

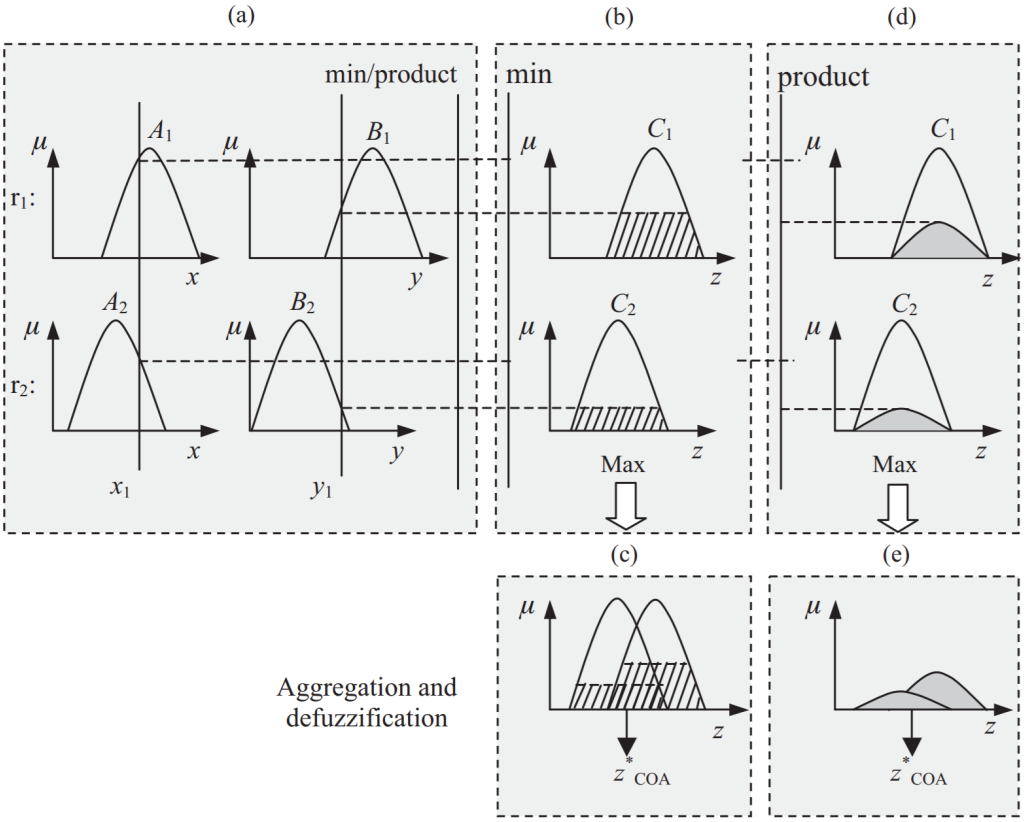


Fig. 4.5 Two-input single-output Mamdani fuzzy model. Reproduced from [117]

max/min, and max/product rules, sequentially. Figure 4.5(c, e) shows the process of rule aggregation for both the max-min as well as the max-product rules of composition. The defuzzification operations have been carried out by applying the centre of area method z^*_{COA} (fig. 4.5(c, e)).

Example 4.1 Assume that we have a Mamdani-FIS with two inputs x and y which express the antecedents, three fuzzy sets (linguistic variables such as low, medium and high), with triangular MF, and a single output z symbolising the consequent. The rule-base includes the following:

$$\text{Rule1: IF } x \text{ is } \textit{low} \text{ AND } y \text{ is } \textit{medium} \text{ , THEN } z \text{ is } \textit{low} \text{ ,}$$
$$\text{Rule2: IF } x \text{ is } \textit{medium} \text{ AND } y \text{ is } \textit{high} \text{ , THEN } z \text{ is } \textit{high} \text{ .}$$

Let us assume that $x_1 = 3.89$ and $y_1 = 5.58$ are two numeric values. Their membership values $\mu_k(x_1)$ and $\mu_k(y_1)$ (k symbolises the MFs low, medium, or high) can be determined

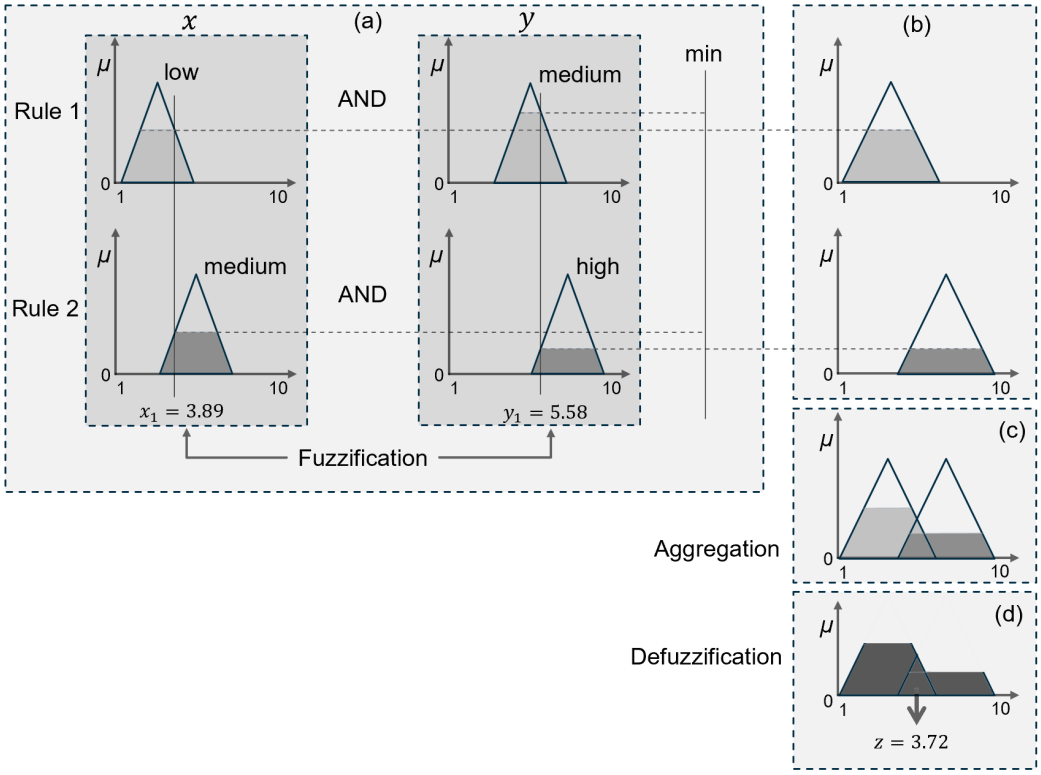


Fig. 4.6 Max/min Mamdani fuzzy inference method. Adapted from [117]

using the triangular MFs. The aggregated outputs of the rules are given by

$$\text{Rule 1:} \quad \mu_{low}(z) = max\ [\ min\ [\mu_{low}(x),\ \mu_{medium}(y)]]$$
$$\text{Rule 2:} \quad \mu_{high}(z) = max\ [\ min\ [\mu_{medium}(x),\ \mu_{high}(y)]]$$

Figure 4.6(a) shows the antecedents part of the Mamdani inference system for our example which contains two rules $r1$ and $r2$. At this stage, the minimum membership value of $[\mu_{low}(x),\ \mu_{medium}(y)]$ and $[\mu_{medium}(x),\ \mu_{high}(y)]$ are to be propagate into the consequent part (fig. 4.6(b)). The consequent MFs of each rule is to be computed by clipping the maximum values based on the produced firing strength, i.e., $max\ [\ min\ [\mu_{low}(x),\ \mu_{medium}(y)]]$ and $max\ [\ min\ [\mu_{medium}(x),\ \mu_{high}(y)]]$ as shown in figure 4.6(b). After that, the consequent MFs are to be aggregated using the max operator (fig. 4.6(c)), then defuzzified using the centroid of area method (fig. 4.6(d)).

4.3.2 Sugeno Fuzzy Inference

This section will introduce the Sugeno fuzzy inference system, which is also known as a Sugeno–Takagi model (TSK). This model was first introduced by Takagi, Sugeno, and

Kang [126] [130] as an effort to formalize a systematic approach to generate fuzzy rules from an input-output data set. A general form of a common rule in a Sugeno model with two inputs and a single output is defined by:

$$k \ : \ \text{if } x \text{ is } A_i \text{ and } y \text{ is } B_j \text{ then } z_k = f(x,y), \tag{4.8}$$

where $k = 1,2,\ldots,R$, $i = 1,2,\ldots,N$, and $j = 1,2,\ldots,M$. N, M and L are the numbers of MFs for the inputs and output, respectively, x and y are the inputs, R is the maximum number of rules, A_i and B_j are fuzzy MFs for the inputs at the antecedent part and $z_k = f(x,y)$ is a crisp function in the consequent part.

The firing strengths w_k (weights) in the antecedents part are to be obtained using the minimum or product composition as the inferencing method of the rules using the AND operator. Thus

$$w_k = \begin{cases} min(\mu_{A_i}, \mu_{B_j}) \\ \text{or,} \qquad\qquad \text{for} \quad k = 1,2,\ldots,R. \\ prod(\mu_{A_i}, \mu_{B_j}). \end{cases} \tag{4.9}$$

The consequence function $z_k = f(x,y)$ can be a polynomial or any other type of function as long as it can appropriately explain the output of the model within the fuzzy region identified by the premise of the rule. In the case of a polynomial function, it usually comes with $\{a_k, b_k, c_k\}$ parameters. Once these parameters are known, the consequence z_k can be computed for each rule. After that, the overall output z (aggregated results) can be calculated via both the firing strengths (weights) and the computed values of z_k. This will compute the weighted average of the crisp output z which works as an alternative method of the defuzzification process in the Mamdani model. Mathematically, the weighted average ($\overline{w}$) form can be as follows:

$$z = \sum \overline{w}_k f_k = \frac{\sum w_k f_k}{\sum w_k} \quad \text{for } k = 1,2,\ldots,R. \tag{4.10}$$

Ordinarily, Sugeno systems are utilised for modelling the inference system when the output MFs are either linear or constant values. According to the degree of order, we can classify the fuzzy inference system into two types as follows:

4.3.2.1 First-order Sugeno Fuzzy Model:

Takagi and Sugeno (1985) proposed the first-order Sugeno fuzzy model [130] [126] to be when the consequent functions of the Sugeno fuzzy model are a first-order (linear)

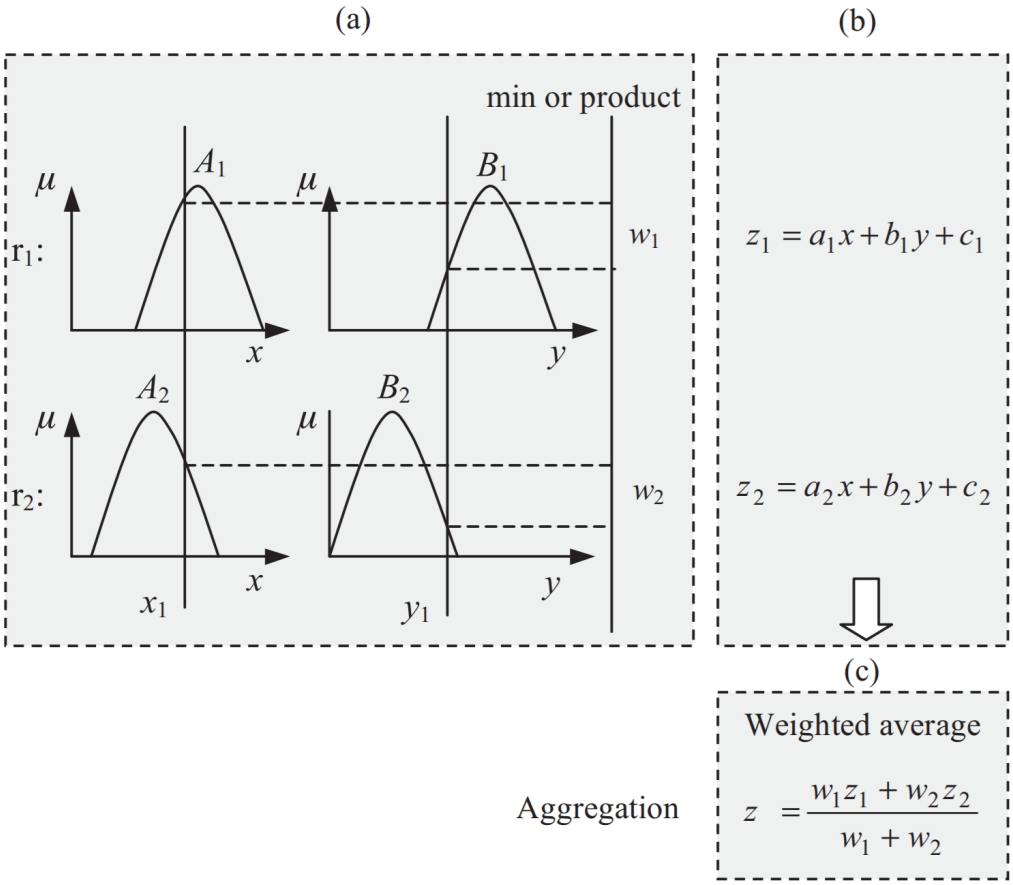


Fig. 4.7 First-order Sugeno fuzzy model. Reproduced from [117]

polynomial. Assume that we have a Sugeno model containing two inputs x and y and one output z. Each input has two membership functions $\{A_1, A_2\}$ and $\{B_1, B_2\}$, respectively, and $\{z_1, z_2\}$ represents the output (consequent) functions. Two rules explain this system, thus:

$$\text{Rule1: IF } x \text{ is } A_1 \text{ AND } y \text{ is } B_1 \text{ , THEN } z_1 = a_1x + b_1y + c_1$$
$$\text{Rule2: IF } x \text{ is } A_2 \text{ AND } y \text{ is } B_2 \text{ , THEN } z_2 = a_2x + b_2y + c_2$$

where $\{a_1, b_1, c_1\}$ and $\{a_2, b_2, c_2\}$ represent the parameters of the polynomial function $z_k = f(x, y)$, where $k = 1, 2$. Figure 4.7 illustrates the previous first-order Sugeno fuzzy model were two crisp values x_1 (for input x) and y_1 (for input y) are used. Figure 4.7(a) illustrates the rules' inferencing procedures after applying the minimum-product composition in order to compute the firing strengths w_1 and w_2. Figure 4.7(b)) demonstrates the consequent part of the Sugeno model, where z_1 and z_2 are obtained. The overall output is computed by equation 4.10 using the weighted average of the crisp outputs z_k and the weights (firing

strengths w_i) of every individual rule determined in Equation 4.9. It can be computed by:

$$z = \frac{w_1 z_1 + w_2 z_2}{w_1 + w_2}$$

Figure 4.7(c)) shows the overall output z (aggregated) result.

4.3.2.2 Zero-order Sugeno Fuzzy Model:

The Sugeno model can be considered zero-order only if the output function in the consequence part is a constant. Referring to the case that we discussed in the previous enumerate (4.3.2.1), the general form of the two rules of zero-order Sugeno fuzzy model can be expressed as follows:

$$\text{Rule1: IF } x \text{ is } A_1 \text{ AND } y \text{ is } B_1 \text{ , THEN } z_1 = c_1$$
$$\text{Rule2: IF } x \text{ is } A_2 \text{ AND } y \text{ is } B_2 \text{ , THEN } z_2 = c_2$$

where c_1 and c_2 are constant values. The fuzzy singleton can be used to determine the consequent of each rule. This can be considered as a special case of the Mamdani model. The obtained values of zero-order Sugeno model is a smooth function of its input variables as long as the contiguous membership functions in the antecedent have enough overlap. By comparing the zero-order Sugeno model with Mamdani model, in Mamdani, the overlap of the membership functions in the consequent part does not have a certain effect on the smoothness of the output. The input/output behaviour and the smoothness of the fuzzy system can be identified from the overlap of the MFs at the antecedent part [60]. In some cases, if a zero-order Sugeno fuzzy model is processed under certain minor constraints, then it can be considered equivalent to the radial basis function network with respect to the functional perspective [120].

Figure 4.8 demonstrates the discussed case of Sugeno model in the previous section (4.3.2.1) with respect to zero-order Sugeno model. The antecedent part (fig 4.8(a)) is using the minimum or product composition in order to compute the firing strengths w_1 and w_2. Whereas, the aggregation part (fig 4.8(c)) is using the weighted average to compute the overall output z_i. Figure 4.8(b) illustrates the consequent portion of the zero-order Sugeno model where z_1 and z_2 are computed. The triangular and step function represents the special cases for the consequent membership function in which provide the defuzzified values c_1^* and c_2^* of the MFs.

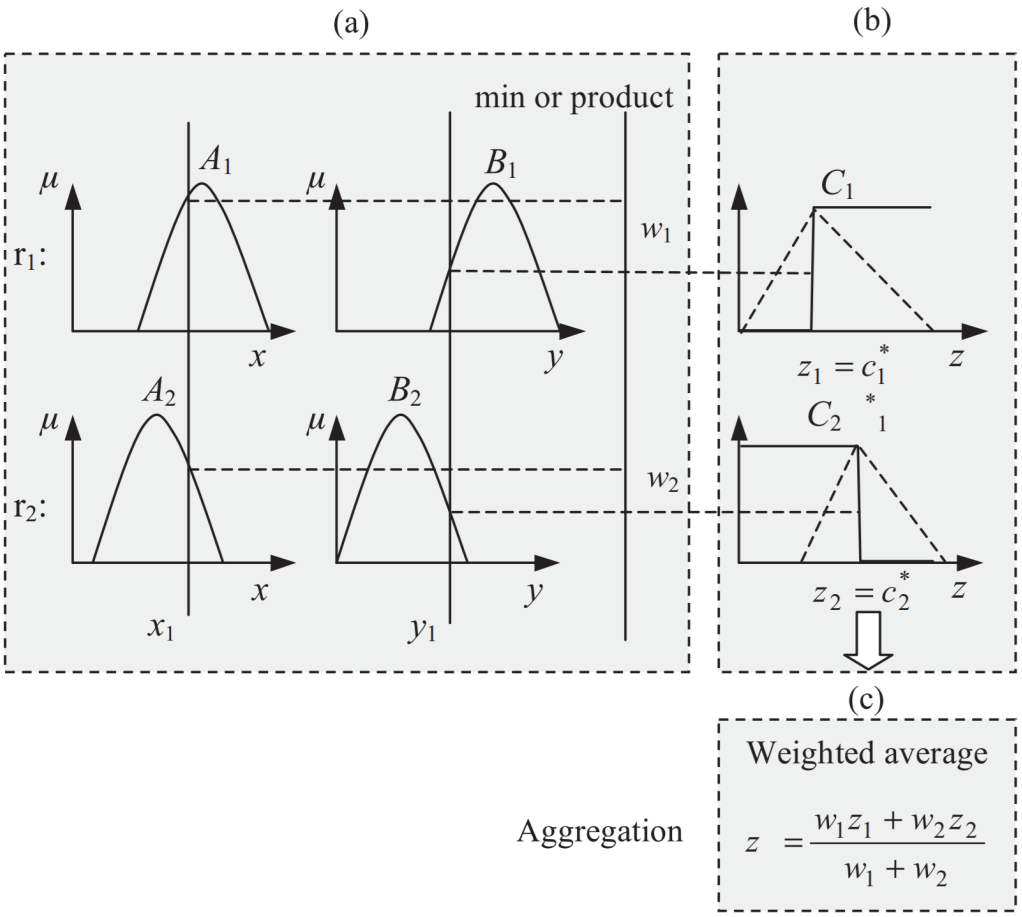


Fig. 4.8 Zero-order Sugeno fuzzy model. Reproduced from [117]

4.3.3 Comparison Between Sugeno and Mamdani Models

This section will discuss the difference between the Mamdani and Sugeno fuzzy models. It is very important to find a suitable model that can be used to solve a particular problem. Giving the main difference between these two models will allow the researchers to find the specifications and advantages of each model in order to choose the right one among them. The differences between these two fuzzy inferences, also called fuzzy models, are mainly represented by the consequent parts of their fuzzy rules, aggregations and defuzzification procedures. Table 4.1 summarizes these differences.

Table 4.1 Comparison between Sugeno and Mamdani models

		MAMDANI	SUGENO
Advantages:		• Intuitive. • Well-suited to human input. • More interpretable rule base. • Have widespread acceptance.	• Computationally efficient. • Works well with linear techniques. • Works well with adaptive techniques and optimization. • Guarantees output surface continuity. • Well-suited to mathematical analysis.
Specifications:	Fuzzification	• Generating the MFs rely on experts.	• Generates the MFs automatically.
	Inputs	• Single values.	• Data sets.
	Rules composition	• Max-mini or Max-product.	• Max-mini or Max-product.
	Consequent	• Fuzzy sets.	• Functions (usually polynomial).
	Aggregation	• Fuzzy set.	• Weighted average.
	Defuzzification	• Crisp value.	• Constant.

Chapter 5

Adaptive Neuro-Fuzzy System

This chapter introduces the theoretical framework of the adaptive neuro-fuzzy system. As the name implies, the neuro-fuzzy system can be defined as a combined model consisting of two approaches; an adaptive Artificial Neural Network (ANN) and a Fuzzy Inference System (FIS). In other words, it is a hybrid, intelligent system [96]. The concepts of the fuzzy inference system have been discussed in detail in the previous chapter. Therefore, the first part of this chapter describes the needed concepts of adaptive ANNs. Accordingly, in the second part, we present a class of adaptive networks that are combined with FIS, i.e., **Adaptive Neuro-Fuzzy Inference System (ANFIS)**. This will provide a unifying framework that subsumes almost all the needed information which includes FIS and the adaptive ANN.

In this chapter, all the figures, diagrams, and examples have been created by the author, except where indicated with the word "reproduced".

5.1 Adaptive Neural Networks

5.1.1 Introduction to ANNs

A neural network can be defined as a set of processing units (also known as *nodes* or *neurons*) that proceed by sending signals to each other along weighted connections. The way in which these units are connected depends on the specific network model [29]. Each unit can accept a number of input signals and produce one output signal. In general, inputs are combined by calculating the weighted sum of all inputs. The output is then computed by passing this weighted sum of inputs through an activation function. In general, a neural network is composed of an input layer, an output layer, and will contain any number of hidden layers between. Network topology refers to the type of network being created (inputs, hideen layers, outputs) and helps us to define the complexity of the network [151]. This varies from simple perceptrons to more complex convolution networks. The complexity of the network is controlled by the number of layers and nodes chosen and the transfer function used. The higher the number of layers and nodes, the greater the degrees of freedom in the underlying network model. This needs controlling carefully. Too few nodes will result in an under-trained network that has inferior training capabilities. Too many parameters can result in an over-trained network that has low prediction capability.

Data quality and quantity is a crucial factor. There needs to be enough data to ensure convergence of the network from the given samples. Data is usually separated into training data and test data. The network is trained on the first set and validated on the second. Variations of this approach, using n-fold cross-validation, can be employed. The data can be synthetic data providing a good understanding of the problem space is known so that data

that meaningfully represents the underlying problem can be generated and used to test the network.

5.1.2 Architecture of ANN

5.1.2.1 Neuron Architecture

Neurons, also referred to as *units* or *nodes*, can be defined as a processor that is processing input signals through an activation rule in order to calculate the output signal. Therefore, we can identify the essential elements that represent the main structure of the neuron, as shown in Figure (5.1). A single network neuron consists of:

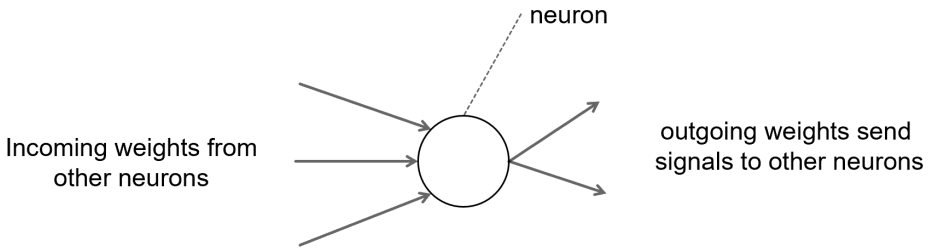


Fig. 5.1 Single neuron.

- Input signals: which represents the incoming weights from other neurons.
- Node: the summing junction where the input signals are collected and processed via an activation function (referred to as the *transfer function*) in order to produce the output. In other words, it represents the core of the neuron, which consists of two components. First is the summing junction where the sum of the product of the input signals and its corresponding weights $\sum x_i w_i$ of the $i-th$ neuron to be calculated. The second part is the transfer function $f(\sum x_i w_i)$ in which the output signal is to be computed.
- Output: which represents the calculated outgoing weights signal to be sent to other neurons (usually in the next layer).
- A threshold value: also referred to as the bias, is a fixed value added to the summing junction in order to prevent the case of when the summing is equal to zero. It plays a vital role in adapting the output weights as an additional parameter of the neuron.

Assume that we have n inputs $x_1, x_2, \ldots, x_n$ to a neuron i with corresponding weights $w_{ki} = [w_{1i}, w_{2i}, \ldots, w_{ni}]$. Thus:

$$\begin{aligned} sum_i &= (w_{1i}.x_1 + w_{2i}.x_2 + \cdots + w_{ni}.x_n) \\ &= \left(\sum_{k=1}^{n} w_{ki}.x_k \right) \end{aligned} \tag{5.1}$$

Furthermore, by adding threshold value b_i, which represents a bias:

$$sum_i = \left(\sum_{k=1}^{n} w_{ki}.x_k \right) + b_i \tag{5.2}$$

The equation 5.2 will then represents the input to the transfer function (f). Thus the output (y_i) of ($i-th$) neuron can be obtained by:

$$y_i = f(sum_i) = f\left(\left(\sum_{k=1}^{n} w_{ki}.x_k \right) + b_i \right) \tag{5.3}$$

In ANNs, the output of each neuron depends on its transfer function. We can identify two types of transfer function according the nature of its parameters, i.e., modifiable or non-modifiable parameters. Therefore, neurons can be classified into two types:

1. **Fixed neuron:**
   The ANN neuron can be defined as fixed if the node has a function with a non-modifiable parameter set, sometimes called a non-parameterized function. We use a circle to represent the fixed neuron. Figure (5.2a) shows a schematic view of this type. Here we introduce some different types of transfer function that can be used under this classification with its corresponding form:

   - Hard-limit function:

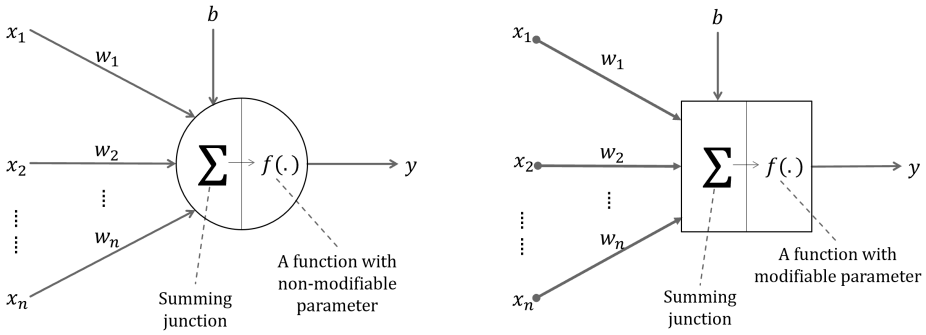
$$y = f(sum) = \begin{cases} 0 & \text{if } sum \leq 0, \\ 1 & \text{if } sum > 0. \end{cases} \tag{5.4}$$

   - Linear function:

$$y = sum \tag{5.5}$$

   - Step function:

$$y = f(sum) = \begin{cases} -1 & \text{if } sum \leq 0, \\ +1 & \text{if } sum > 0. \end{cases} \tag{5.6}$$



(a) Fixed neuron.

(b) Adaptive neuron.

Fig. 5.2 A schematic view of an artificial neuron.

2. **Adaptive neuron:**
The adaptive neuron (Fig. 5.2b) is a node whose overall behaviour is determined by a parameterized function with a set of adjustable parameters. Ordinarily, the adaptive network is composed of a set of neurons (fixed and/or adaptive) connected by directed links. These neurons are to use its functions to process incoming signals in order to generate the node output. Changing the parameters of the adaptive neurons means we modify the neuron function, which affects the overall network behaviour. We use a square to signify the adaptable neuron, Figure (5.2b) shows a schematic view of this type. Again we show some different types of parameterized functions that can be considered under this classification with its corresponding form:

- Log-Sigmoid function:

$$y = f(sum) = \frac{1}{1+e^{-a(sum)}}, \tag{5.7}$$

where a denotes the parameter of the activation function. We can clearly see that the nature of the distribution of this function depends on the value of a.

- Tan-Sigmoid function:

$$y = f(sum) = \frac{1-e^{-a(sum)}}{1+e^{-a(sum)}} \tag{5.8}$$

More types of parameterized functions with details can be found in section 3.4.2.

5.1.2.2 Network Architecture

In general, the neural network pattern assumes that the basic structure of a network is comprised of layers connected by direct links. Each layer should have at least one neuron. The layers can be divided into three types, i.e. input, hidden, and output. A composition of these three types represents the network structure or topology. Thus, the hidden layer can be more than one layer. Figure (5.3) shows a schematic view of a single layer of m neurons. Let us consider that there are n inputs connected to this layer, such as:

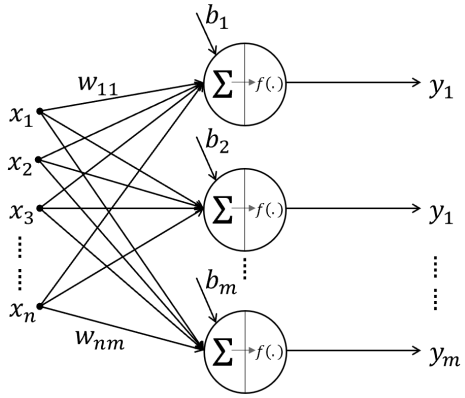


Fig. 5.3 A schematic view of a single-layer of NN.

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}.$$

The weights of the network and the biases are defined by the following weight matrix W and bias vector b simultaneously:

$$W = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1m} \\ w_{21} & w_{22} & \dots & w_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \dots & w_{nm} \end{bmatrix}, \qquad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

Then, the output of the network Y can be written in vector form as:

$$Y = f(W.x + b) \tag{5.9}$$

5.1.3 Adaptive Network Architecture

When all neurons in all layers of the network are fixed (non-modifiable), then the network can be classified as typical neural architecture. Otherwise, if the network contains at least one adaptive neuron, then it is an adaptive network. In other words, an adaptive network (Fig. 5.4) is a network whose overall behaviour of its input-output mappings is determined by a set of modifiable parameters. In most common cases, adaptive networks are heterogeneous. Every single neuron has a particular function, which often differs from the others. Links in this type of network network are only used to designate the propagation path of a neuron's outputs. In other words, the links are not accompanied by weights [61]. Each node in the adaptive network has its local parameter set. The union of these parameters represents the network's overall parameter set.

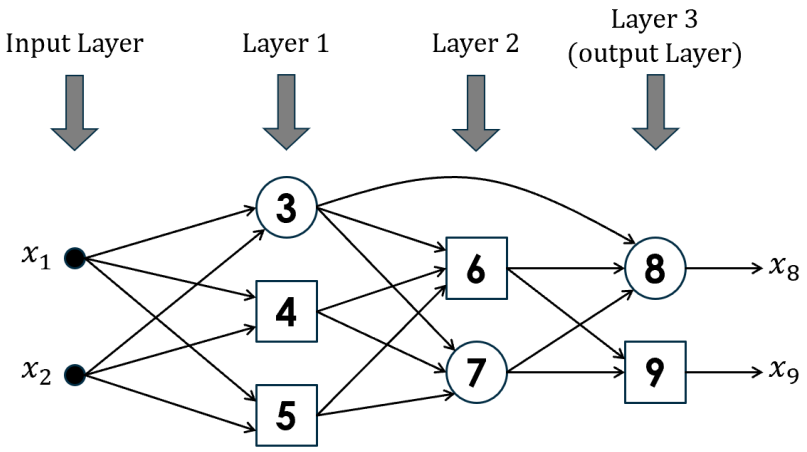


Fig. 5.4 Adaptive network in layered representation [61].

5.1.4 Feed-Forward ANN

According to the directional connectivity of the neurons, the artificial neural networks can be categorised into two classes, **feed-forward network** and **recurrent network**. The network can be considered as feed-forward if the propagation process of each individual neuron within the network follows one direction (in the forward pass) from the input (left) to the output (right). On the contrary, if there is one link (or more) that forms a feedback path in the network, then it can be considered as recurrent. There are seven different types of feed-forward neural network that are distinguished, such as:

- Multilayer perceptron networks.
- Radial basis function networks.

- Generalized regression neural networks.
- Probabilistic neural networks.
- Belief networks.
- Hamming networks.
- Stochastic networks.

Taking a different viewpoint, if we look at the structure of a feed-forward network from a graph theory side, it can be represented by an acyclic graph with one direction and without directed cycles. Whereas, a recurrent network always contains at least one directed cycle. Figure (5.4) represents a layered feed-forward adaptive network. It can be noticed that the neurons in the same layer are not linked to each other. Moreover, the outputs of all neurons in a specific layer are to be fed-forward to the succeeding layer. The input-output mapping of the adaptive feed-forward network is static; depending on the network structure, this mapping can be a linear or nonlinear relationship. Our target is to model a system that is constructed of a desirable network based on nonlinear mapping adjusted via the input-output data set. This data set is commonly named training data.

Adjusting the network parameters plays a vital role in improving the network's performance; these procedures are known as **adaptation algorithms** and also termed as the **learning rules**. Ordinarily, measuring a network's performance can be obtained by calculating the difference between the desired output and the network's output following similar input conditions; this difference is termed as an **error measure**. By applying a specific metric (and thus optimization technique) to obtain an error measure, we can derive the learning rule.

5.1.5 Supervised Learning

In general, learning (also referred to as a training algorithm) in the neural network can be defined as the procedures of updating and modifying the weights, biases, and parameters of a network. It represents the techniques that can be used to train a network and optimize a particular input-output mapping to a specific desired targeted output. Broadly, the ANN can be divided into two types of learning, i.e. either **supervised** or **unsupervised**. The ANN can be classified as unsupervised if there are no target values on which to be compared with the network outputs to determine the errors. In other words, there is neither calculation nor feed back of the predicted error in order to update the network.

In contrast, the mechanism of the supervised ANN operates on modifying its parameters and updating the weights depending on the feedback obtained from error calculations (i.e.,

the difference between the target (desired) output and the network output). In other words, the form of supervised learning is applicable if the input-output relationship of the training scenarios is obtainable. As the input is applied to the network, the forward pass will be propagated through each layer to the output layer. In the output layer the network provides its predicted value. An error can be determined by comparing the network's output with its corresponding target value. These errors then get fed back to the network in order to be used by the learning rule to adjust the network parameters and move the network outputs as close as possible to the targets. The adaptation algorithm will keep repeating the process of minimizing the current errors of all processing elements continuously in order to modify the network parameters until an acceptable global error reduction is reached. During training, the parameters of a network are optimized by a applying supervised learning rule. Figure (5.5) shows a schematic view of supervised learning.

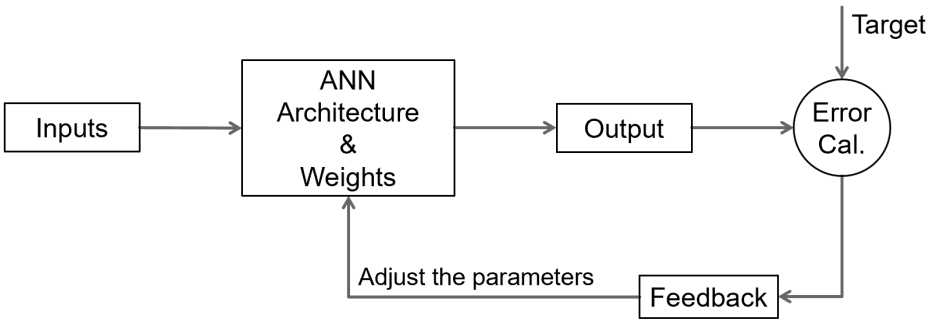


Fig. 5.5 A schematic view of supervised learning.

There are many supervised learning rules, some of which are:

- Widrow–Hoff learning rule
- Gradient descent learning rule
- Delta learning rule
- Backpropagation learning rule
- Cohen–Grossberg rule
- Adaptive conjugate gradient model of Adeli and Hung learning rule.

We will focus on introducing the required learning methods that represent the core of the structure for the adaptive neuro-fuzzy inference system that will be discussed in the next sections, such as the backpropagation learning algorithm for adaptive feedforward networks.

5.1.5.1 Backpropagation Learning Algorithm

The following explanation of the Backpropagation algorithm follows closely the description is given by Jang *et al.* [61].

Backpropagation, in its standard from, is a gradient descent algorithm. In general, the process of computing the gradient vector in a network and using this to update the network's weights is known as backpropagation. It is termed like this due to the way in which errors are propagated back from the output layer, towards the input layer. The backpropagation learning rule uses the gradient descent (also referred to as the **gradient method** or **steepest descent**) algorithm as a backward pass optimization method. This section introduces a learning rule for adaptive networks, which is, in essence, the gradient descent method. The core of the gradient descent learning rule concerns how to use the chain rule to find a gradient vector in which each element is defined as the derivative of an error measure with respect to a parameter. Once the gradient vector is obtained, the parameters can be updated via several regression techniques and derivative-based optimization.

Assume that we have a feed-forward, adaptive neural network as shown in Figure (5.6) which has L layers. The network has an $N(l)$ neurons in layer l (i.e. $l = 0, 1, \ldots, L$; where $l = 0$ is the input layer). Then the output of node i $[i = 1, 2, \ldots, N(l)]$ in layer l can be denoted as $O_{l,i}$ and its corresponding function as $f_{l,i}$. As mentioned, the output of neurons in the adaptive networks relies on two main factors. The incoming signals and the set of parameters which accompaniment to its function. Therefore, the output of a neuron can be expressed as:

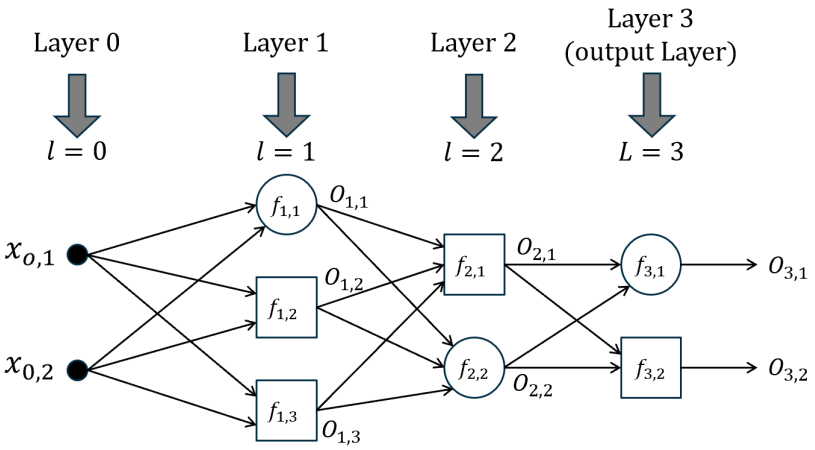


Fig. 5.6 A feedforward adaptive neural network.

$$O_{l,i} = f_{l,i}(O_{l-1,1},\ O_{l-1,2},\ \ldots, O_{l-1,N(l-1)},\ \alpha,\ \beta,\ \gamma,\ \ldots) \tag{5.10}$$

where $\alpha,\ \beta,\ \gamma, \ldots etc.$ are the neuron's functions parameters.

Suppose we have D entries in our data set, then the error measure for the d-th entry (where $1 \leq d \leq D$) can be defined as the sum of squared errors as follows:

$$E_d = \sum_{k=1}^{N(L)} (t_k - O_{L,k})^2 \tag{5.11}$$

where t_k is the k-th element of the d-th targeted (desired) output vector. while $O_{L,k}$ is the k-th actual output component resulting from processing the d-th input vector forwards through the network. The network then can achieve its goals when E_d is equal to zero, which means, the network's output vector is equal to the target output vector in the d-th training data.

In order to proceed to the minimization of the error measure by using gradient descent, we first need to obtain the gradient vector. Prior to computing the gradient vector, the following causal relationships of a network must be observed:

modifying parameter α	$\Longrightarrow$	effecting neurons outputs (containing α)	$\Longrightarrow$	effecting the outputs of the network	$\Longrightarrow$	change in error measure

where the symbol $\Longrightarrow$ refers to the causal relationships. This shows that even a small modification applied to the parameter α can affect the output of all neurons which contain this parameter. As a result, the overall outputs, including the output of the final layer, will be affected. This, in turn, will affect the training accuracy of the network. This explains the concept behind the calculation of the gradient vector, which can be applied by feeding back a form of the derivative, beginning at the output layer and then passing through all layers until reaching the input layer.

Let us assume that the error signal is denoted as $\delta_{l,i}$ (concerning the output of node i in the layer l) which represents the derivative of the error measure E_d. Thus:

$$\delta_{l,i} = \frac{\partial^+ E_d}{\partial O_{l,i}} \tag{5.12}$$

This expression is called **ordered derivative** [145]. This type of derivative is different from the typical partial derivative with respect to the way of the function's differentiation. This can be explained as follow, if we consider the output of an internal neuron $O_{l,j}$ (where $l \neq L$), the partial derivative $\frac{\partial E_d}{\partial O_{l,i}}$ is equal to zero, where E_d does not depend directly on $O_{l,i}$. However, it is clear that E_d depends indirectly on $O_{l,i}$, because if the output of an internal neuron i in layer l, i.e. $O_{l,i}$ has been changed and propagate through indirect paths towards the output

layer; this will provide a similar change in E_d. Consequently, $\delta_{l,i}$ represents the ratio of these two changes when they are made infinitesimal.

- **Difference between the ordered derivative and the ordinary partial derivative:** Before we proceed with the explanation of the differentiation of the error measures, let us discuss the difference between the ordered derivative and the ordinary partial derivative in more details from a mathematical viewpoint. Assume that we have a

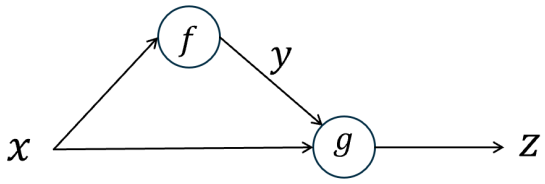


Fig. 5.7 Simple ordered derivative adaptive network. Reproduced from [61]

simple adaptive network with two nodes as its illustrate in figure 5.7, where z is a function of x and y, while, y is a function of x. Thus:

$$\begin{cases} z = g(x,y), \\ y = f(x). \end{cases}$$

Consider the ordinary partial derivative $\frac{\partial z}{\partial x}$; it has been assumed that all other input variables (such as y) are constant. Thus:

$$\frac{\partial z}{\partial x} = \frac{\partial g(x,y)}{\partial x}.$$

This means by using the ordinary partial derivative; we assume that the function g has two independent inputs variables such as x and y, and ignoring the actual fact of y is a function of x. Whereas, the ordered derivative considers this indirect causal relationship. Thus:

$$\begin{aligned} \frac{\partial^+ z}{\partial x} &= \frac{\partial g(x,f(x))}{\partial x} \\ &= \frac{\partial g(x,y)}{\partial x}\bigg|_{y=f(x)} + \frac{\partial g(x,y)}{\partial y}\bigg|_{y=f(x)} \frac{\partial f(x)}{\partial x}. \end{aligned}$$

This explains the benefit of use ordered derivative, which takes direct and indirect paths into account when dealing with the causal relationship.

Therefore, calculating the error signal at layer L can be extracted directly using the following expression:

$$\delta_{L,i} = \frac{\partial^+ E_d}{\partial O_{L,i}} = \frac{\partial E_d}{\partial O_{L,i}} = -2(t_i - O_{L,i}). \tag{5.13}$$

Whereas, the error signal for the i-th internal neuron of the l-th layer can be derived by using the chain rule. Thus:

$$\delta_{l,i} = \underbrace{\frac{\partial^+ E_d}{\partial O_{l,i}}} = \sum_{n=1}^{N(l+1)} \underbrace{\frac{\partial^+ E_d}{\partial O_{l+1,n}}} \frac{\partial f_{l+1,n}}{\partial O_{l,i}} = \sum_{n=1}^{N(l+1)} \delta_{l+1,n} \frac{\partial f_{l+1,n}}{\partial O_{l,i}}, \tag{5.14}$$

where the expression $\frac{\partial^+ E_d}{\partial O_{l,i}}$ represents the error signal at layer l, and $\frac{\partial^+ E_d}{\partial O_{l+1,n}}$ represents the error signal at layer $l+1$. Also $0 \leq l \leq L-1$. This means, the error signal arriving into an internal neuron at layer l is obtained as linear combination propagated back from the neuron at layer $l+1$. Therefore, by applying Equation (5.13) we can find the error signals of the output layer L, and then applying Equation (5.14) in order to find the error signals for all i-th neurons in l-th layer. This has to be run repeatedly until the target output is reached. Obtaining the error signals sequentially and processing them, starting from the output layer towards the input layer, is referred to as *back-propagation*.

As mentioned earlier, the chain rule has been applied to find the derivative of the error measure with respect to each parameter, which is referred to as the gradient vector. Assume that α represents a parameter at the ith neuron at layer l. Thus:

$$\frac{\partial^+ E_d}{\partial \alpha} = \frac{\partial^+ E_d}{\partial O_{l,i}} \frac{\partial f_{l,i}}{\partial \alpha} = \delta_{l,i} \frac{\partial f_{l,i}}{\partial \alpha}. \tag{5.15}$$

In some cases, the parameter α can be shared among different neurons; thus Equation (5.15) is to be expressed into the more general form:

$$\frac{\partial^+ E_d}{\partial \alpha} = \sum_{O^* \in S} \frac{\partial^+ E_d}{\partial O^*} \frac{\partial f^*}{\partial \alpha}, \tag{5.16}$$

Where S represents the set of neurons which contains the parameter α; f^*, and O^* are the function and output, sequentially, for the generic neuron in S.

The derivative of E (the overall error measure) with respect to α is:

$$\frac{\partial^+ E}{\partial \alpha} = \sum_{d=1}^{D} \frac{\partial^+ E_d}{\partial \alpha}. \tag{5.17}$$

Therefore, if the parameter α comes under a straightforward gradient descent without line minimization, then its formula can be written as:

$$\Delta\alpha = -\eta \frac{\partial^{+} E}{\partial \alpha}, \tag{5.18}$$

where η represents the **learning rate**, this can be further expressed as:

$$\eta = \frac{k}{\sqrt{\sum_{\alpha} \left(\frac{\partial E}{\partial \alpha}\right)^2}}, \tag{5.19}$$

in which k represents the **step size**. The step size can be defined as the length of every transition within the gradient direction with respect to the parameter range.

5.1.6 Hybrid Learning Rule

As the name implies, hybrid learning can be defined as a combination of two optimization learning technique. Generally, in order to identify the parameters of an adaptive network, we can use the gradient descent optimization method (i.e. the backpropagation learning rule). However, we have to take into account that this method does (in some cases) take more time to converge compared with other methods. Therefore, in 1990's, Jang [59] [60] introduced a hybrid model comprised of two combined methods, i.e. the **gradient descent (GD)** and **least-squares estimator (LSE)**. This hybrid model can speed up the processing time of the parameter's identification when the output of a network is linear. Thus, if some of the network's parameters are linear, then it can be trained by using the linear least-squares method. In order to explain this, first, we need to introduce two types of learning models. That is, (**off-line** and **on-line**) learning. The main difference between the two types lies in the method of updating the parameters. Off-line learning takes place only after processing all the training data set. In other words, it can be modified after a full epoch has taken place. It can be presented based on Equation (5.17); this type is also referred to as batch learning. Whereas, the on-line learning mechanism lies in pattern-by-pattern learning. Consequently, the parameters are to be modified instantly after processing each input-output data pair, iteratively. The processing procedure can be performed using the formula in Equation (5.15).

Since we are interested in presenting the essential concept in which will be used in ANFIS, therefore, the off-line learning will be discussed in more details as follow:

5.1.6.1 Off-Line Learning

Suppose we have an adaptive network producing a singular output. Thus:

$$O = F(\mathrm{i}, P), \tag{5.20}$$

where F represents the overall function that been used to perform the network's rules, **i** represents the input variables vector, and P is the parameters set. Assume that some of the parameters included in P are linear, thus if a function is to be used, such that the composite of H with F ($H \circ F$) is leaner in any element of P, then we can modify these parameters using the least-square method [61]. Therefore, the overall parameters set P is to be split into two parts:

$$P = P_1 \oplus P_2, \tag{5.21}$$

(where $\oplus$ is a direct sum) if the component of P_2 is linear, then $H \circ F$ represents the linear relationship within the parameters in P_2, and by applying H to Equation (5.20) we have:

$$H(O) = H \circ F(x_i, P), \tag{5.22}$$

where $H(O)$ linear with respect to the parameters in P_2. By assuming that the value of the elements of P_1 are given, we then able to process D training data using Equation (5.22) and produce the following equation in matrix form:

$$A\theta = Y \tag{5.23}$$

where θ in unbeknown vector in which all its components are parameters in P_2. It can be clearly noticed that Equation (5.23) is identical to least square estimation form. This means that the problem we are discussing is a standard linear least-squares. However, we can find the solution with respect to θ by minimizing $\|A\theta - Y\|^2$, which represent what is known as the least-square estimator (LSE) θ^*:

$$\theta^* = (A^T A)^{-1} A^T Y, \tag{5.24}$$

here we have the term A^T represents the transpose of A. If $A^T A$ is non-singular, then $(A^T A)^{-1} A^T$ is the pseudo-inverse of A. Considering Equation (5.23), if we assume that the i-th row of A to be a_i^T and the i-th element of y to be y_i^T, respectively, then we can iteratively

compute θ By:

$$\begin{aligned} \theta_{i+1} &= \theta_i + D_{i+1} a_{i+1} (y_{i+1}^T - a_{i+1}^T) \\ D_{i+1} &= D_i - \frac{D_i a_{i+1} a_{i+1}^T D_i}{1 + a_{i+1}^T D_i a_{i+1}} \end{aligned} \quad , \tag{5.25}$$

where $i = 0, 1, \ldots, D-1$, D is the entry data set, and θ^* is equal to θ_D. However, in order to start processing Equation (5.25), we initially need to set a satisfied startup value such as $\theta_0 = 0$ and $D_0 = \gamma \mathbf{I}$, where γ is a large positive number and $\mathbf{I}$ is an $M \times M$ identity matrix.

At this stage, the parameters of an adaptive network can be optimized by a combination of LSE and GD. The two combined methods represent the core of the hybrid learning that can be presented in an off-line (batch) learning, where each epoch consisting of a **forward-pass** and **backward-pass**, respectively. By applying the forward-pass, we are processing the network input vector through the network's layers towards the output layer. The calculation of each neuron's outputs in every layer will produce corresponding values for a single row in A and Y matrices in Equation (5.23), then we can iteratively compute all rows using all the training data pairs in order to complete A and Y. This will allow us to determine all parameters in P_2 by using either Equation (5.24) or Equation (5.25). By completing forward-pass the P_2 is determined, then we can identify the network output vector to be compared in its turn with the desired output in order to compute the error measure.

Accordingly, the backwards-pass can use the derivative of the error measure to propagate the error signals in a backward path starting from the output layer toward the input layer [see Equations (5.13) and (5.14)]. Consequently, after using all training data, we reach the end of the backwards-pass and all the parameters in P_1 are updated using GD method [see Equation (5.18)]. The hybrid learning has some advantages that can be summarised as follows:

- It is decreasing the search space dimension that initially has been explored by the GD method.
- It is reducing the convergence time.
- Using the squared error measure will ensure the global optimum point in the space of P_2 parameters by fixing the values of P_1 parameters.

5.2 Adaptive Neuro-Fuzzy Inference System (ANFIS)

5.2.1 Introduction to ANFIS

In the previous section, the concept of adaptive networks has been discussed from the viewpoint of structure and adaptation algorithms. Functionally, there are two constraints that can restrict the configuration of an adaptive network. Firstly, the piece-wise differentiability of the neuron's functions, and secondly by compelling it to use the feedforward network type. These minimal constraints have given the adaptive networks the chance to be used in a wide variety of applications.

In this section, we introduce the **adaptive neuro-fuzzy inference system (ANFIS)**, which can be defined as a type of adaptive network that is equivalent to FIS in its functional manner. The adaptive neuro-fuzzy inference system can be considered as an intelligence approach which consists of a combination of adaptive artificial neural networks and a fuzzy inference system. Therefore, ANFIS has the advantages of both methods and the ability to formulate an input-output mapping using specific data pairs based on if-then rules as well as suitable membership functions. ANFIS was first introduced in the 1990s by Jang [59] [60] [61]; he proposed the methods of combining different neural network techniques with the fuzzy inference system (FIS).

Technically, ANFIS is a supervised multi-layer feed-forward adaptive network combined with a Sugeno-type fuzzy model. There are two main approaches which represent the basis of ANFIS; this has given it the advantage of using both numerical and linguistic knowledge. Since we are using first order Takagi-Sugeno type fuzzy models, the fuzzification process represented by the extraction of the rule and membership functions' shape can be determined using a given input-output data set without relying only on expert knowledge. Usually, the membership functions come with adjustable parameters. For the framework of adaptive learning, ANFIS uses the hybrid learning process consisting of a combination of backpropagation (gradient descent) and the least-squares estimator algorithm. This will allow FIS to learn from the data, which plays an important role in identifying the rules depending on actual values within the fuzzy logic. Therefore, ANFIS can be considered as one of the most used systems for parameters estimation of the complex systems [10]. Table (5.1) illustrates ANFIS features.

ANFIS can be considered as one of the most superior intelligence techniques that are capable of dealing with fuzziness, complexity, uncertainty, adaptation capability, non-linearity, ambiguity, and rapid learning capacity; particularly when high precision and reliability in prediction problems is required [92] [168]. The features of ANFIS are compared with other methods to show its advantages over these models. Table (5.2) illustrates this comparison.

Table 5.1 ANFIS features.

Description	Specification
Network type	multi-layer feed-forward adaptive network
Learning type	Supervised learning
Network nature	Dynamic
Learning paradigm	Off-line (Batch learning)
Learning algorithm	Hybrid learning process
Forward learning rules	Least-square estimator (LSE)
Backward learning rules	Gradient descent (GD)
Training process	Input-output data set
Fuzzy model	Takagi-Sugeno type fuzzy model
Inference engine process	Given input-output data
Output	Either constant or linear

5.2.2 ANFIS Structure

Assume that the FIS under consideration has two inputs x_1 and x_2 and one output Z. For a first-order Sugeno fuzzy model, the rule-base contains two fuzzy if-then rules - generally denoted as in the following form:

$$\begin{aligned} &\text{Rule1: If } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } B_1 \text{ ,then } f_1 = p_1x_1 + q_1x_2 + r_1, \\ &\text{Rule2: If } x_1 \text{ is } A_2 \text{ and } x_2 \text{ is } B_2 \text{ ,then } f_2 = p_2x_1 + q_2x_2 + r_2. \end{aligned}$$

A_i and B_i are the fuzzy sets in the antecedents part, and f_i is the output set in the consequence part within the fuzzy region specified by the fuzzy rule. The p_i, q_i and r_i are the design parameters that are determined during the training process. Five layers represent the ANFIS structure, which is considered to be a feed-forward adaptive neural network. Figure (5.8) shows an example ANFIS structure containing the five layers with two input variables.

(i) **Layer 1**: Every i-th neuron in this layer is a neuron function that can be adapted. The output of this layer can be denoted by:

$$\begin{aligned} O_{1,i} &= \mu_{A_i}(x_1), \qquad \text{for } i = 1,2 \\ O_{1,i} &= \mu_{B_{i-2}}(x_2), \qquad \text{for } i = 3,4. \end{aligned} \tag{5.26}$$

Table 5.2 ANFIS features compared to other methods. Reproduced from [92].

	Features						
Model	Data complexity and non linearity	Data uncertainty and non crisp data set	Intelligent modelling and forecasting	Fuzzy data modelling	High precision and reliability	Dealing with ambiguity	Data Pre-Post Processing
ANFIS	✓	✓	✓	✓	✓	✓	✓
Genetic Algorithm	✓			✓	✓	✓	
ANN	✓		✓		✓		
Nonlinear Regression	✓		✓		✓	✓	
Decision Tree	✓			✓	✓	✓	
Linear Regression	✓				✓		
Fuzzy Regression						✓	

The term $O_{1,i}$ represents the membership grade of a fuzzy set $A' = \{A_1, A_2, B_1, B_2\}$ and it determines the degree to which the given input x_1 or x_2 satisfies the quantifier A'. The membership function for A' can be any parametrized membership function such as the generalized bell function for example:

$$\mu_{A_i}(x_1) = \frac{1}{1 + \left|\frac{x_1 - c_{A_i}}{a_{A_i}}\right|^{2b_{A_i}}}, \quad \mu_{B_{i-2}}(x_2) = \frac{1}{1 + \left|\frac{x_2 - c_{B_{i-2}}}{a_{B_{i-2}}}\right|^{2b_{B_{i-2}}}}. \tag{5.27}$$

$\{a_{A_i}, b_{A_i}, c_{A_i}\}$ when $i = 1,2$ and $\{a_{B_{i-2}}, b_{B_{i-2}}, c_{B_{i-2}}\}$ when $i = 3,4$ represents the parameter set. Changing these parameter values will affect the bell function's shape. Parameters in this layer are referred to as *premise parameters*.

(ii) **Layer 2**: The neurons in this layer are fixed and labelled T_i. Their output (firing strength) is the product of all the incoming signals:

$$O_{2,i} = w_i = \mu_{A_j}(x_1)\mu_{B_j}(x_2), \qquad j = 1,2 \tag{5.28}$$

Each neuron's output represents the firing strength of a rule. In general, any other T-norm operators that perform fuzzy **AND** can be used as the neuron function in this layer.

(iii) **Layer 3**: The neurons in this layer are fixed nodes labelled N. The i-th neuron calculates the ratio of the i-th rule's firing strength to the sum of all rules firing

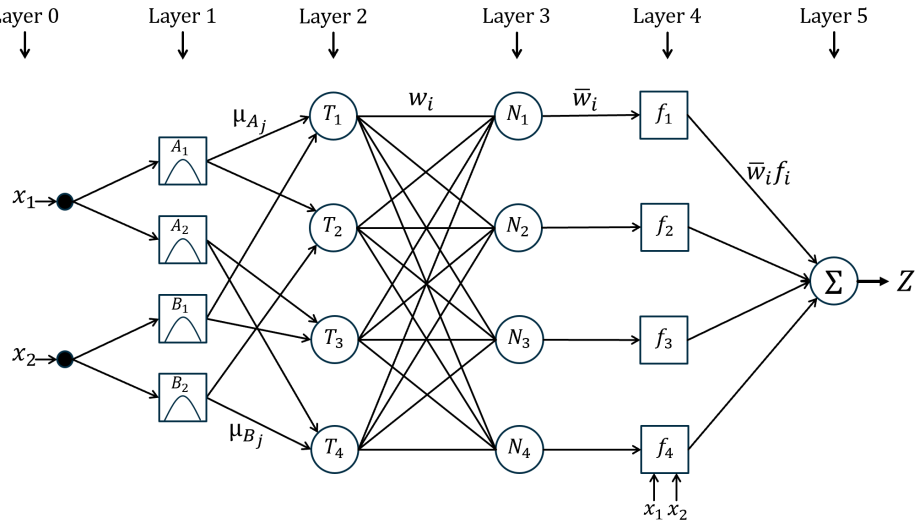


Fig. 5.8 ANFIS structure for the Sugeno fuzzy model.

strengths:

$$O_{3,i} = \overline{w}_i = \frac{w_i}{w_1 + w_2}. \tag{5.29}$$

The outputs of this layer are called normalized firing strengths.

(iv) **Layer 4**: Each neuron i in this layer is an adaptive neuron with a neuron function:

$$O_{4,i} = \overline{w}_i f_i = \overline{w}_i (p_i x_1 + q_i x_2 + r_i), \tag{5.30}$$

$\overline{w}_i$ is a normalized firing strength from Layer 3 and $\{p_i, q_i, r_i\}$ is the parameter set of the i-th neuron. In this layer, the parameters are called *consequent parameters*.

(v) **Layer 5**: There is one single neuron in this layer, labelled $\sum$. It computes the overall output as the summation over all incoming signals:

$$\text{overall output } (Z) = O_{5,i} = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{5.31}$$

Table 5.3 Two passes of ANFIS hybrid learning algorithm.

	Signals	**Premise parameters**	**Consequent parameters**
Forward Pass	Neurons output	Fixed	Least-squares estimator
Backward Pass	Error signals	Gradient Descent	Fixed

The ANFIS models of the Sugeno-type, constructed for the problems discussed here, are solved using a hybrid learning algorithm. A combination of backwards-error propagation, using Gradient Descent, is combined with Least-squares estimation. In other words, there are two learning strategies that have been used in order to train ANFIS; a) learning of antecedent MFs and b) the consequent parameters. The Gradient Descent method is used to adapt the premise parameters at Layer 1, while the Least-squares estimator method is used to identify the consequent parameters at Layer 4. The hybrid learning algorithm can be simply divided into two passes, a *forward pass* and a *backward pass*. In the forward pass, the node outputs are propagated forward to Layer 4, where the Least-squares estimator is used to solve for the consequent parameters. In the backward pass, the error signals are propagated backwards in order to update the premise parameters using the Gradient Descent method. Table (5.3) demonstrate the activities of each pass.

PART 3

Models Development & Practical Implementation

Chapter 6

Data Expansion Model for ANFIS Optimization: Proposed Model 1

6.1 Introduction

The conceptual framework of the adaptive neuro-fuzzy inference system (ANFIS) architecture and its relative approaches have been introduced in the previous part [Chapters (3 to 5)]. We described the mechanism of combining a fuzzy inference system (Sugeno-type) with an adaptive neural network ANN in order to provide ANFIS. Typically, an ANFIS structure contains 5 layers (detailed in Section (5.2.2)), the neuron's functions at the second and fourth layers are to be adaptable. Since the network is a feed-forward type, it uses the backpropagation algorithm to optimize the network parameters. The parameters at the first layer are referred to as premise parameters. While the parameters at the fourth layer are termed consequent parameters. A hybrid learning algorithm which includes two passes (forward and backwards) is to be used to optimize the parameters of the membership functions (MFs) (which determines their shape) as part of the problem-solving process. On the forward pass, the least-squares estimator (LSE) method will be used to update the consequent parameters at layer 4. Whereas, the gradient descent (GD) method is to be employed on the backward pass to optimize the premise parameters at layer 1.

Functionally, the inference engine of the Sugeno-type fuzzy model and the parameter's modification process of the hybrid learning algorithm, both are using the input-output data sets to process ANFIS. It means, when using ANFIS, the quality and quantity of the given data sets play a very important role in solving the problem. ANFIS can be considered as one of the best techniques in dealing with prediction problems which have fuzziness, complexity and uncertainty in their nature. Nevertheless, this can only be true if the data quality and quantity are good and enough. However, the scarcity of the data raises concerns when attempting to construct models of this type.

When data sets are large, there is usually sufficient coverage of the problem space to ensure that both the premise parameters and consequent parameters can be resolved optimally. When there are small amounts of data however, the data may not capture the problem well and, as a result, over-fitting can occur. Even though ANFIS is theoretically known to be a universal approximator [61], training them accurately on small data sets is a significant problem in practice.

In ANFIS modelling, problems often arise when the number of samples used for training is significantly less than the number of modifiable parameters in the model. Even for simple models, with only a small number of membership functions, the total number of modifiable parameters (linear and non-linear) can be much higher than the number of data samples. When this occurs, the system has a high tendency of over-fitting, which means the residual errors on the training data are very small but the prediction error is very large. However, prediction errors are largely due to the system's inability to generalise well to the unseen data

on which a prediction is required. N-fold cross-validation can help to a point, given that the sample data is often reduced into training and test sets, but when the total samples of data available is low, this does little to solve the problem.

When data are scarce, the literature suggests that the number of membership functions should be fixed in advance, together with their premise parameters (thus fixing the number and initial shape of the MFs) to reduce the likelihood of over-fitting [163]. Therefore, when data is scarce, the premise parameters are fixed, and only the consequent parameters, linking the decision rules together, are optimised by the learning algorithm.

One way to overcome the challenge of having a small data set is to find a way of expanding the data by capturing its distributional properties. Though oversampling is reasonably well understood for classification tasks (SMOTE and its variants for example [38]), it not so well understood for regression-type problems. This work will show that by replacing the discrete training data with a carefully chosen and carefully optimised continuous model, the new model can be re-sampled, and from it, a finer granularity of data obtained. The expanded data can be used to improve prediction accuracy for an ANFIS model.

In this chapter, the equations (6.1 to 6.6) and the equations (6.11 to 6.12), in additional to all of the work, tables, figures, and diagrams have been created and developed by the author.

6.2 Development of Model 1

6.2.1 Model 1 Structure

The main goal of this work is to develop a model that can deal with the data scarcity problem in ANFIS modelling. We are proposing a combined model to improve ANFIS performance in dealing with low numbers of data samples. This can be achieved by producing a carefully optimised continuous function which can be used to reliably generate input data as an alternative to the original discrete given data. Referring to Figure (5.8), the ANFIS structure contains five active layers (1 to 5) preceded by the input layer labelled as (0) layer. This work will propose two combined models; thus, the proposed ANFIS model will have two parts, pre-processing model at layer 0 will represent the first part, and ANFIS with the layers 1 to 5 represents the second part.

- **Pre-processing model:**
  In this part of the proposed model, we intend to employ layer 0 as a pre-processing layer rather than an input layer only. The structure of this part is composed of two sub-models, i.e. re-sampling and scaling models. This works sequentially. Firstly, for the re-sampling model, we use radial basis function (RBF) interpolation approaches

(Linear, Cubic, and Multiquadric) as a data expansion model. These methods can be used to expand (by re-sampling) the raw data into a new data set in order to have enough training data samples to allow the learning algorithm to optimize the premise parameters in layer 1. With the proposed model creating more data within the original data space, this gives ANFIS the ability to optimize the premise parameters and fine-tune the membership functions. It will overcome the over-fitting problem that can occur if the total number of samples in the training set is less than or equal to the number of model parameters [88] being optimized.

Secondly, for the scaling model, we used Chebyshev transformations and zero mean methods in order to deal with re-scaling the data if it has a significant variance of its values among the inputs [52].

- **ANFIS model:**
  This part will represent the ANFIS architecture which contains five layers that are combined with the pre-processing model. The first layer is the fuzzification layer, where membership functions and their parameters are to be generated. The second and third layers are to determine the structuring and normalization of the rule strengths, respectively. The fourth layer is to determine the consequent parameters of the rules. Consequently, layer five will compute the overall output as a summation of all incoming signals. It has been obvious that the first and fourth layers are the parameterized layers to be modified by the hybrid learning algorithm. The premise parameters (Layer 1) and the consequent parameters (Layer 4) are optimized using the Gradient Descent method and Least-squares estimator method, respectively. It can proceed if the ANFIS model has enough data in which to converge reliably.

Figure (6.1) illustrates the different steps performed to implement the proposed model, where Layer 0 acts as a pre-processing data layer. The flowchart of this layer shows the process of the proposed pre-processing model. The first step is to determine whether the number of samples in the data sets D are enough to process ANFIS or not. At this stage, the size of the entry data sets D, and the total parameter number P are to be compared. It can be done after determining the total number of model parameters P, which can be found as follows:

$$P = P_1 + P_2, \tag{6.1}$$

where P_1 and P_2 represents the number of premise and consequent parameters, respectively. It can be noticed that Equation (6.1) is identical to Equation (5.21). It means we are following the hybrid learning rules by defining the MF's parameters (non-linear) as the premise parameters and the first-degree polynomial function (linear) as the consequent parameters.

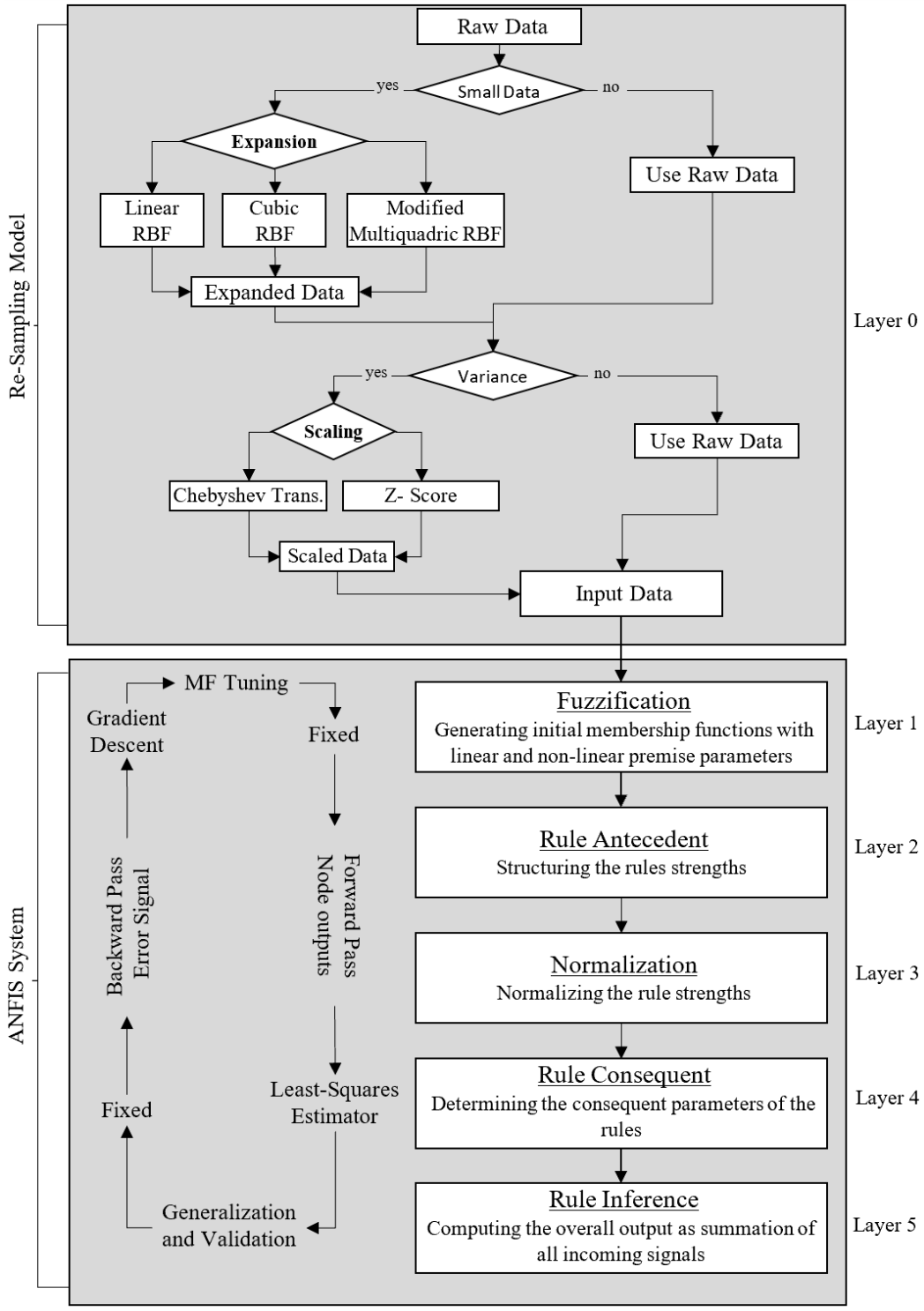


Fig. 6.1 The first proposed model.

However, P_1 can be calculated by adding the parameters of each membership function that

been assigned to the inputs using the following formulas:

$$P_1 = \sum_{i=1}^{n} num_{\mu_A(x_i)} \times a_i \tag{6.2}$$

where n represents the number of inputs, $num_{\mu_A(x_i)}$ is the number of MFs (fuzzy sets) assigned to input i, and a_i is the number of non-linear parameters of a specific MF type for input i.

Example 6.1 Assume that we have an ANFIS model containing three inputs, i.e. (x_1, x_2, and x_3) each input has **two** MFs. The types of these MFs are (Triangular, Gaussian, Triangular), respectively. Referring to Equation (3.25) which represents the general formula of the Triangular function, we can find that it has three parameters. Also, Equation (3.28) showed that the Gaussian function has two parameters. Then, by using the Equation (6.2) we find that:

$$P_1 = (2 \times 3) + (2 \times 2) + (2 \times 3) = 16$$

The consequent parameters P_2 can be computed by multiplying the total number of rules R by the number of coefficients C of the linear polynomial function corresponding to each rule. The total number of rules can be extracted using the following formula:

$$R = \prod_{i=1}^{n} num_{\mu_A(x_i)}. \tag{6.3}$$

In order to find C we need first to demonstrate the general formula of the linear polynomial function, thus:

$$f_j(x) = \left(\sum_{i=1}^{n} p_{ij}.x_i \right) + q_j, \qquad \text{for } j = 1, 2, \ldots, R \tag{6.4}$$

where p_{ij} represents the coefficient of the i-th input in the j-th rule , and q_j is the additive constant. Consequently, C is computed by:

$$C = n + 1, \tag{6.5}$$

Then we can find P_2 as follows:

$$P_2 = R \times C \tag{6.6}$$

Referring to Example (1) discussed above, we can find that:

$$\begin{aligned}&\text{Total number of rules } (R) = 2 \times 2 \times 2 = 8,\\ &\text{Number of coefficients } (C) = 3 + 1 = 4,\\ &\text{Number of consequent parameters } (P_2) = 8 \times 4 = 32.\end{aligned}$$

The total number of parameters P will be:

$$P = 16 + 32 = 48.$$

If $D \leq P$ then the data set is too small and we are facing an over-fitting problem [88]. However, to avoid the risk of over-fitting, as well as the suggestion of fixing the premise parameters and restricting our approach to finding only the consequent parameters [163], then we employ one of the three suggested data expansion methods.

After that, we move into another screening point to deal with the problem of data entry that may have to differ in its orders of magnitude between different inputs. At this point, the model will determine the needed action and process the right path in dealing with it. Two actions are available, either pass the data directly into ANFIS (if no scaling is needed), or move into the scaling model which has two scaling approaches. One of these scaling approaches (i.e. the Chebyshev transformation method or the zero mean method) will be performed in order to process the data.

The task of the first part of the flowchart shown in Figure (6.1 - layer 0) is to pass the data into ANFIS after pre-processing it in order to improve the prediction performance. The second part illustrates the layers from 1 to 5. It represents the ANFIS structure where the hybrid learning algorithm will be applied. It will gain the optimal model, which contains the best values of the premise parameters for the fine-tuned membership functions' shape that can be used for future prediction.

6.2.2 Proposed Data Expansion Model

6.2.2.1 Data Expansion - Proposed Models

In this section, we describe how each set of data (inputs and outputs), are expanded by fitting a continuous radial basis function (RBF) model to the discrete samples and then re-sampling the models to obtain a larger, but representative, data set. The proposed method below is to be applied to each discrete sample in turn.

We assume that each discrete set of data $\{(x_i, y_i)\}_{i=1}^{m}$ takes the form $y = f(x)$, where x and y are known and f is usually unknown. Our aim is to accurately approximate f by finding coefficients $c_1, c_2, \ldots, c_n$ such that

$$y_i = \sum_{j=1}^{n} c_j \phi_j(x_i), \qquad \text{for} \quad i = 1, 2, 3, \ldots, m, \tag{6.7}$$

which, for simplicity, we write as $y_i = F(x_i)$. By enforcing the conditions (6.7) we can solve a system of linear equations of the form

$$A\mathbf{c} = \mathbf{y}, \qquad \text{where} \qquad A_{i,j} = \phi_j(x_i), \tag{6.8}$$

which is square if $m = n$. We explore the effectiveness of this approach when ϕ takes the forms

$$\phi(x) = r \qquad \text{Linear RBF} \tag{6.9}$$

$$\phi(x) = r^3 \qquad \text{Cubic RBF} \tag{6.10}$$

$$\phi(x, \rho) = (r^2 + \rho^2)^{\frac{1}{2}} \qquad \text{Multiquadric RBF,} \tag{6.11}$$

where

$$r = \|x - x_j\|_2$$

is the Euclidean distance between the data point x and the function centres x_j and ρ is a scalar [115]. We have deliberately chosen these functions for their desirable properties.

The linear RBF is shape-preserving, and so there is less chance that undesirable properties, such as over-smoothing, will affect the ability to construct accurate expansion models. This function is particularly useful when no knowledge of the underlying smoothness proprieties of the data is known. A simple straight-line fit is returned fitting the convex hull of the data set.

The cubic RBF is known to be the best interpolating function to uni-variate data in terms of minimising a certain variation measure [28], and so is included to minimise over-fitting and remove the need for a regularisation term. When fitting smooth functions to data, it is important that the reconstructed curve has good generalisation properties. The properties of cubic radial basis functions ensure that the fitted model will have minimum variation (i.e., *be flatter*) over the range of the data and is, therefore, more reliable than say polynomial models which suffer considerably from over-fitting. It can be easily seen in the polynomial interpolation case where the degree of the fitting function needs to be very high to meet the interpolation conditions (6.7) - causing significant "wobble" in the underlying fit. Moreover,

without careful consideration of the degree of the fitting function, or how the abscissa domain is scaled, the polynomial will very likely not yield a model at all due to significant ill-conditioning of the resulting linear system. Alternates such as Chebyshev polynomials or piece-wise splines can be considered, but they are not as straight forward as RBFs to work with.

A trade-off between fidelity (Linear RBF) and smoothness (Cubic RBF) can be found by using the multiquadric function. The multiquadric function allows us to mimic the shape-preserving properties of the linear RBF while introducing a measure of curvature through the parameter ρ [20]. In fact, the standard multiquadric approach is modified (as described below) rather than applied strictly as defined above. As we shall see from the later sections, in certain circumstances, fitting a cubic RBF may not be appropriate. So we introduce a modified multiquadric approach and justify its use. It should be noted that while the multiquadric function allows us to prevent the reconstruction from over-smoothing at turning points, it does not interpolate the data - and so there will be some potential loss of accuracy. The extent to which this occurs depends on the trade-off required between over-smoothing and reproduction accuracy. For high-accuracy requirements, the cubic function is advised.

6.2.2.2 Expansion Using a Modified Multiquadric Approach

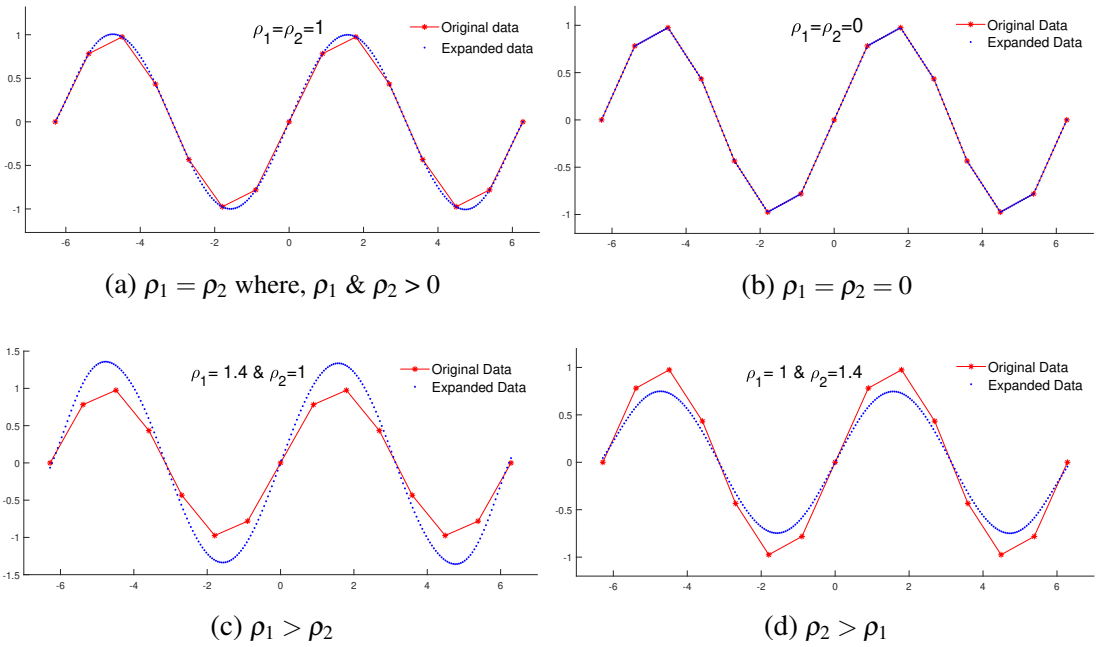
When calculating the expansion data using the multiquadric function we choose two parameters ρ_1 and ρ_2 for the basis functions $\phi(x,\rho)$. The first value is used to calculate the fitting coefficients in (6.7) by solving the system (6.8) and the second is used to calculate the expansion values y^* on the expanded data set $x_k \in [x_{min}, x_{max}]$ using

$$y^* = A^*\mathbf{c}, \qquad \text{where} \qquad A^*_{k,j} = \phi_j(x_k, \rho_2). \tag{6.12}$$

Figure (6.2) shows four different cases of the curve fitting using the multiquadric expansion approach. As an example, a sinusoidal function, sampled at 15 equally spaced points on the interval $[-2\pi, 2\pi]$ and fitted with a linear RBF are to be expanded using the multiquadric RBF. According to the expected values of ρ_1 and ρ_2, there are four fitting scenarios can be considered:

- $\rho_1 = \rho_2 > 0$: This will result in a very similar looking fit to the cubic RBF.
- $\rho_1 = \rho_2 = 0$: When the smoothness parameter gets closer to zero, the multiquadric function approaches the linear RBF.
- $\rho_1 > \rho_2$: This case will show an over-smoothing fitting.

- $\rho_2 > \rho_1$: This case results an under-smoothing fitting.



(a) $\rho_1 = \rho_2$ where, ρ_1 & $\rho_2 > 0$ (b) $\rho_1 = \rho_2 = 0$

(c) $\rho_1 > \rho_2$ (d) $\rho_2 > \rho_1$

Fig. 6.2 Expansion of modified multiquadric RBF approach using various values for the parameters ρ_1 & ρ_2.

Figure (6.2a) illustrates the effects of fitting the piece-wise linear data with a smooth multi-quadric curve when $\rho_1 = \rho_2$; here we used $\rho = 1$ for both that results in an over-smoothing fit, which is identical to the cubic RBF. Figure (6.2b) clearly shows the effects of using zero values for both parameters, which leads to the linear RBF. Figure (6.2c) shows a significant over-smoothing when ρ_1 is greater than ρ_2. However, choosing a value for ρ_1 that is greater than ρ_2 will increase the curvature and results in more over-smoothing. The over-smoothing is particularly noticeable at the turning points. While this phenomenon may not always be a problem, there are occasions when over-smoothing could cause significant problems. If the data represents values that are constrained to be strictly greater than zero, for example, then over-smoothing near boundaries could result in negative quantities being included in the expansion set. While Figure (6.2d) shows the case of under-smoothing when ρ_2 is greater than ρ_1.

We have found that replacing the original smoothing parameter with one of slightly higher value when re-sampling, results in a fitting curve that stays within the convex hull of the original data set while still allowing for a measure of smoothness. The level of smoothness required can be determined by varying the choice of values for both ρ_1 and ρ_2. The restriction here resulting in the fitting function lying within the convex hull of the data set is entirely

governed by choice of the two smoothness parameters. Smooth near-interpolation can take place without this restriction by using different parameter values.

Figure (6.3) shows how the use of different values for both ρ_1 and ρ_2 (under the condition $\rho_2 > \rho_1$) can provide different fits to the data. It can be clearly noticed that all the fits are located within the convex hull of the original data set. For instance, consider the blue dotted line which used the values of $\rho_1 = 1$ and $\rho_2 = 1.2$. This can be considered a better curve fit because it satisfies the condition of staying within the convex hull of the original data and it does not have significantly under-smoothing curvature. Further refinements of the Mulitquadric's shape parameters could see the approximation curve move towards near-interpolation whilst continuing to satisfy the desirable properties already discussed.

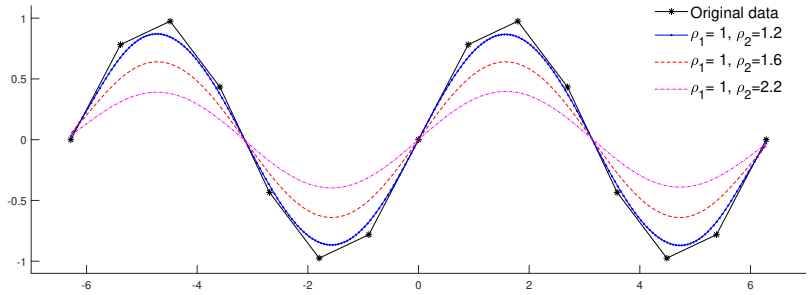


Fig. 6.3 The multiquadric RBF is used here with different values of ρ_1 and ρ_2.

6.2.3 Proposed Data Scaling Model

Combining inputs with significantly different scales can cause problems in ANFIS modelling. In some cases, real-life problems have inputs that differ by many orders of magnitude. The main problem is that each membership function corresponds to a different input. Hence, the coefficients (multipliers) need to be combined effectively to produce accurate outputs and so must be changed considerably to do a good job. It does not only introduce additional time needed for training but it also potentially introduces instabilities and poor conditioning. Though this work is not aiming to advocate any one particular method of data scaling (as different applications may require different considerations), we have chosen two specific approaches to investigate, and we present their results for comparison.

Scaling to the interval [-1 1], using a straightforward linear transformation, is known to significantly improve conditioning and stability in polynomial modelling with Chebyshev basis functions. So we use this approach here and evaluate its effectiveness for the ANFIS model. Another prevalent method of standardisation is to normalise data to have zero mean and a standard deviation of one (often called z-scores). Details showing how the data is scaled is shown below [52]:

- **Chebyshev Scaling** can be used to map the general interval $[a,b]$ to $[-1,1]$. If the data vector $\mathbf{x} \in \Re^m$ and $a = \min(\mathbf{x})$ and $b = \max(\mathbf{x})$ then the transformation function takes the form

$$\hat{x} = \frac{(x-a)+(x-b)}{b-a}, \qquad \forall x \in \mathbf{x}. \tag{6.13}$$

- **Zero-mean Scaling** takes a data vector $\mathbf{x} \in \Re^m$, calculates its mean (μ) and standard deviation (σ) and computes

$$\hat{x} = \frac{(x-\mu)}{\sigma}, \qquad \forall x \in \mathbf{x}. \tag{6.14}$$

We intend to investigate whether different scaling approaches and data expansion models yield significantly improved results.

6.3 Implementation of Model 1: Case Study

6.3.1 Problem Definition

Economically, oil production is one of the most important fields worldwide. The oil industry can be classified into two types, i.e. either producing the crude oil or the petroleum products such as fuels (gas, gasoline, jet fuel, diesel,..., etc.) and other oil derivatives.

Iraq is one of the major oil-producing countries of the world. Modelling the production of petroleum products in this country has been chosen as a test case for exploring opportunities for improving the production planning for the medium-term planning horizon. Accordingly, the discussion has been opened with the expert's team at the Iraqi Ministry of Oil in order to develop an understanding and to explore and evaluate the current status. The historical data for the production and consumption rates for four fuel products (Liquefied Petroleum Gas, Gasoline (Petrol), Kerosene, and Diesel Fuel) as well as the methods used for forecasting have been requested to be provided. This represents the first stage of this project which seeks to evaluate the accuracy and efficiency of the existing method of consumption forecasting and exploring the scientific alternative forecasting techniques that can improve the prediction accuracy. The discussion provides the following indicators:

- The consumption is uncertain and fluctuating due to seasonality and some other factors.
- The consumption data built is scarce, all available historical data are available for three years only covering the period of 2015 to 2017 at a monthly basis; that is, the total number of observations is 36 values only.

- The current prediction method depends totally on the experience of the experts using simple classical methods.
- There are set factors affecting the consumption rate. These factors can be classified as quantitative and qualitative. Some factors can be specified under fuzzy or vague concepts, such as, when the weather is "hot" the consumption increases.
- The most important product that contains consumption forecasting and production planning issues is Gasoline (Petrol).

According to the indicators mentioned above, an accurate prediction technique is needed in order to obtain an accurate prediction. This technique needs to deal with a hybrid state that contains qualitative (expert judgement and opinion) and quantitative variables at the same time. It also needs to deal with the uncertainty and vagueness of the consumption and the affecting factors. Moreover, this technique should be able to deal with the lack of information.

6.3.2 Proposed Structure: Classification of Variables

So far, we have introduced the ANFIS structure as an input-output mapping system. Therefore, using this type of system will need to classify the input and output variables. In our case, the input variables will represent the affecting factors, and the output variable will be the predicted consumption rates. According to the experts at the Iraqi Ministry of Oil, one of the main executives interest at this stage is the gasoline consumption problem. This concern arises because of the fluctuation of consumption of this product during the year's seasons. Therefore, extensive discussions with the experts have been made to identify the problem and the influencing factors that causes such fluctuations in gasoline consumption. Some assumptions have been determined, and we need to take these into consideration when we obtain the structure of the model, thus:

- The consumption is seasonal and uncertain.
- There is a monopoly of producing petrol in this country and all petrol supplied is provided only by the Ministry of Oil. Which means, there are no other competitive companies. This gives the supplier a big challenge of making sure that there is sufficient quantity to reach the consumption rates.
- The total supply capacity can be obtained from production, inventory, and imported quantities. However, it can be affected by a set of internal and external factors. The

internal factors are represented by the scarcity of crude oil, unexpected breakdown, and scheduled maintenance. The external factors, which are considered to be the hardest to control and uncertain, are things such as the unscheduled and random cuts in the main power source (Iraq has suffered from power problems for a long time). Terrorist actions are also one of the factors that affect capacity as they happen surprisingly from time to time. This can also affect the crude oil supplies, manpower availability, and even the power supply. Another external factor that can affect the capacity is unexpected bad weather, which can affect the planned periodic maintenance schedules, and so forth. Consequently, the fuzziness of the capacity variable results from a combination of all these factors.

The experts suggested three main factors that can affect consumption. These are production capacity, weather temperature, and the number of cars. These factors will be considered as the model variables that provide a cause and effect relation with gasoline consumption. The current method for predicting consumption depends totally on the experience of experts using simple, classical methods. We propose an ANFIS model of the form shown in figure 6.4, based on the Takagi-Sugeno approach [130]. This enables ANFIS to be developed based on the data (that will be collected) to give an understanding of how the consumption can be better predicted. These three factors can be considered as ANFIS inputs with the gasoline consumption representing the output. Therefore, we have:

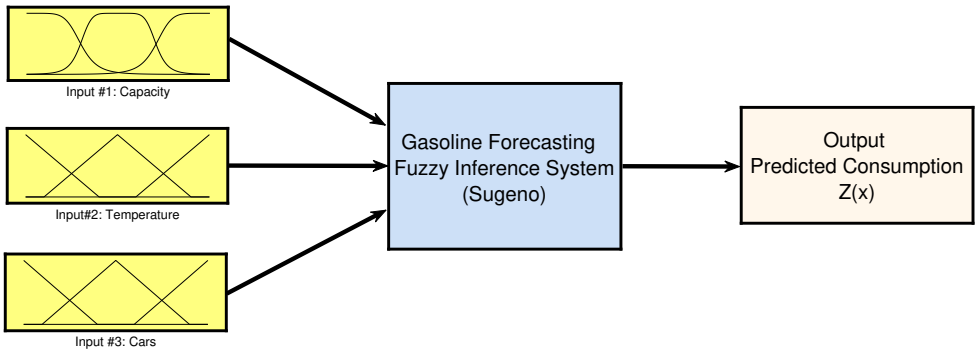


Fig. 6.4 Proposed ANFIS model - variables structure.

- Capacity (input 1): This estimate the amount of gasoline the Ministry of Oil can supply.
- Temperature (input 2): The temperature values measure the variation of the weather during the year, which is known to affect the consumption of gasoline.
- Number of cars (input 3): This represents the number of cars using gasoline and thus impacts significantly on its consumption.

- Consumption (output): This quantity represents the amount of sales expected.

According to the FIS methodology, each input can be represented by assigning various types and numbers of membership functions.

It is important to make the clear distinction that the model is not expected to use data at time t to predict data for time $t+1$. Though data is collected over time, we are not producing a time series model in the traditional sense. The factors that affect consumption in a given year are impacted by an estimate of the values in the same time period and it is this relationship that the ANFIS model seeks to determine. By training a model that is able to learn a functional relationship between its input (independent) and output (dependent) quantities, using historical data, then, with reliable estimates available, accurate predictions can be made.

6.3.3 Data Collection

As a test case, we are using data provided by the Iraqi Ministry of Oil. In order to build an accurate model, we have been supplied with three years of historical data for 2015, 2016 and 2017, which is recorded at monthly intervals giving a total of only 36 samples per variable. The historical data for the consumption rates for gasoline that have been collected showed that the range of the consumed quantities was between 450 and 670 thousands cubic meters per month. Figure (6.5) illustrates the line chart for the provided data. The x-axis represents the time periods, and the y-axis represents the consumption rate measured in cubic meters.

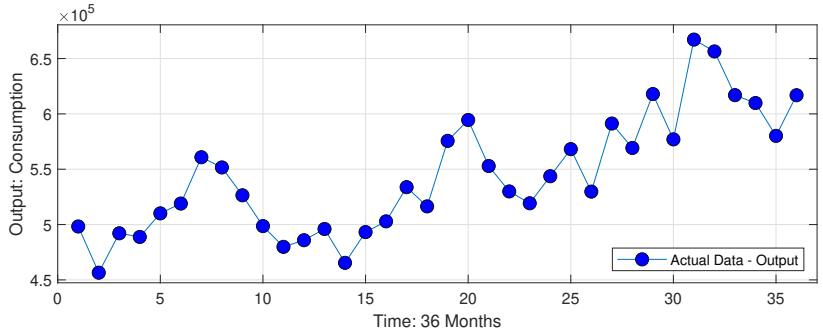


Fig. 6.5 The actual consumption for gasoline recorded at monthly intervals between 2015 and 2017.

Due to the nature of the data, we can notice that:

- The consumption is uncertain and fluctuating during each period (year) due to seasonality and other effects.

- The demand increased significantly to reach its peaks and declined again between the period of months from April to November of each year.
- The trend of demand for all 36 months shows that it rose.

Experts believe the reason behind this fluctuation is caused by a set of factors, some of which can be defined as internal; such as the production capacity, and others as external; such as the weather temperature or consumer behaviour. Therefore, the data for the affecting factors have been collected from different sources according to its nature. Weather data (temperature) have been collected from the Iraqi Meteorological Organization and Seismology. Whereas the data for the number of cars provided by the Central Statistical Organization Iraq. The capacity data has been collected from the same source at the Ministry of Oil-Iraq. Figure 6.6 shows the plots of the historical data for the same period (2015-2017).

It is not difficult to see that there are likely correlations within the data. For example, we can see that the highest consumption for each year is recorded during the hottest months of the year. Moreover, the overall trend in consumption across the three-year period is increasing in line with the rising number of cars on the roads in the country.

6.3.4 Variables Correlation Analysis

Statistically, correlation is a way of measuring the relationship between two variables. It indicates to what extent the variables are depending on each other. The range of expected correlation values is between [+1 -1]. The positive correlation of two variables means the relationship is positive, and they increase and decrease in parallel and vice versa.

Table (6.1) illustrates the correlation between the inputs-output variables of the proposed model showed in Figure (6.4). The SPSS software has been used to obtain the Pearson correlation between the consumption and the other three affecting factors (Capacity, Weather Temperature, Cars) in order to use them as ANFIS inputs. Figure (6.7) illustrates the Correlation scattering of r-value size for all the variables.

Table 6.1 Correlation between consumption and all inputs.

	Capacity	Cars	Temperature
Consumption	0.812**	0.810**	0.477*

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).

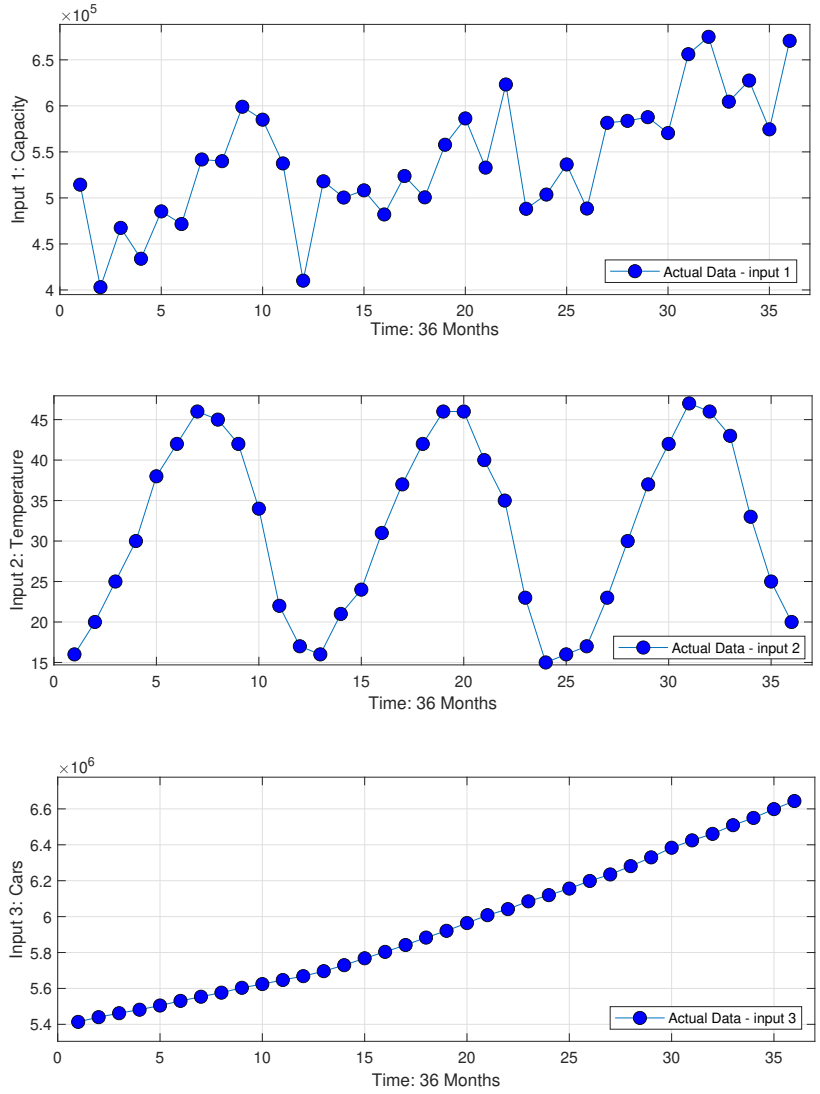


Fig. 6.6 Raw input data determined by experts to have the greatest influence over consumption prediction. The data has been recorded at monthly intervals between 2015 and 2017.

Notably, the correlation coefficient (r-value) between consumption and both the capacity and numbers of cars are significant with r equal to (0.812, 0.810), respectively, at the level of (0.01). This means the relationship between the consumption and these two variables is positive and strong. In other words, this positive relation means whenever the value of the capacity and the number of cars increases, the consumption will increase. Consumption is correlated positively with weather temperature but not strong with r = 0.477 with a significance level of (0.05).

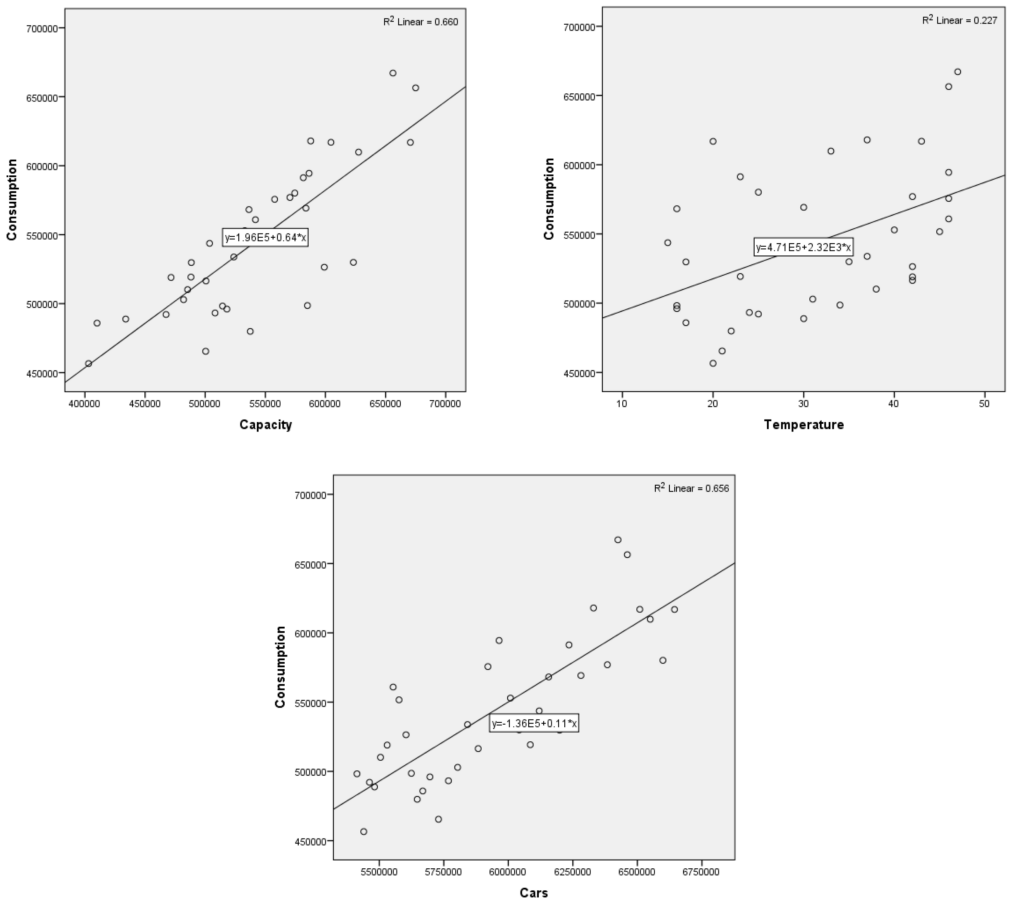


Fig. 6.7 Correlation scattering of r-value size.

6.3.5 RBF Interpolation (Data Expansion) Performance

As mentioned previously, the collected raw data set is consists of only 36 values per variable. Each value represents one-month data in a particular set. This number of data is scarce compared to other similar types of problems. We know that ANFIS is a hybrid system which utilizes both the neural networks and fuzzy approach. The primary key consideration for the neural networks is the quality and quantity of the data. There will need to be enough data to ensure the convergence of the network from the given samples can be satisfied. Therefore, if we use the thirty-six pairs of data sets (months), we gain this kind of accuracy; while if we re-sample it into a higher number of data sets by employing mathematical tools, then we may improve our results. Logically, we are dealing with monthly data as a base framework, so we can either re-sample it into weekly or daily amounts as a logical smaller time framework (period). Therefore, we are looking at re-sampling every year into either 52 weeks or 365 days. By using the forms (6.9) to (6.11), the function $F(x)$ is constructed and consistently

re-sampled over a refined domain to expand the data sets for ANFIS. Each data set of 36 monthly values is expanded into two further sets of 156 data samples (representing weekly data) and 1095 data samples (representing daily data). There are three methods that will be used as an expanding tool. They are the Linear RBF, Cubic RBF, and Modified Multiquadric RBF as already defined.

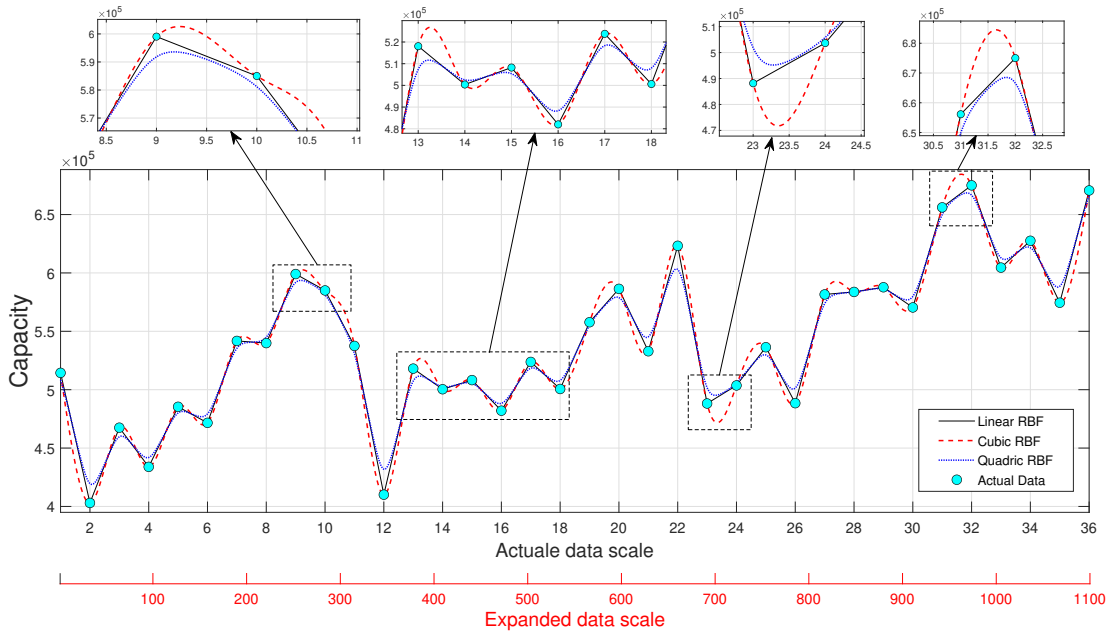
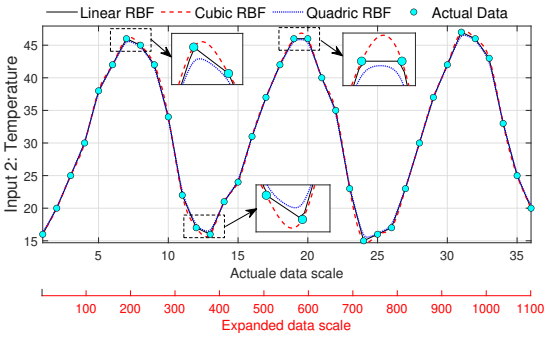
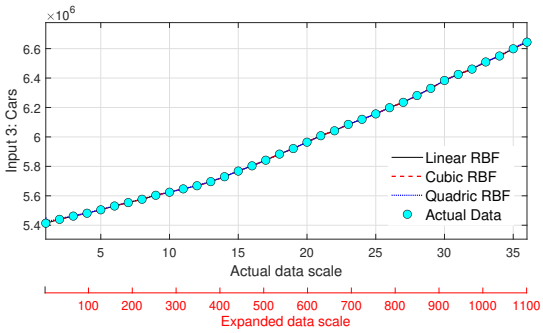


Fig. 6.8 The three radial basis functions are shown here modelling the original data labelled Input 1 - Capacity. Additional highlighting shows the behaviour of the fitting functions at some of the turning points.

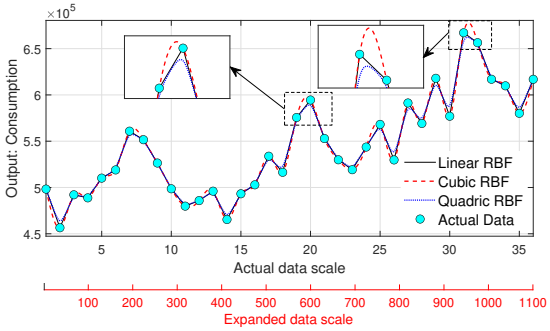
Figure (6.8) shows how the different RBF fitting functions for the data of the first input (capacity) fair at the turning points, clearly highlighting where over-smoothing needs to be considered. The solid black line represents the Linear RBF expanded data after applying Equation (6.9) to the raw data of input 1. The red dash line demonstrates the Cubic RBF interpolated data using Equation (6.10). The blue dotted line illustrates expanded data resulting from applying the Modified Multiquadric RBF - Equation (6.11). We used $\rho_1 = 0.05$ and $\rho_2 = 0.2$ as the Multiquadric smoothing parameters, chosen based on our interpretation of the underlying curvature of the data. These two specific values produced a fitting curve that stays within the convex hull of the original data set while still allowing for a measure of smoothness. The required level of smoothness can be determined by choosing different values for both ρ_1 and ρ_2.



(a) Input 2 (Temperature) RBF interpolation.



(b) Input 3 (Cars) RBF interpolation.



(c) Output (Consumption) RBF interpolation.

Fig. 6.9 The three RBFs are shown here re-sampling the original data sets of the input 2, input 3 and the output variables.

The same fitting functions have been applied into the second and third inputs, i.e. temperature and cars, respectively, as well as the output (consumption) in order to provide the system with the same number of data pairs. Figures (6.9a to 6.9c) shows the resulted expanded data.

6.3.6 Model Solving Procedures

6.3.6.1 Data Splitting

In order to train and validate the quality of the ANFIS model, it is necessary to partition the data into two sets, one for training and one for validation. The proposed methodology here is to split the data into two sets. The first two years, 2015 and 2016, will constitute training data and will be either the original 24 months of data or expanded data (104 weeks or 730 days). The third-year, 2017, will be used for validation and will be either the original 12 months data or expanded data (i.e. 52 weeks or 365 days). This will provide a corresponding ratio of 67% of the data set for training, and 33% for testing.

6.3.6.2 Validation Models

In ANFIS models, both topology and membership function types determine the total number of parameters that need optimising. Hence, if a model is chosen such that the total number of parameters exceeds the number of data, then it is expected to perform poorly. As a test case, three ANFIS models have been chosen to validate the proposed model (see Section 6.2.1) and demonstrate the effect of the model's complexity on the total number of parameters that must be resolved to produce a good predictor. The structure of the first validation model (ANFIS 1) contains two MFs for each input, i.e. two Triangular, two Difference Sigmoidal, and two Triangular, respectively. The second validation model (ANFIS 2) contains four Triangular, three Difference Sigmoidal and two Gaussian MFs for inputs (1 to 3), respectively. While the third validation model (ANFIS 3) is to have four Gaussian MFs for each input. All these validation models will be discussed in more details in the next sections.

Table (6.2) demonstrates the overall information of the three models. By reviewing the information listed in this table, it can be noticed that the first validation model (ANFIS 1) has the lowest complexity with total parameters of (52) and eight fuzzy rules. While the complexity of the second validation model (ANFIS 2) is fairly complex compared to the first and third models; with total parameters of (124) and (24) fuzzy rules. However, (ANFIS 3) can be considered as the most complicated validation model among the other two by containing (280) parameters and (64) fuzzy rules.

6.3.6.3 Evaluation Methods

Different statistical methods can be used in order to assess the performance of the models. However, in this work, the coefficient of determination and the normalised root mean square error measure has been used to evaluate the performance of the proposed model. Let us define the output of the FIS as $Z(\mathbf{x}_i)$, where $\mathbf{x}_i \in \Re^3$ contains the three inputs at time i and let t_i represent the corresponding target or desired output (consumption). To ensure a consistent interpretation of the accuracy of the ANFIS models, we use two metrics. The first one is the *coefficient of determination* (i.e., R^2) which can be defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{m}(t_i - Z(\mathbf{x}_i))^2}{\sum_{i=1}^{m}(t_i - \bar{t})^2}.$$

Where m is the number of validation samples and $\bar{t}$ is the average of the target values. The R^2 value assesses how well the ANFIS model explains and predicts the consumption. The

Table 6.2 Structure of the three validation models - case study 1.

Model	Input	MF Type	MF Num	ANFIS Info	
ANFIS 1	1 2 3	Triangular D.Sigmoidal Triangular	2 2 2	Number of Node	34
				Linear Parameter	32
				Non-linear parameter	20
				Total Parameter	52
				Fuzzy rules	8
ANFIS 2	1 2 3	Triangular D.Sigmoidal Gaussian	4 3 2	Number of Node	72
				Linear Parameter	96
				Non-linear parameter	28
				Total Parameter	124
				Fuzzy rules	24
ANFIS 3	1 2 3	Gaussian Gaussian Gaussian	4 4 4	Number of Node	158
				Linear Parameter	256
				Non-linear parameter	24
				Total Parameter	280
				Fuzzy rules	64

second one is the *normalised root mean square error* measure, defined as:

$$\text{NRMSE} = \frac{\text{RMSE}}{t_{max} - t_{min}},$$

where

$$\text{RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(t_i - Z(\mathbf{x}_i)\right)^2}.$$

Dividing the RMSE by the range over which the predictions are required allows us to compare accuracy across the different scales of prediction approaches. Although the training data and test data are mapped to similar intervals via the two scaling approaches they are not identical.

6.3.7 Empirical Results

In this section, we present the results of the investigations carried out on the gasoline consumption prediction problem. By applying our proposed model (section 6.2.1) into the three validation models (i.e., ANFIS 1, 2 and 3) listed in Table (6.2), we get the results shown in Tables (6.3, 6.5 and 6.7) respectively. Each validation model has eight experiments. These

experiments include all three proposed expansion methods (i.e., Linear RBF, Cubic RBF, and Modified Mutiquadric RBF) combined with two scaling methods (i.e. Chebyshev and Zero-mean).

The results of the eight experiments for each ANFIS model are separated into two parts according to the scaling method. The first four experiments represent the results of using Chebyshev scaling. While the second four expressing the results of utilizing the Zero-mean scaling. In order to better evaluate the different approaches, model performance is classified into two types, i.e. **local performance** and **global performance**. In the local performance, we compare and contrast the results of using non-expanded (original) and expanded data within a specific scaling method. However, we highlight and explain the improvements and differences in performance for each part internally, whereas the global performance explains the overall results for all eight experiments of each model.

A total of 24 experiments have been carried out using the three validation ANFIS models listed in Table (6.2). All experiments were solved by using Matlab's FIS modelling software (i.e., *genfis1, anfis* & *evalfis*). However, we chose to use the Matlab functions directly with the grid partitioning method rather than via the available toolbox. All of the networks were trained for 100 epochs. We could not find an acceptable model for any combination of choices without first scaling the data. Therefore, we do not present results for non-scaled data. All of the 24 experiments were using the data for three years spanning the period 2015 to 2017. The 2015 and 2016 data was used to train the models, and the 2017 data was used to evaluate the prediction accuracy. In the Tables (6.3 to 6.5), for each combination of choices (shown row-wise) the data can be either monthly, weekly or daily. The RBF expansion method can be none (—), Linear (Lin), Cubic (Cub) or Multiquadric (Mul). The scaling method is either Chebyshev (Ch) or Z-scores (Z-s).

6.3.7.1 First Validation Model (ANFIS 1)

Referring to the ANFIS validation models explained in Table (6.2), the structure of the first model (ANFIS 1) contained two MFs for each input. However, in this section, we will discuss this model in more detail. The first input (capacity) is represented by triangular MFs using two linguistic terms, namely Low Capacity (LC) and High Capacity (HC). The second input (temperature) is expressed with two Difference Sigmoidal MFs named as Cold Weather (CW) and Hot Weather (HW). Whereas the third input (number of cars) characterized by two Triangular MFs titled as Small Number (SN) and Large Number (LN), respectively. This architecture produced (20) non-linear (premise) parameters and (32) linear (consequence) parameters. Figure (6.10) shows the network structure of ANFIS 1 (MATLAB output).

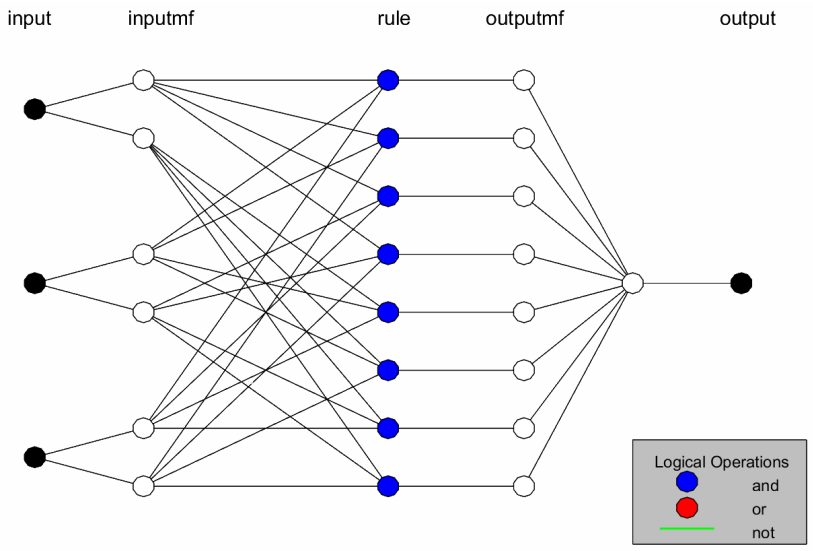


Fig. 6.10 ANFIS 1 structure.

Depending on the structure showed in Figure (6.10), the complexity of this model can be classified as low. Applying Equations (6.1 to 6.6) into ANFIS 1 showed that the total number of parameters is (52), which is exceeding the number of original data pairs (36). In this case, we need to expand the original data into a proper order of magnitude. This will overcome the problem of having the number of data samples that is less than the number of parameters. In other words, the data needs to be expanded into a reasonable level that can provide the model with enough number of data pairs in order to keep the balance between the model complexity from one side, and accuracy and efficiency from the other side. Therefore, finding the best number of expanded data is important in order to gain better performance of the model.

Logically, as a time framework, the lower levels of time period from the original monthly data will be either weeks or days. Therefore, the monthly data are to be replaced with (52×3) data points provided (156) weeks data samples using the forms (6.9 to 6.11). This will increases the number of input data up to an appropriate level which is compatible with the complexity of this model. Consequently, it can help to overcome the over-fitting problem for this model and gain a good predictor.

Table (6.3) illustrates eight experiments resulting from solving ANFIS 1 using all possible combinations of expansion and scaling methods proposed in model 1 (Fig. 6.1).

By reviewing the results listed in Table (6.3) we can find that:

1. ANFIS 1 Local performance (Chebyshev scaling), experiments (1-4):

Table 6.3 Results of the first validation model (ANFIS 1).

Experiment	Data	Samples	Expansion	Scaling	NRMSE	R^2	Performance	
							Local	Global
1	Monthly	36	——	Ch	0.4406	0.5427	——	——
2	Weekly	156	Lin RBF	Ch	0.1039	0.8875	↗ 34%	↗ 34%
3	Weekly	156	Cub RBF	Ch	0.1299	0.8391	↗ 30%	↗ 30%
4	Weekly	156	Mul RBF	Ch	0.1047	0.9196	↗ 38%	↗ 38%
5	Monthly	36	——	Z-s	0.7373	0.7476	——	↗ 20%
6	Weekly	156	Lin RBF	Z-s	0.0829	0.9097	↗ 16%	↗ 37%
7	Weekly	156	Cub RBF	Z-s	0.0935	0.9055	↗ 16%	↗ 36%
8	**Weekly**	**156**	**Mul RBF**	**Z-s**	**0.0698**	**0.9529**	↗ **21%**	↗ **41%**

- All expanded data (experiments 2-4) performed significantly better than non-expanded data (experiment 1). Nearly 30% to 38% improvement in prediction accuracy compared to experiment one.
- The best performance resulted from using the Multiquadric RBF approach (where $\rho_1 = 0.01$ and $\rho_2 = 0.2$) which provides up to 4% improvement compared with Linear RBF and 8% compared to Cubic RBF.

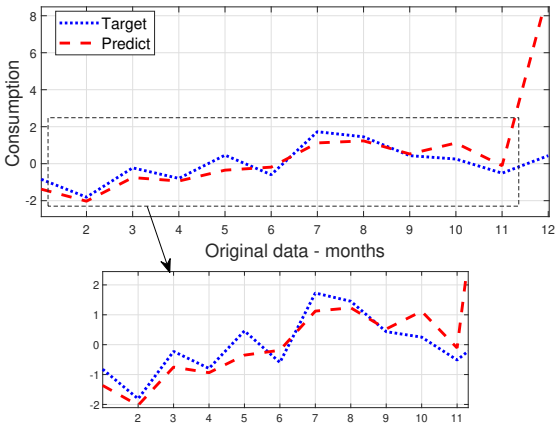
2. ANFIS 1 Local performance (Z-score), experiments (5-8):
   - Again, we can see that all expanded data (experiments 6-8) perform better than non-expanded data. Approximately 16% to 21% improvement in prediction accuracy compared to experiment five.
   - The best performance resulted from using the Multiquadric RBF approach, which gives an improvement of up to 5% compared to Linear and Cubic RBF.
3. ANFIS 1 Global performance, where all experiments (2-8) will be compared with experiment one as a base because it shows the lowest performance for this model.
   - It can be noticed that the Z-scores scaling method outperformed (by nearly 20%) the Chebyshev scaling when using non-expanded data.
   - The results of solving this model showed that in all experiments, looking at both NRMSE and R^2, the expanded data performs significantly better than non-

expanded data. It is interesting to see that if no underlying knowledge of the shape of the data is known, then even a simple linear fit to the data can outperform a non-expanded model.

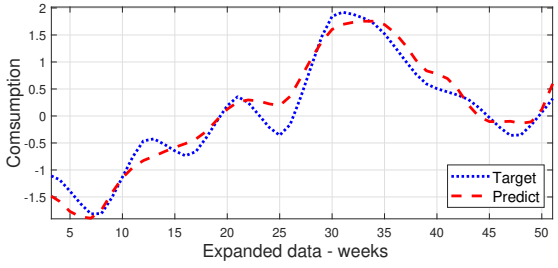
- We can also see that using the Z-scores scaling approach gives an improvement of between 3% and 6% depending on the expansion methods used when compared to Chebyshev scaling, respectively.
- When care is taken, expanding the data using a more sophisticated model, such as the Multiquadric RBF (where $\rho_1 = 0.01$ and $\rho_2 = 0.2$) the model can perform even better with an order of magnitude difference in the error (seen in experiment 4 and 8). Approximately 38% and 41% improvement in prediction accuracy for the two scaling approaches can be seen, respectively.
- The highest performance resulted from experiment eight with the lowest prediction error of (0.0698) and the highest R^2 (95.29%).

Depending on the results listed in Table (6.3) and its discussion above, we introduce the results of experiment eight (i.e. optimal results) in more detail below.

Figure (6.11) shows the predicted consumption values for 2017 using the non-expanded data - experiment 5 (fig. 6.11a) and expanded data - experiment 8 (fig. 6.11b). The prediction graphs clearly show that by re-sampling the original data into optimised expanded data, we can obtain a substantial improvement. This is significant considering the impact this model has on potential manufacturing decisions.

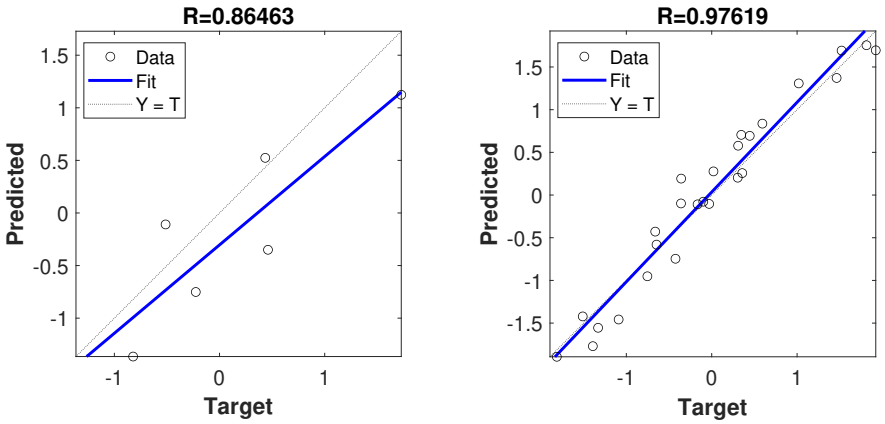


(a) Before expansion.



(b) After expansion.

Fig. 6.11 Targeted and predicted gasoline consumption for validation data of ANFIS 1: (a) Before expansion (experiment 5), (b) After expansion (experiment 8).



(a) Before expansion. (b) After expansion.

Fig. 6.12 Regression scatter plot of the targeted and predicted gasoline consumption for validation data of ANFIS 1: (a) Before expansion (experiment 5), (b) After expansion (experiment 8).

Figure 6.12 confirms how well the model fits the data. The regression scatter plots demonstrate how well the model captures the variation in the underlying data. The plot corresponding to experiment eight (6.12b) clearly shows how 95% of the variability (equal to $R = 0.97619$) is explained by the model constructed using the expanded data. The corresponding fit for experiment five is shown in figure (6.12a).

The initial and final (optimal) values of the premise parameters for experiment eight are listed in detail in Table (6.4). Each input has corresponding MFs assigned to it. Inputs 1 and 3 used two triangular MFs; each MF has three parameters (a, b, c) (see Equation 3.24). Input 2 used two Difference Sigmoidal MFs; each MF has four parameters (two pairs of $\{a_1, c_1, a_2, c_2\}$) (see Equation 3.32).

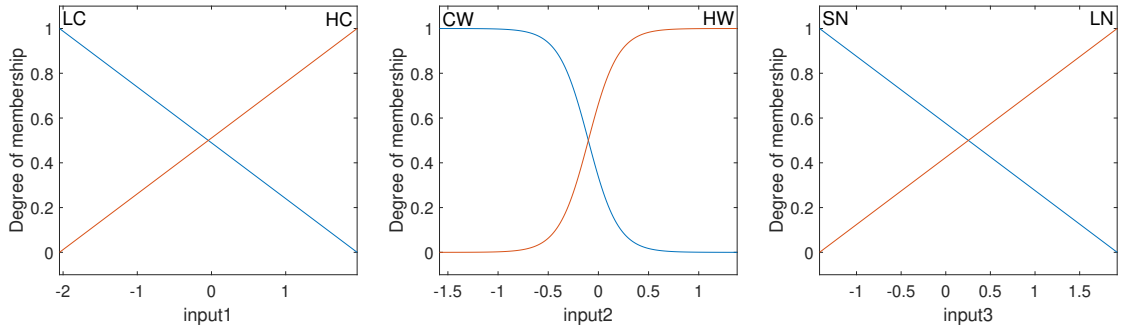
The initial default values of the premise parameters are to be placed automatically by ANFIS. Figure (6.13) shows the plot of initial MFs of these three inputs.

After training with hybrid learning algorithms, the optimal values of the premise parameters showed in Table (6.4) are described in Figure (6.14). By reviewing the shapes of the trained membership functions of Capacity (Fig. 6.14a), they are not showing a noticeable change after training. Both membership functions, low capacity (LC) and high capacity (HC) have slight changes that can barely be noticed.

In Figure (6.14b), after training, the membership function of temperature shows evident changes. The crossover point between the cold weather (CW) and the hot weather (HW) has moved upward. For the (CW), the slope has slightly decreased from $a_{2_{initial}} = 6.742$ to $a_{2_{optimal}} = 6.725$. While the crossover point has moved towards the right, this resulted from the increase of the c_2 value, i.e., from $c_{2_{initial}} = -0.1$ to $c_{2_{optimal}} = 0.0417$ which changed the

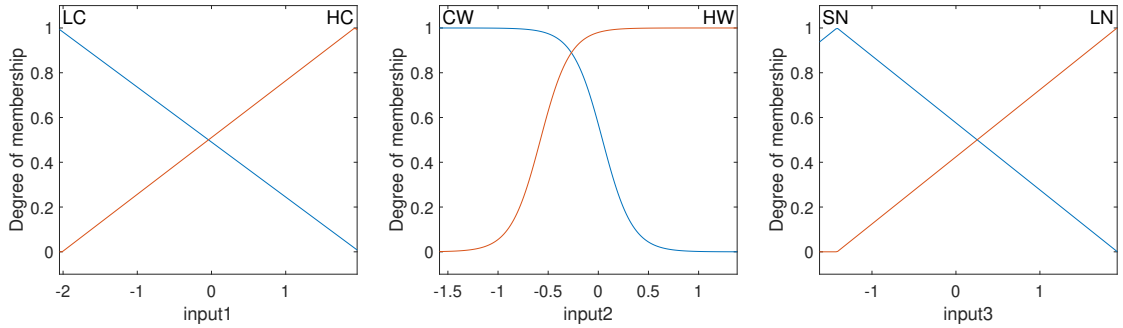
Table 6.4 Premise parameters - experiment 8 - ANFIS 1.

Input	MF Type	MFs	Initial Parameters				Optimal Parameters			
			a	b	c		a	b	c	
Capacity (Input 1)	Triangular	(LC)	-6.06	-2.048	1.963		-6.06	-2.075	1.996	
		(HC)	-2.048	1.963	5.974		-2.012	1.93	5.974	
			a_1	c_1	a_2	c_2	a_1	c_1	a_2	c_2
Temp. (Input 2)	Difference Sigmoidal	(CW)	6.742	-3.067	6.742	-0.1	6.742	-3.067	6.725	0.0417
		(HW)	6.742	-0.1	6.742	2.866	6.737	-0.578	6.742	2.866
			a	b	c		a	b	c	
Cars (Input 3)	Triangular	(SN)	-4.749	-1.414	1.92		-4.749	-1.416	1.923	
		(LN)	-1.414	1.92	5.255		-1.415	1.919	5.255	



(a) Capacity. (b) Temperature. (c) Cars.

Fig. 6.13 Initial membership functions of ANFIS 1 inputs.



(a) Capacity. (b) Temperature. (c) Cars.

Fig. 6.14 Final (trained) membership functions of ANFIS 1, experiment 8.

corresponding x-axis coordinate value. For the (HW), the slope also slightly decreased from $a_{1_{initial}} = 6.742$ to $a_{1_{optimal}} = 6.737$. While the crossover point has moved towards the left, this

resulted from the increase of c_1 value from $c_{1_{initial}} = -0.1$ to $c_{1_{optimal}} = 0.578$ which changed the corresponding x-axis coordinate value. This increased the intersection area between the two membership functions (CW and HW), which can provide better performance and efficiency for the model when optimizing the firing strength of the rules for the second input.

In Figure (6.14c), evident changes can be noticed for the trained MFs of input 3 (number of cars). For the triangular membership function of the small number (SN), the bottom left vertex "parameter a" maintained the same initial value. While the peak "parameter b" and the bottom right vertex "parameter c" has been increased. In another side, the bottom right vertex of the large number (LN) membership function kept fixed. Slight changes in the peak and bottom left vertex values have been made after training. These optimal parameters for both membership functions (SN & LN) have extended the support areas toward the right. This provided a larger intersection area from the left side for both MFs, which can be reflected positively in the fuzzification process.

The above analysis indicates that when using models with low complexity, precisely like ANFIS 1. The expanded data performed significantly better than the original data. The fine-tuning of the MFs resulted from optimizing the premise parameters. It showed that the second "temperature" and the third "number of cars" inputs have a significant impact on the gasoline consumption problem while the first input "capacity" has less impact than the other two inputs.

6.3.7.2 Second Validation Model (ANFIS 2)

The structure of the second validation model (ANFIS 2) contains four Triangular, three Difference Sigmoidal and two Gaussian MFs for inputs (1 to 3), respectively. The MFs of the first input named as Very Low Capacity (VLC), Low Capacity (LC), High Capacity (HC) and Very High Capacity (VHC). The MFs of the second input termed as Cold Weather (CW), Natural Weather (NW) and Hot Weather (HW). Finally, the MFs of the third input titled Small Number (SN) and Large Number (LN). Figure (6.15) shows the network structure of ANFIS 2 (MATLAB output).

It can be noticed that this model is more complicated than the first one in which the number of the total parameters has increased up to (124), and the topology showed (24) fuzzy rules. Thus, the original data are to be re-sampled into a reasonable level that can keep the model solving the over-fitting problem. Here, the weekly (156) expanded data can be tested in order to find the best fitting model that can overcome the over-fitting problem. The results of solving this model using all proposed scaling and expanding approaches combination are shown in Table (6.5). Here we used ($\rho_1 = 0.02$ and $\rho_2 = 0.2$) as the Multiquadric RBF approximation parameters.

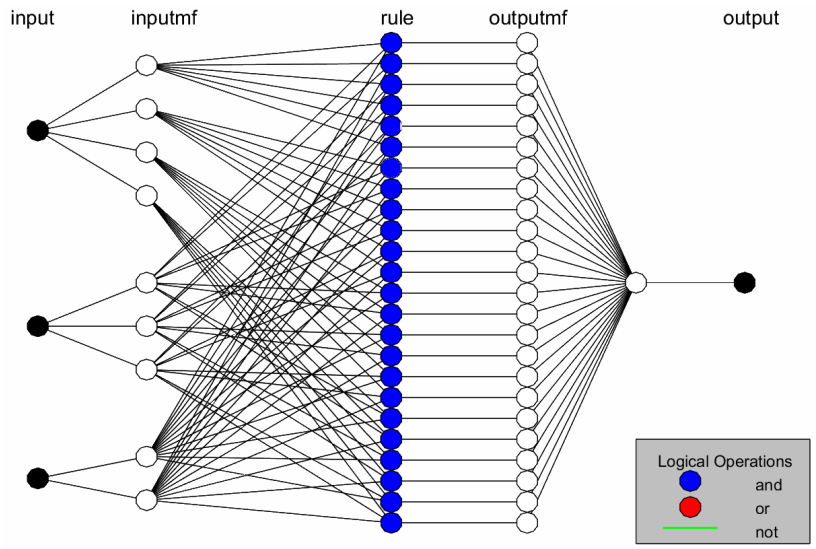


Fig. 6.15 ANFIS 2 structure.

Table 6.5 Results of the second validation model (ANFIS 2).

Experiment	Data	Samples	Expansion	Scaling	NRMSE	R^2	Performance	
							Local	Global
9	Monthly	36	——	Ch	0.1852	0.5805	——	0%
10	Weekly	156	Lin RBF	Ch	0.1100	0.8433	↗ 26%	↗ 26%
11	Weekly	156	Cub RBF	Ch	0.1039	0.8424	↗ 26%	↗ 26%
12	**Weekly**	**156**	**Mul RBF**	**Ch**	**0.0633**	**0.9543**	↗ **37%**	↗ **37%**
13	Monthly	36	——	Z-s	0.1609	0.6937	——	↗ 11%
14	Weekly	156	Lin RBF	Z-s	0.1327	0.8900	↗ 20%	↗ 31%
15	Weekly	156	Cub RBF	Z-s	0.1813	0.5856	↘ -11%	↗ 1%
16	Weekly	156	Mul RBF	Z-s	0.0642	0.9495	↗ 26%	↗ 37%

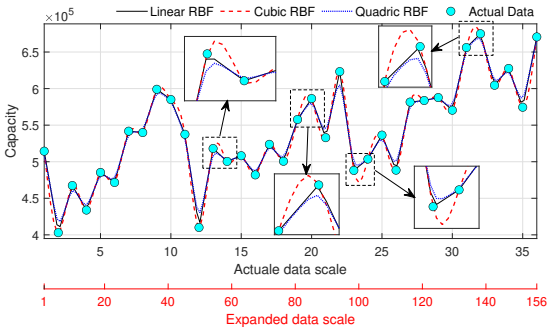
By reviewing the results listed in Table (6.5) we can find that:

1. ANFIS 2 Local performance (Chebyshev scaling), experiments (9-12):
    - The results showed that experiments (10-12) performed significantly better than experiment (9), Nearly 26% to 37% improvement in prediction accuracy compared to experiment 9. In other words, the proposed expansion methods provide better results than non-expanded by overcoming the over-fitting problem.

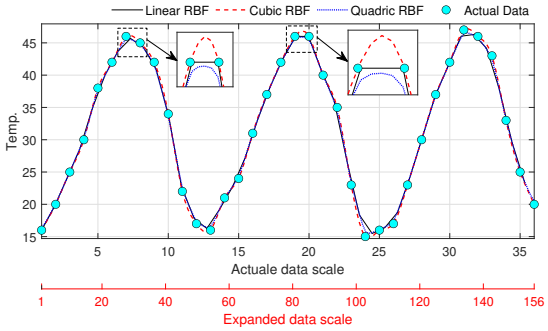
- The modified Multiquadric RBF approach (where $\rho_1 = 0.02$ and $\rho_2 = 0.2$) showed the best performance among the other expansion approaches. It provides significant improvement up to 11% compared with Linear RBF and Cubic RBF.

2. ANFIS 2 Local performance (Z-score), experiments (13-16):

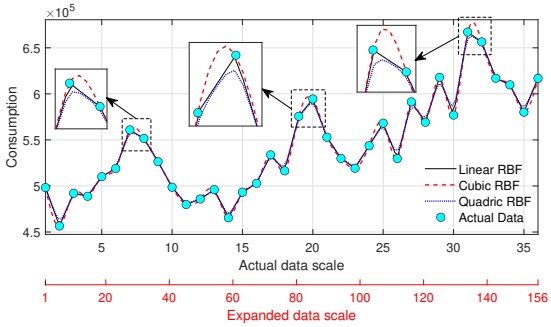
   - The results of using Multiquadric and Linear RBF expansion (experiments 14 & 16) showed improvements of up to 26% & 20% respectively, compared to experiment (13).
   - Using the Cubic RBF expansion (experiment 15) indicates a defect in performance compared to the non expanded data (experiment 9). The results showed that the NRMSE increased up to nearly 2%, offset by a fall in R^2 value down to -11%. That is because the Cubic RBF is showing significant over-smoothing near boundaries. Figure (6.16) shows the plots of RBF approximation for this experiment. It can be noticed that there is a significant over-smoothing in some turning points which caused this defect.



(a) Input 1 (Capacity) RBF interpolation.



(b) Input 2 (Temperature) RBF interpolation.



(c) Output (Consumption) RBF interpolation.

Fig. 6.16 The RBFs of experiment 15 (ANFIS 2) with additional highlights in some turning points.

This defect occurs in this particular case of experiment (15) when the Z-score scaling is combined with the Cubic RBF (specifically for the weekly expansion of 156 samples). However, it can be considered as a case of poor prediction for the structure of ANFIS 2 model.

- The best performance resulted from using the Multiquadric RBF expansion and Z-score scaling approaches, which gives up to 26% improvement compared to non-expanded data (experiment 13) and 6% compared to the Linear RBF expansion.

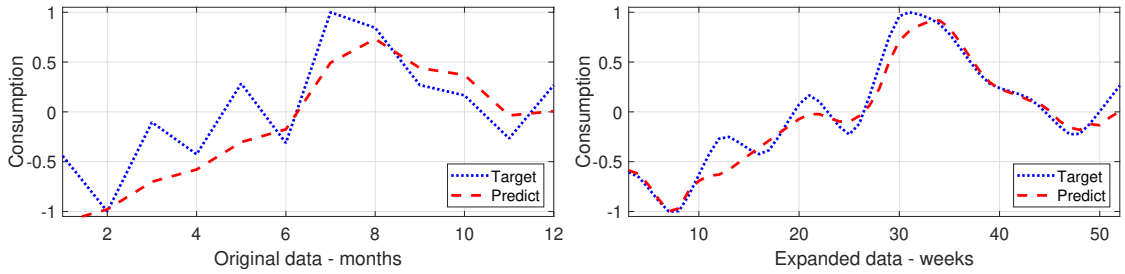
3. ANFIS 2 Global performance: Here we are analysing the results of experiments (10 to 16) compared to experiment (9) as a base because it shows the lowest (best) performance for this model.

    - Here we can see that using the Z-scores scaling method performed up to 11% better than the Chebyshev scaling when using non-expanded (original) data.
    - Looking at the two scaling methods, it can be noticed that the performance measures of each pair of any particular expansion method showed disparity behaviour compared to other expansion methods. For the Linear RBF expansion (experiments 10 & 14) , using Z-score scaling showed an improvement of 5% compared to Chebyshev scaling. Whereas, the use of the Cubic RBF with Chebyshev scaling (experiment 11) outperformed the Z-score results (experiment 15) by nearly 25%. The use of Multiquadric RBF expansion with both scaling methods (experiments 12 & 16) provides a similar performance value of 37%.
    - Although the Cubic RBF showed a defect when combined with Z-score (experiment 15), it did however, have an improvement of 1% in overall performance compared to experiment nine. Therefore, we can conclude that using expanded data showed improvements in all experiments.
    - The best performance resulted in experiment twelve, represented by the lowest prediction error (0.0633) and the highest R^2 (95.43%).

The discussion above indicates that experiment twelve has the best performance among all results listed in Table (6.5). We introduce the results of this particular experiment in more detail below.

Figure (6.17) illustrates the graphs of convergence between the target and predict values of the validation data for the gasoline consumption. Figure (6.17a) represents the results of experiment (9) using the original data. While Figure (6.17b) shows the results of experiment

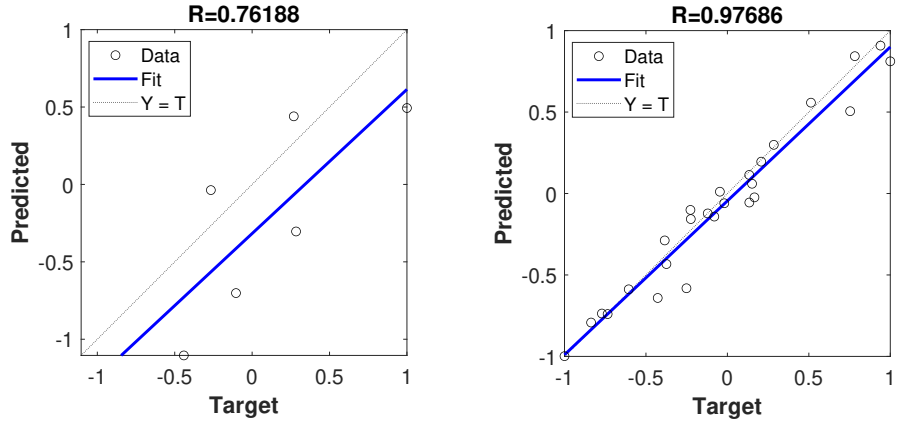
(12) using the expanded data. An evident improvement can be seen by re-sampling the original data into optimised expanded data.



(a) Before expansion. (b) After expansion.

Fig. 6.17 Targeted and predicted gasoline consumption for validation data of ANFIS 2, Table (6.5): (a) experiment 9 (b) experiment 12.

Figure (6.18) emphasises the ability of the model to fit the data. The regression scatter plots shows how well the model captures the variation of data. By reviewing Figure (6.18b) which represents experiment twelve, we can notice how nearly 95% of the variability (equal to $R = 0.97686$) is explained by the model constructed using the expanded data; compared to ($R = 0.76188$) when using original data at experiment nine in Figure (6.18a).



(a) Before expansion. (b) After expansion.

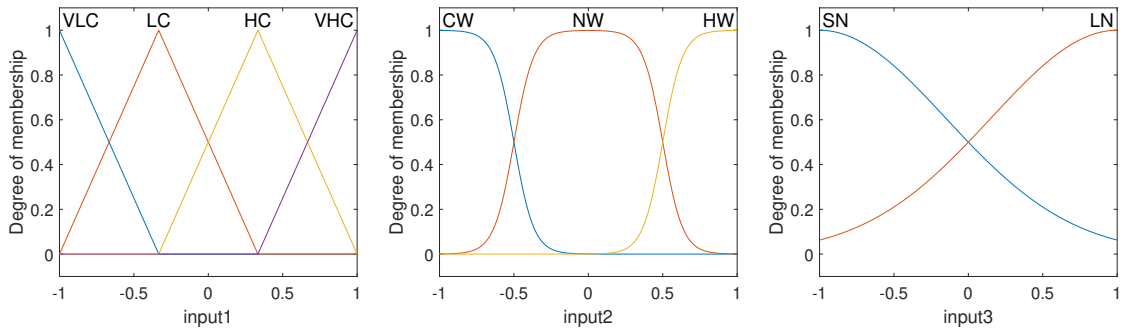
Fig. 6.18 Regression scatter plot of the targeted and predicted gasoline consumption for validation data of ANFIS 2: (a) experiment 9 (b) experiment 12.

Table (6.6) illustrate the initial and optimal premise parameters for each membership function of all inputs of experiment twelve. The first input uses four Triangular MFs; each MF has three parameters (a, b, c). The second input uses three Difference Sigmoidal MFs; each MF has four parameters (two pairs of $\{a_1, c_1, a_2, c_2\}$). Whereas the third input utilizes two Gaussian MFs; each MF has two parameters of (σ& c) (see Equation 3.28).

Table 6.6 Premise parameters - experiment 12 - ANFIS 2.

Input	MF Type	MFs	Initial Parameters				Optimal Parameters			
			a	b	c		a	b	c	
Capacity (Input 1)	Triangular	(VLC)	-1.667	-1	-0.3333		-1.667	-0.9969	-0.334	
		(LC)	-1	-0.3333	0.3333		-1.002	-0.3259	0.346	
		(HC)	-0.3333	0.3333	1		-0.3293	0.3364	1.002	
		(VHC)	0.3333	1	1.667		0.3098	0.9976	1.667	
			a_1	c_1	a_2	c_2	a_1	c_1	a_2	c_2
Temp. (Input 2)	Difference Sigmoidal	(CW)	15	-1.5	15	-0.5	15	-1.5	15	-0.6087
		(NW)	15	-0.5	15	0.5	15	-0.5509	15	0.5156
		(HW)	15	0.5	15	1.5	15	0.5386	15	1.5
			σ	c			σ	c		
Cars (Input 3)	Gaussian	(SN)	0.8493	-1			0.8387	-1.006		
		(LN)	0.8493	1			0.8391	1.003		

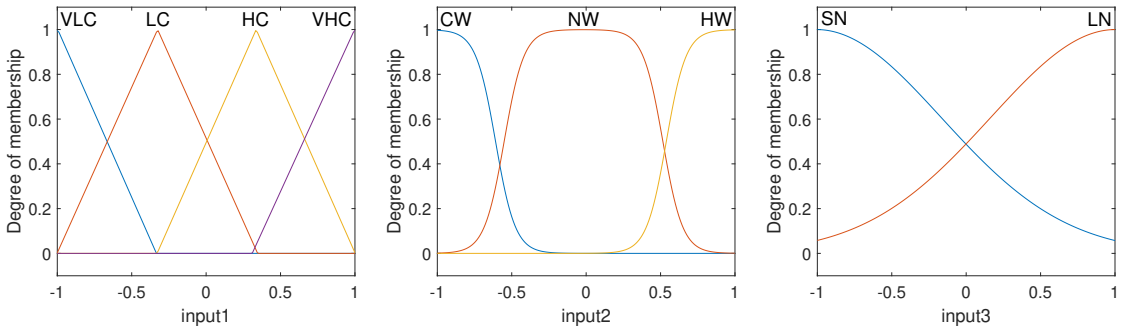
Figures (6.19 & 6.20) show the plots of the initial and optimal MFs of the three inputs of ANFIS 2 listed in Table (6.6), respectively.



(a) Capacity. (b) Temperature. (c) Cars.

Fig. 6.19 Initial membership functions of ANFIS 2 inputs.

Despite the fact that experiment twelve provides a significant improvement in model performance, the shapes showed in Figures (6.19 & 6.20) indicate that no significant changes have occurred on the trained MFs when compared to the initial ones. In other words, there are no major differences between the initial and optimal values of the premise parameters. This means, when solving ANFIS 2 using the scaling and expanding combination specified in experiment (12), it provides a useful initial topology design that is not far from the optimal. This has sped up the ANFIS system computations and uses only 15 epochs in the training process in order to find the optimal model.



(a) Capacity. (b) Temperature. (c) Cars.

Fig. 6.20 Final (trained) membership functions of ANFIS 2, experiment 12.

For instance, looking at final MFs of the second input (temperature) (Fig. 6.20b), at first glance, it looks similar to the initial MFs (Fig. 6.19b). Nevertheless, even slight changes in the optimal parameter values of (c_2) at the cold weather (CW) MF, (c_1&c_2) at the natural weather (NW) MF and (c_1) at the hot weather (HW) MF; have made some influence on the fuzzification area. These changes have moved (CW) towards the left and (HW) towards the right, slightly. Also, it extends the (NW) area by moving its left and right sides outward. As a result of these movements, the crossover points have moved downward, and the overlap between all MFs have been decreased. However, this provides a wider area for (NW), which gives more flexibility when composing the fuzzification process with other MFs.

Overall, all the inputs (capacity, temperature and cars) nearly have the same impact in the model.

6.3.7.3 Third Validation Model (ANFIS 3)

The third validation model (ANFIS 3) is to have four Gaussian MFs for each input. This can be labelled as Very Low Capacity (VLC), Low Capacity (LC), High Capacity (HC) and Very High Capacity (VHC) for the first input. For input two, we used Cold Weather (CW), Natural Weather (NW), Hot Weather (HW) and Very Hot Weather (VHW). While input three can have Small Number (SN), Medium Number (MN), Large Number (LN) and Very Large Number (VLN). Figure (6.21) illustrates the network structure of ANFIS 3 (MATLAB output).

When the number of membership functions increases, this is accompanied by an increase in non-linear parameters alongside the number of rules and the linear parameters simultaneously. In other words, the topology of the network will increase; thus, the model is to be classified as high complexity. Looking at the number of MFs assigned to each input of ANFIS 3 and the network structure shown in (Fig. 6.21), ANFIS 3 can be classified as

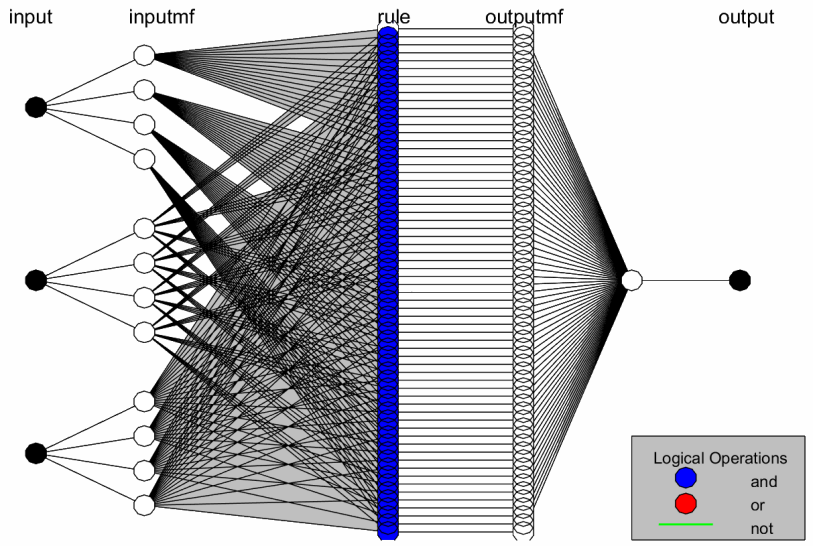


Fig. 6.21 ANFIS 3 structure.

high complexity model. Consequently, the total number of parameters of this validation model has raised to (280). This increases in parameters occurred due to the use of four Gaussian MFs for each input. It means there is a higher number of both linear and non-linear parameters to be resolved. An expansion into (156) weekly data would not have covered the (280) total parameters for this model. Given the temporal nature of the data, it made sense to consider expansion that followed a monthly-weekly-daily pattern. Therefore, sampling the RBF fits at (365 x 3) data points provided three year's worth of daily data on which to evaluate the approach. Solving ANFIS 3 using daily data (i.e. 1095 samples) with all possible combinations of expansion and scaling methods will provide eight experiments. The results of these experiments are shown in Table (6.7).

By reviewing the results listed in Table (6.3), we can observe that increasing the model complexity has a positive effect on the non-expanded data compared to ANFIS 1. The prediction error improved from 0.4406 at experiment (1) to 0.1181 at experiment (17). Particularly, ANFIS 3 results can be discussed as follow:

1. ANFIS 3 Local performance (Chebyshev scaling), experiments (17-20):

    - All expanded data (experiments 18-20) outperformed non-expanded data (experiment 17).
    - Both Linear RBF and Multiquadric RBF (where $\rho_1 = 0.01$ and $\rho_2 = 0.2$) provides approximately up to 16% of improvements.

Table 6.7 Results of the third validation model (ANFIS 3).

Experiment	Data	Samples	Expansion	Scaling	NRMSE	R^2	Performance	
							Local	Global
17	Monthly	36	——	Ch	0.1181	0.7922	——	0%
18	Daily	1095	Lin RBF	Ch	0.0646	0.9536	↗ 16%	↗ 16%
19	Daily	1095	Cub RBF	Ch	0.1162	0.8357	↗ 4%	↗ 4%
20	**Daily**	**1095**	**Mul RBF**	**Ch**	**0.0666**	**0.9566**	↗ **16%**	↗ **16%**
21	Monthly	36	——	Z-s	0.1218	0.7984	——	↗ 1%
22	Daily	1095	Lin RBF	Z-s	0.0987	0.9360	↗ 14%	↗ 14%
23	Daily	1095	Cub RBF	Z-s	0.0848	0.8896	↗ 9%	↗ 10%
24	Daily	1095	Mul RBF	Z-s	0.0943	0.9531	↗ 15%	↗ 16%

- The Cubic RBF provides up to 4% of performance improvement, which 12% less when compared to the other two expansion approaches. Again this can be attributed to the over-smoothing of the interpolated data at the turning points.

2. ANFIS 3 Local performance (Z-score), experiments (21-24):

   - All expanded data (experiments 22-24) performs better than non-expanded data. Approximately 9% to 15% improvement in prediction accuracy compared to experiment (21).
   - The best performance resulted from using the Multiquadric RBF approach, which gives an improvement of up to 6% compared to the Cubic RBF and 1% compared to the Linear RBF.

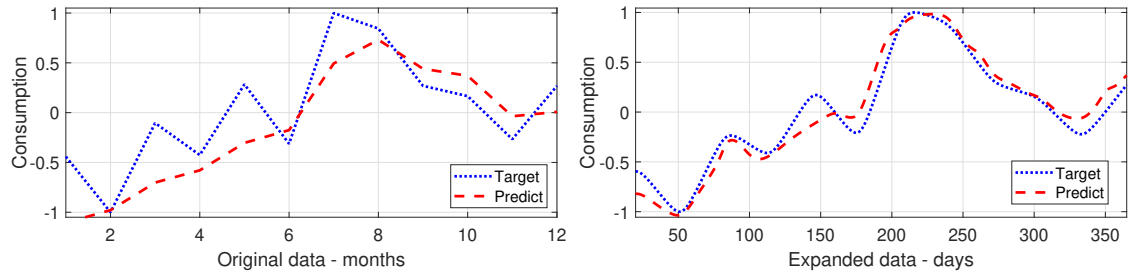
3. ANFIS 3 Global performance, where all experiments (18-24) will be compared with experiment seventeen as a base lowest performance model.

   - Looking at both NRMSE and R^2 in Table (6.3), we can observe evident improvements in the enhanced data sets when compared to the original data.
   - A comparison of Linear RBF expansion results for both scaling methods showed that Chebyshev scaling outperformed the Z-scores by nearly 2%. Whereas, when using Cubic RBF, the Z-scores outperformed Chebyshev by 6%, while

the Multiquadric RBF produced a similar performance increase for both scaling methods of approximately 16%.

- It can be noted that three different experiments (18, 20 & 24) were very close in their results; in which their global performance were approximately up by 16%. However, by considering both NRMSE and R^2, the highest performance among these three experiments was in experiment twenty, represented by its prediction error of (0.0666) and highest R^2 of (95.66%).
- An interesting difference, however, is the shift in performance due to the way in which the data has been scaled. Here we see that the result of experiment twenty, which was obtained by using the Chebyshev scaling, combined with the Multiquadric RBF approach (where $\rho_1 = 0.01$ and $\rho_2 = 0.2$), has outperformed its rivals.

The result of experiment twenty will be introduced in more detail and compared to experiment seventeen in order to show the improvement in prediction accuracy. Figure (6.22) shows how well the model can predict the consumption for 2017 for the expanded model (Fig. 6.22b), when compared with the best non-expanded model (Fig. 6.22a). A significant improvement in performance can be seen after applying our proposed model. It is very clear that the curve of the predicted values is very close to the target values with an evident rate of convergence.



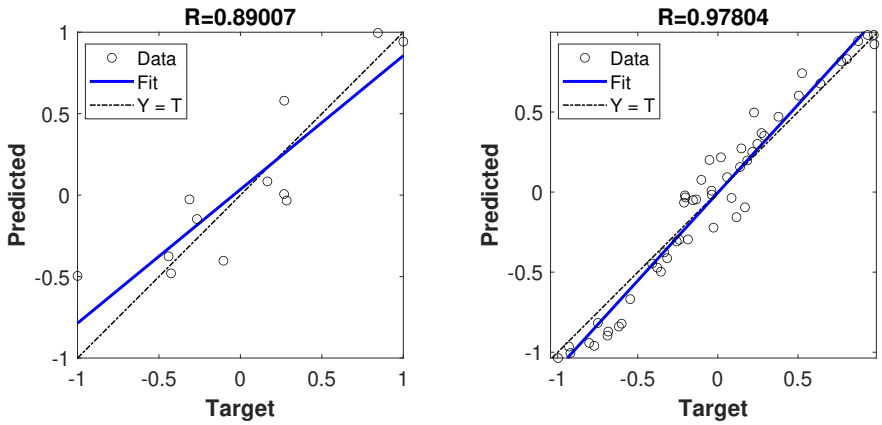
(a) Before expansion.

(b) After expansion.

Fig. 6.22 Targeted and predicted gasoline consumption for validation data of ANFIS 3: (a) experiment 17 (b) experiment 20.

Figure 6.23 shows the relation between target and predicted consumption using original data (experiment 17) and expanded data (experiment 20). The regression model of experiment twenty (Fig. 6.23b) shows how nearly 95.66% of the variability (equal to R = 0.97804) is explained by the model when using expanded data. Whereas, experiment seventeen (Fig. 6.23a) shows (R = 0.89007) when using the original (non-expanded) data. These

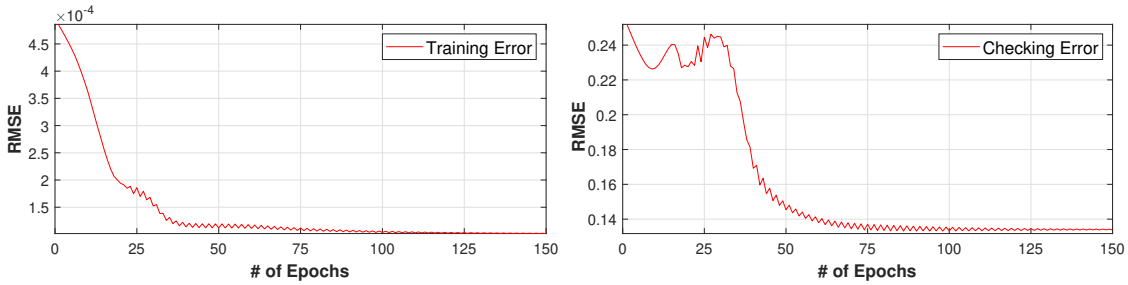
regression scatter plots illustrate the goodness of our model in reducing the deviation of the predicted data.



(a) Before expansion. (b) After expansion.

Fig. 6.23 Regression scatter plot of the targeted and predicted gasoline consumption for validation data of ANFIS 3: (a) experiment 17 (b) experiment 20.

Figure (6.24) gives an indication for a typical ANFIS rate of convergence of experiment (20). Figure (6.24a) illustrates the rate of decay of the residuals for the training data (training error). It shows the greatest reduction in the training error at the start of the modelling process with finer tuning occurring later. Figure (6.24b) demonstrates the rate of convergence to the 2017 test data (checking error). A typical ANFIS behaviour is showing initial oscillations in the prediction error as the parameters begin to train before settling to a gradual rate of convergence.



(a) Training error. (b) Checking error.

Fig. 6.24 Typical rate of convergence ANFIS 3: (a) Training error. (b) Validation Error.

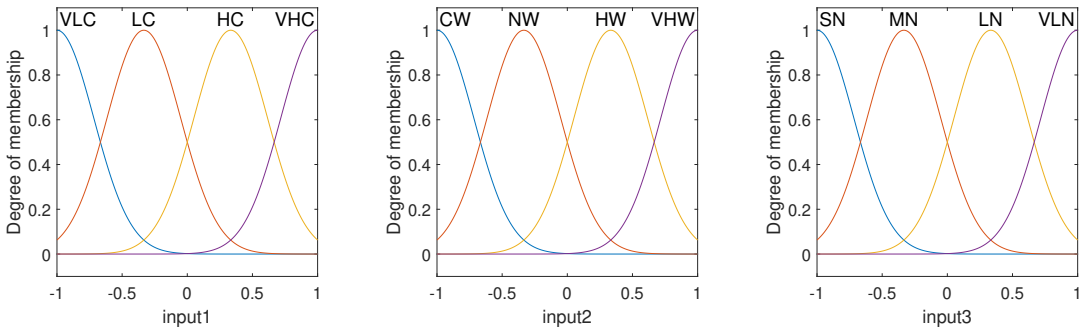
Table (6.8) illustrates the initial and final (optimal) values of the premise parameters for experiment twenty. The Gaussian function has been used for all three inputs. Each input contains four membership functions. Each membership function has two parameters

of (σ& c) (see Equation 3.28). Where, c determines the centroid of the bell shape and σ represents the width.

Table 6.8 Premise parameters - experiment 20 - ANFIS 3.

Input	MF Type	MFs	Initial Parameters		Optimal Parameters	
			σ	c	σ	c
Capacity (Input 1)	Gaussian	(VLC)	0.2831	-1	0.2833	-0.9901
		(LC)	0.2831	-0.3333	0.1663	-0.4733
		(HC)	0.2831	0.3333	0.2139	0.2707
		(VHC)	0.2831	1	0.3087	0.9668
Temp. (Input 2)	Gaussian	(CW)	0.2831	-1	0.2811	-0.9808
		(NW)	0.2831	-0.3333	0.258	-0.2893
		(HW)	0.2831	0.3333	0.3263	0.3687
		(VHW)	0.2831	1	0.3087	0.9957
Cars (Input 3)	Gaussian	(SN)	0.2831	-1	0.237	-1.007
		(MN)	0.2831	-0.3333	0.0767	-0.3836
		(LN)	0.2831	0.3333	0.1539	0.2368
		(VLN)	0.2831	1	0.2668	0.9819

The plot of these initial and optimal MFs for all three inputs of ANFIS 3 can be found in Figures (6.25 & 6.26).

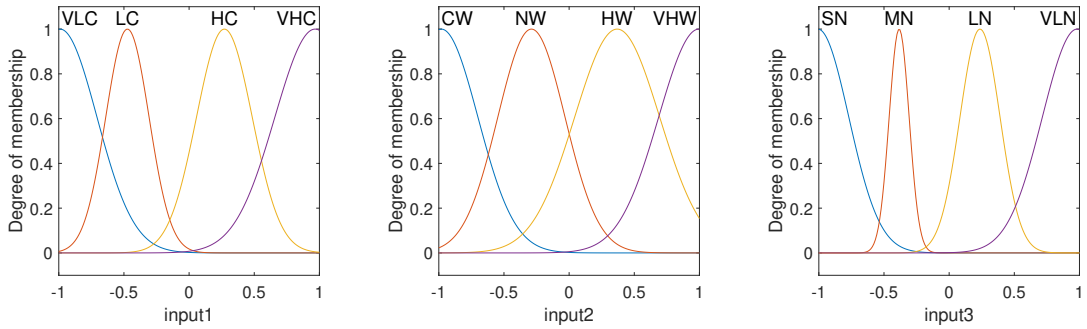


(a) Capacity. (b) Temperature. (c) Cars.

Fig. 6.25 Initial membership functions of ANFIS 3 inputs.

By reviewing the shapes of the final (optimal) membership functions showed in Figure (6.26) we can find the following:

- Figure (6.26a) shows the four trained MFs for capacity. It can be seen that some MFs show significant changes.



(a) Capacity. (b) Temperature. (c) Cars.

Fig. 6.26 Final (trained) membership functions of ANFIS 3, experiment 20.

- The membership function of very low capacity (VLC) shows slight changes. These changes are represented by a small increase in value of ($\sigma_{VLC_{initial}} = 0.2831$) to ($\sigma_{VLC_{optimal}} = 0.2833$) and ($c_{VLC_{initial}} = -1$) to ($c_{VLC_{optimal}} = -0.9901$). These changes have made the MF width slightly shrink together with a little movement toward the right, which can barely be noticed.
- The low capacity (LC) membership function has significant changes. The value of σ decreased by approximately 12% (difference between ($\sigma_{LC_{initial}} = 0.2831$) and ($\sigma_{LC_{optimal}} = 0.1663$). This effected (LC) by shrinking the width of the bell shape to be notably narrower. The value of c decreased by nearly 14% (difference between ($c_{LC_{initial}} = -0.3333$) and ($c_{LCoptimal} = -0.4733$). This reduced the value of the centroid of the Gaussian curve which, in turn, moved the position of the bell shape centre toward the left on the corresponding value on the x-axis coordinate.
- The high capacity (HC) membership function shows evident changes. The value of σ decreased from ($\sigma_{HC_{initial}} = 0.2831$) to ($\sigma_{HC_{optimal}} = 0.2139$) as well as the value of c from ($c_{HC_{initial}} = 0.3333$) to ($c_{HC_{optimal}} = 0.2707$). This caused approximately 7% decrease in the width of the bell shape and moved its centre toward the left by 0.0626 points on its corresponding value on the x-axis coordinate.
- The membership function of very high capacity (VHC) showed slight changes. The σ value increased by nearly 2.5%, and the c value decreased by a very small amount of 0.43%. These made the bell shape a little broader and moved it slightly toward the left.

The trained MFs of the first input (capacity) shows significant changes in the crossover points of the bell shape curves. Consequently, the overlaps between the MFs were

almost squeezed. This will make the fuzzification areas more specific, which can help in providing more efficient firing strengths.

- Figure (6.26b) shows the final trained MFs of temperature. At first glance, the optimal MFs give the impression that there is no significant changes, when compared to the initial MFs (Fig. 6.25b). However, if we look closely, we will notice that:
  - The width of the natural weather (NW) membership function bell has been shrunk by 2.5%, and the shape has moved slightly toward the right.
  - The area of hot weather (HW) membership function extended by nearly 4% and moved toward the right.
  - The area of very hot weather (VHW) membership function extended by nearly 2.5% and moved slightly toward the left.

  As a result of these changes, all the overlap area between the (CW & NW) shrinks and the crossover point has decreased. While the intersection areas between (NW & HW) and (HW & VHW) have been extended and the crossover points have increased.

- The optimal MFs of the third input "number of cars" (Fig. 6.26c) showed significant changes. The medium number (MN) and the large number (LN) MFs are the most considerable shifts. These changes are similar to some extent to the optimal MFs of the first input (Fig. 6.26a), especially for the second and third MFs. It is not difficult to see that the medium number (MN) bell shape is significantly compressed by nearly 20% as well as the large number (LN) by 13%. They have both moved toward the left.

Again, as a result of those significant changes, the overlap areas were considerably squeezed. Furthermore, all the crossover points moved down toward the x-axis. In other words, the fuzzification range for each membership function became more specific. This yields more accurate firing strengths when composing the fuzzification process with other MFs.

The above analysis shows that increasing the model complexity has a positive effect in optimizing the premise parameters. The expanded data performed significantly better than the original data. It showed that the first "capacity" and the third "number of cars" inputs have a significant impact on the gasoline consumption problem while the second input "temperature" has less of an impact than the other two inputs when the model complexity is increased.

6.3.8 Model Validation

This section demonstrates alternative data sets that have been used to validate our proposed model. A case study formed from a real-life example of natural gas demand prediction presented by [16]. The data used in this case study is taken directly from their work. Based on the most standard indicators used in the literature, the authors designed their FIS prediction model to include five inputs (independent variables), which are: national income, consumer price index, gross domestic product, population, and demand taken from the previous year. There is one target output (dependent variable) which represents the natural gas demand. They collected 34 samples of actual data per input. Their data was collected annually and covered the period from 1973 to 2006. The data was gathered from Energy Balance 2006 of Iran and Statistical Centre of Iran. We use this data to evaluate the effectiveness of our approach for both expanded and non-expanded data. It should be noted that the original work applied limited pre-processing of the data prior to modelling. This consisted of scaling the data to the range [0, 1]. Furthermore, their ANFIS model used a constant output in the forward process to calculate the consequent part of their model, whereas we are using the linear (first-degree polynomial function) to calculate the output. The authors did not provide any information about which generation method they used to construct the ANFIS model; such as Grid Partitioning, Subtractive Clustering, or FCM Clustering for example. For our experiment, the first 29 years' data (from 1973 to 2001) have been used for the training process, with the remaining five years used to test the capability of the trained model to accurately predict the demand in those years.

A total of 9 experiments have been carried out using our proposed model. We restrict all evaluating experiments to use the Chebyshev scaling method, Gaussian membership functions and the modified Multiquadric expansion method; where we show results for a range of values of ρ_1 and ρ_2. This simple but effective trial and error approach allows us to show how the modified Multiquadric method can be used to improve the prediction accuracy and provide even better results when tuning its parameters. The original work demonstrated a best prediction accuracy of 0.0029. After squaring the NRMSE for experiment 22 we obtain a comparative measure of 0.0011. Not only is this a significant improvement in prediction accuracy, but it is obtained by applying a much simpler model combined with more intelligent pre-processing. We showed that almost 99% of the variability (equal to R = 0.9988) is explained by the model constructed using the expanded data; compared to 94% (R = 0.9722).

6.3.9 Discussions and Conclusions

This chapter aimed to demonstrate the importance of using carefully chosen expansion and scaling methods as pre-processing data techniques in order to develop and enhance the performance of ANFIS prediction. Three radial basis function approximation methods and two scaling methods have been combined with ANFIS to produce our proposed model 1, shown in this chapter. A real-life problem of predicting the consumption of petroleum products has been chosen as a test case. Three of ANFIS's models have been chosen as validation models. These validation models were chosen with different levels of complexity in order to evaluate the proposed model. A total of twenty four experiments have been carried out and solved using all potential combinations of using different expansion and scaling methods. By reviewing the overall results of the validation models, we can conclude the following:

1. Changing the data pattern, whether through scaling or expanding has improved the ANFIS performance and consequently improves the results.

2. Using RBF expansion methods provides two positive impacts. First, to overcome the over-fitting problem. Second, to improve the prediction accuracy, mainly when the input data are scarce.

3. In all experiments, looking at the global performance, the use of expanded data performed better than the non-expanded data.

4. When care is taken to expand the data using carefully chosen values of the Multiquadric RBF parameters (i.e. ρ_1 & ρ_2). The Multiquadric RBF provides better performance than the Linear and Cubic RBFs.

5. The Cubic RBF almost always provides the lowest performance compared to the Linear and Multiquadric. This is potentially due to the over-smoothing problems of the expanded data at the turning points.

6. Even slight changes in the ANFIS parameter values (after being optimized) provides evident improvements in the prediction accuracy and system performance.

Chapter 7

Outliers Mitigation Model for ANFIS Optimization: Proposed Model 2

7.1 Introduction

An outlier can be defined as one or more observed values that shows significant deviation from the underlying trend of a specific data set that it belongs to. In other words, it is a data point that lies at an abnormal distance from the shape of the population. There are several reasons why outliers can occur. These can be mistakes caused by human error, mechanical faults, deceptive behaviour, the behaviour of the system, or natural deviations in populations. Generally, outliers are known as one of the affecting factors that may reducing the ability to produce an accurate model. However, this can be true up to some extent, because not all outliers are to be considered as as invalid values, and has to be removed. For instance, in some cases, when the outliers are generated by the system, it might be illustrating something important about the system behaviour. Removing them in this case means that we miss some important observations. Therefore, only the outliers that are affecting the model are to be mitigated. Many approaches can be used for detecting outliers; such as graphing methods, mathematical modelling, statistical models, and so forth. These approaches are almost identical in their fundamental purpose [54]. However, in this work we intend to introduce an algorithm that performs the outliers removal as a solution method for the outliers problem, rather than discussing the detection methods. This can be achieved by using the model fitting approach. In other words, this chapter aims to propose a robust outlier mitigation model that can help in modelling noisy data and provide a better fitting model to be used in ANFIS.

The use of RBFs (Linear, Cubic, and Multiquadric) as the basis functions for explicit interpolation was introduced in the previous chapter as part of the first proposed model. Practically, the use of these techniques as a direct expansion methods works on interpolating the data utilizing RBFs (specifically in our case) for producing expanded data sets. This can be achieved by using a number of RBF centres that are equal to the original data points when calculating the Euclidean distance between the data points and the function's centres.

Mathematically speaking, for interpolation, we intend to solve a square system of linear equations, in which the number of basis functions is equal to the number of data points (Sec. 6.2.2.1). This will provide a square system of linear equations. Each basis function in this system is to be accompanied by a coefficient (i.e., unknown parameter). In other words, the number of unknown parameters is equal to the number of original data samples. These parameters are to be modifiable and represents the degrees of freedom of the model. We aim to optimize these coefficients in order to provide the best fitting model to the original data. Increasing or decreasing the number of degrees of freedom gives the model more or less flexibility to twist and change when fitting the data. Therefore, when the data are stable and do not have significant noise or outliers, then it is possible to set the degrees of freedom

equal to the number of data points. This will allow the model to go through every single point of the data sample when interpolating.

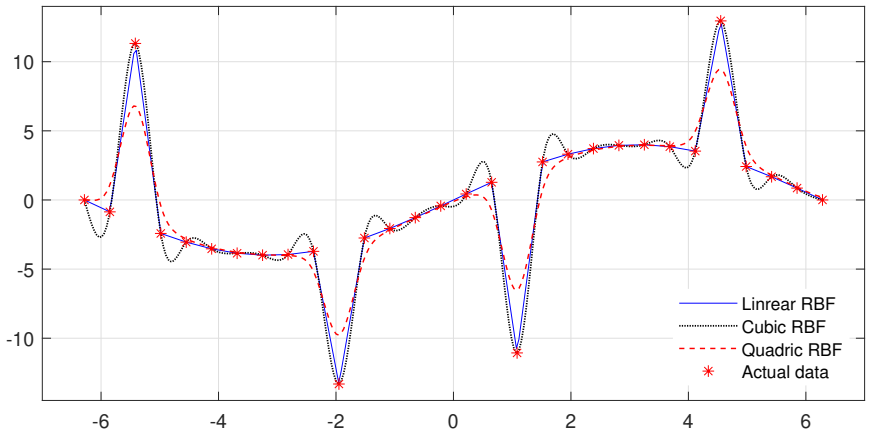


Fig. 7.1 Direct RBFs fitting for a simple sinusoidal curve modified to contain four outliers.

A significant problem in data modelling is dealing with the effects of outliers (noise) in the data [35] [164]. A simple example is shown in Figure (7.1). Here we have a sinusoidal function, sampled at 30 equally spaced points on the interval $[-2\pi, 2\pi]$. These data points have been modified to contain four outliers distributed randomly. These outliers are to be considered as the points that do not represent the underlying trend of the data. Three RBF types (i.e. Linear, Cubic, and Multiquadric) are to be fitted to those points using interpolation. As can be observed, the outliers have had a significant impact on the fitted curves. We can see that the impact of the outliers is such that it has moved the fitted curve away from the underlying shape of the data. Because we are interpolating, the model has sufficient degrees of freedom to fit the data points accurately, which means the curve passes through all of the accurate points but also through all of the outliers – except the Mulitquadric which we discuss below.

Referring to Figure (7.1), it is not difficult to see that the Linear RBF (blue line) and Cubic RBF (black dotted line) passes through all of the original data points when interpolating. This should be expected due to the nature of these two interpolation techniques. In contrast, the Mutiquadric RBF (red dashes line) performs better (though not much) due to the modification of its parameters. However, when near the outlying data samples, it is still showing a tendency to deviate away from the underlying shape of the data when approximating. Consequently, we can conclude that, when the data have noise or outliers, interpolation is not the right choice for modelling. So, we need to find a method that can model good data and ignore the noise and outliers. This can be achieved by restricting the number of degrees of freedom, which means we have to choose a model that has a fewer number of parameters. In other

words, it means that the square system of equations, providing an exact solution, is no longer possible. This restriction will allow the model to avoid modelling the noise and to recover the underlying curve, simultaneously.

Least-squares (LS) is known to be one of the most effective methods in reproducing curves when the original data contains a small amount of noise. Standard least-squares (SLS) is a common and straightforward choice in data approximation, as it is simple to interpret and to apply. The least-squares error function takes the form

$$E_1 = \sum_{i=1}^{m} e_i^2, \quad \text{where} \quad e_i = y_i - F(x_i), \tag{7.1}$$

where, $F(x_i)$ can be any fitting function. Here we were using the RBF fitting functions (Linear, Cubic, and Multiquadric), which we introduced in Equations 6.9 to 6.11 as the basis functions of the standard least-squares (SLS). Again, the LS method uses fewer basis functions than the data points, which means it is not a square system of equations, and so there are fewer coefficients than there are data points. Values for the coefficients are determined by minimising E_1 (7.1). However, we can see that the result of squaring any particular e_i, recorded at an outlying point, is a fitting curve that moves away from the underlying true path of the data.

Figure (7.2) illustrates the SLS fitting using three RBFs as basis functions for the same sinusoidal data points shown in the previous example. Here we reduced the number of degrees of freedom of the model from 30 to 6. These six points represent the updated centres of the model which allows the function centres to be distributed evenly though the data range. The blue line represents the SLS fitting using the Linear RBF; the black dotted line demonstrate the SLS fitting using the Cubic RBF. In contrast, the red dashed line shows the SLS fitting using the modified Multiquadric RBF. Compared to Figure (7.1), it can be noticed that SLS is performing much better than the interpolation approach when the data contains outliers. However, the influence of outliers on the SLS fitted curves is still notable and has to be considered. It is not difficult to see how the first and fourth outliers are pulling the curves up towards them; showing on the left and right side of this fitting model. At the same time, the second and third outliers in the middle are pulling the fitting curve down.

The reason for not getting a good fit is due to the nature of the Least-squares approach. The SLS is working on minimizing the squares of errors between the fitted curve and the original data points. When the data has outliers, the error values between the fitted curves and these outliers are large, particularly after squaring them. The SLS methodology is trying to choose a set of parameters for the model that minimizes the sum of the squared errors. The problem here is there are four of those squared errors that are disproportionately large,

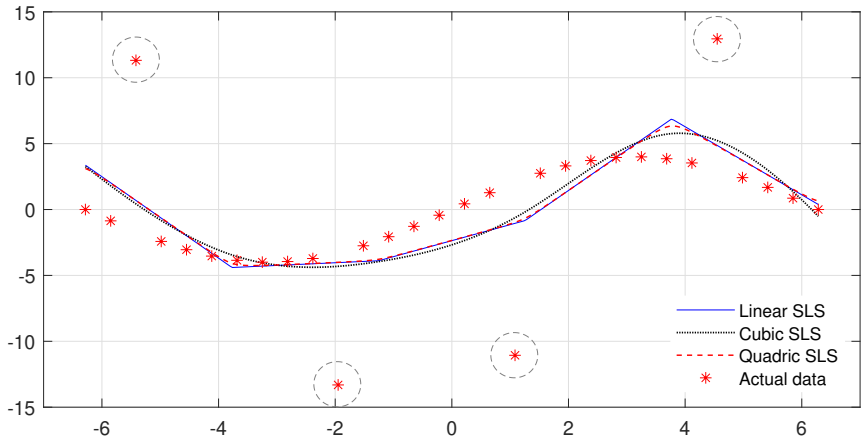


Fig. 7.2 SLS fitting using RBFs for a simple sinusoidal curve modified to contain four outliers.

caused by the outliers. Therefore, in order to reduce the overall least squares error, the fitting curve moves towards those points. Therefore, although the SLS approximation is considered as one of the best methods in dealing with small noise (Gaussian noise), it does not fair well when the data contains outliers - even though this is still better than direct RBF interpolation. Based on this, we need to find an appropriate method that can address the problem of outlier mitigation.

There are some suggestions in the literature for dealing with this kind of problem. For example, solution schemes that seek to minimise a different form of error function can be employed [19].

$$E_2 = \sum_{1=1}^{m} |e_i|. \tag{7.2}$$

Because this error function seeks to minimise the sum of absolute deviations of the fitting curve to the data, it is extremely good at ignoring outliers and fitting the trend. However, there is a disadvantage, due to the fact that E_2 can not be differentiated. This means that we cannot formulate a simple solution scheme, that is as straight forward as SLS for solving for the unknown parameters in the model. Because a traditional least-squares approach can not now be used, other techniques, such as linear programming must be considered. These methods are not as straightforward to apply as SLS and so tend to be avoided by practitioners who do not have strong maths skills.

In order to overcome this problem, we are proposing a second model which uses a modified Least Squares approach, which we call Transformed Least Squares (TLS), and we introduce it in the coming sections. In this chapter, all equations, work, tables, figures, and diagrams have been developed and created by the author.

7.2 Development of Model 2

7.2.1 Model 2 Structure

Here we propose an expanded version of model 1, introduced in chapter 6. The pre-processing model (layer 0 in Fig. 6.1) is to be modified to deal with data sets which are significantly noisy or contain outliers. This has been achieved by adding the Transformed Least Squares (TLS) to the pre-processing layer structure. At this stage, experts needs to have enough understanding of the underlying data in order to make a decision about which approach (algorithm) to be applied. In other words, the raw data are to be investigated by the experts in order to find whether it has outliers or not before proceeding to the re-sampling and scaling models. If yes, the TLS is to be applied as an outlier mitigation model. Then, the re-sampling model, using radial basis functions (RBF) (Linear, Cubic, and Multiquadric) is applied as before as the data expansion model to expand the scarce sample. The two previously discussed scaling methods will then be applied if the data has a significant variance in its values. These three stages (i.e. outlier mitigation, expansion, and scaling) represents the components of the pre-processing (layer 0) of the second proposed model. The processed data (output of layer 0) are to be propagated to the ANFIS layers and separated into training and checking sets in order to process the hybrid learning rules. Figure (7.3) illustrates the flowchart of different steps performed to implement the second proposed model.

7.2.2 Proposed Outlier Mitigation Model

7.2.2.1 Transformed Least-Squares (TLS)

Expanding data sets that contain any noise cannot use interpolation approaches, as the noise will influence the predicted data. Of course, this does assume that the noise is unwanted and detrimental to the expansion process. Taking a least-squares approach, as discussed above, can help for the noise of a particular type, but it does not deal with all cases - such as outliers. It is therefore appropriate that, in this work, we present a modification to the least-squares approach that enables outliers to be considered straightforwardly.

We present a modified, or transformed, least-squares approach (TLS) to model data that is believed to contain a small number of outliers [62]. Rather than minimising the form E_1, we minimize,

$$E_3 = \sum_{i=1}^{m} [G(e_i)]^2, \tag{7.3}$$

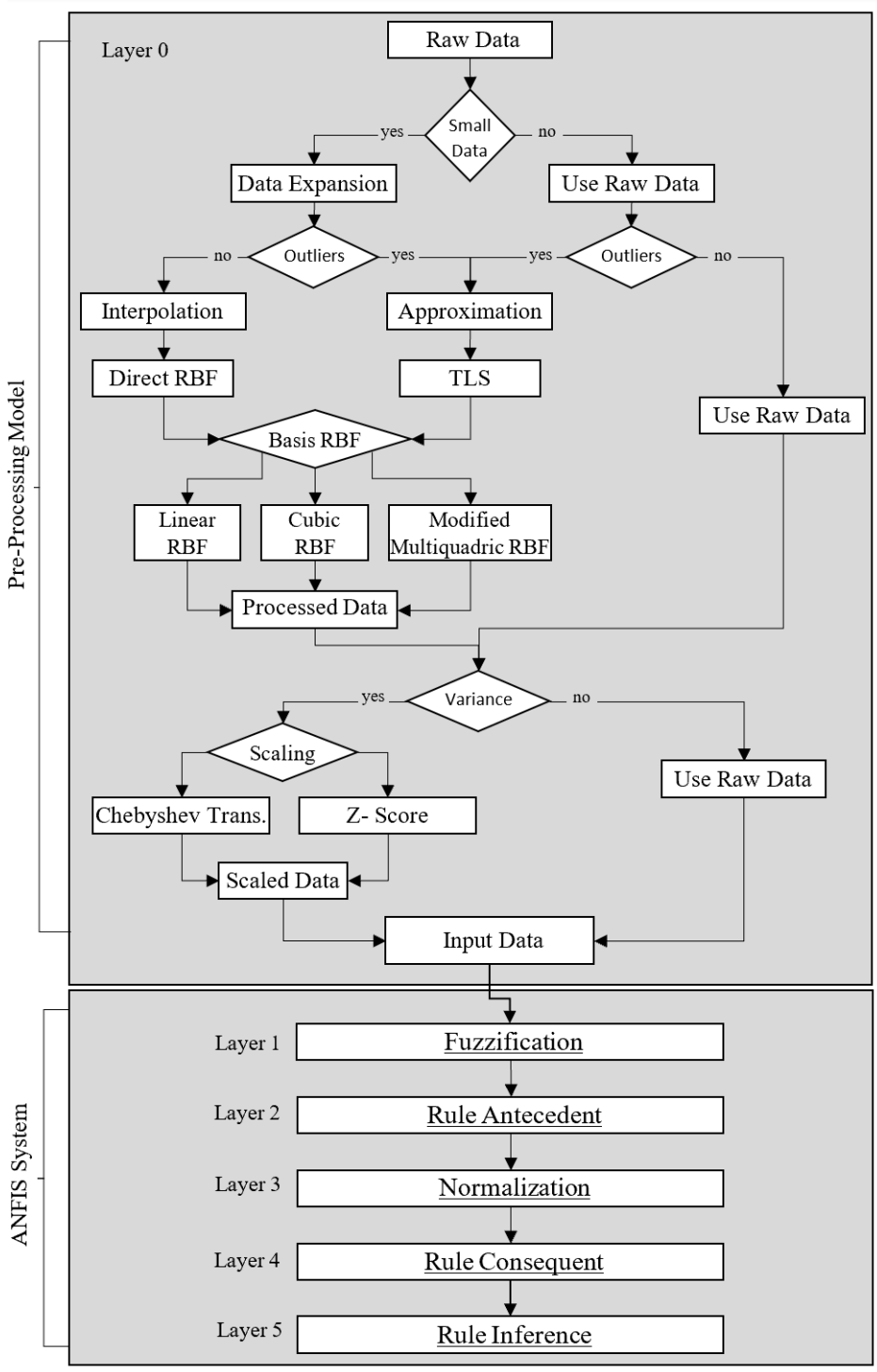


Fig. 7.3 Th second proposed model structure.

where e_i is as defined in (7.1) and

$$G(e_i) = \frac{e_i}{\sqrt{1+\gamma e_i^2}}, \tag{7.4}$$

is the transformation function. γ is a fixed constant and should be chosen based on the asymptotic bound required on any individual contribution to the error function (7.3). To determine the coefficients in (6.7) we find a solution using iteratively re-weighted least-squares. Specifically, we set

$$k_i = \frac{1}{1+\gamma e_i^2}, \tag{7.5}$$

and solve

$$\min_{\mathbf{c}} E_3 = \sum_{i=1}^{m} k_i e_i^2. \tag{7.6}$$

We begin by choosing an initial set of random coefficients $c_1, c_2, \ldots, c_n$, $(n < m)$ to allow the multipliers k_i to be evaluated for the first iteration. This is a similar approach to initialising the weights in a neural network. A random selection of small values is usually fine to start the iterative approach. We then proceed by repeating Steps 1 to 5 below:

Step 1. Calculate the numbers k_i.

Step 2. Construct a diagonal matrix K that has all of the numbers $k_1, k_2, \ldots, k_m$ down the diagonal.

Step 3. Solve the weighted system of equations using standard least-squares

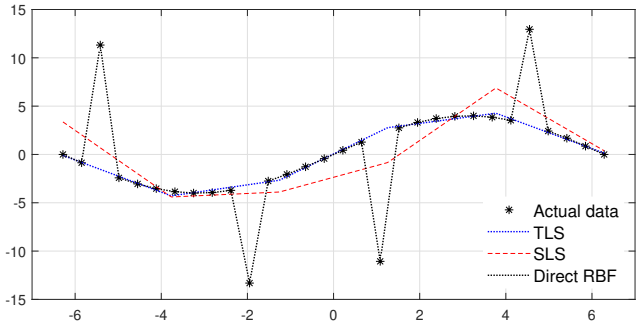
$$KA\mathbf{c} = K\mathbf{y}, \tag{7.7}$$

for the coefficients c_1 to c_n.

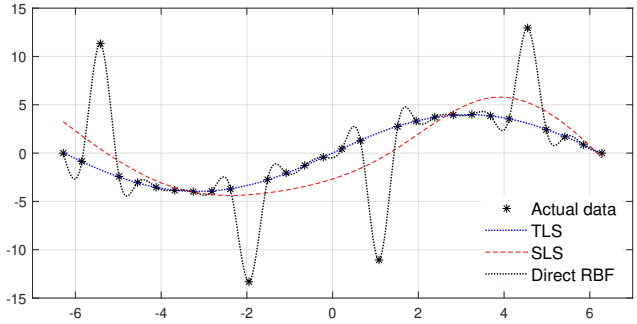
Step 4. If the difference in the value of the error function E_3 for two consecutive iterations is negligible then stop.

Step 5. Otherwise, calculate the errors e_i and repeat from Step 1.

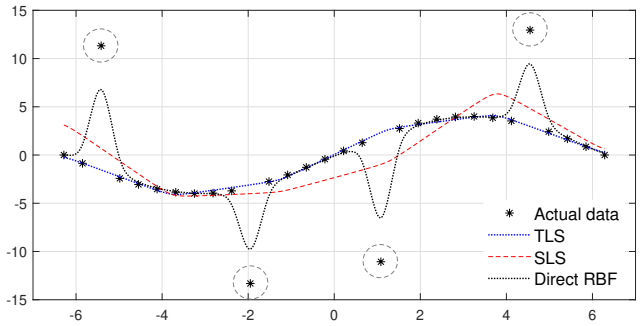
The usefulness of this approach is two-fold. Firstly, taking a least-squares approach, rather than strict interpolation, allows us to reproduce more accurate data from the original data that may contain Gaussian noise. Secondly, data that exhibits either mixed Gaussian noise and outliers or simply outliers are dealt with effectively. The choice of $n < m$ can be used to control the amount of smoothing that needs to take place. An application of the TLS method is shown in Figure (7.4) where a significant improvement over SLS and direct RBFs is demonstrated. Speaking of the sub figures in Fig. (7.4), we used the same sinusoidal function illustrated in the previous examples in Section (7.1) to create thirty data samples, which includes four outliers. Here, each sub figure shows a comparison of the performance of the two approximation approaches (i.e. SLS and TLS) as well as the direct RBF interpolation.



(a) Linear Fitting.



(b) Cubic Fitting.



(c) Modified Multiquadric Fitting.

Fig. 7.4 SLS TLS RBFs comparison.

Figure (7.4a) illustrates using the Linear RBF as the basis of the fitting function for modelling the data using SLS (red dashes line), TLS (blue dotted line), and direct RBF (black dotted line). Figure (7.4b) shows the curve fitting resulting from using the Cubic RBF for the same three approaches. While Figure (7.4c) demonstrates the use of the modified Multiquadric RBF to fit the data. We can clearly see that the asymptotic properties of the TLS fitting method allow a more robust model to be formed to reflect the underlying shape of the data.

To see why a transformed least-squares approach can produce better results, we can examine the properties of the function

$$\frac{e_i^2}{1+\gamma e_i^2}, \qquad i = 1,2,3,\ldots,m.$$

When the value of any point-wise error e_i is extremely small, this function behaves like e_i^2. So,

$$\text{as } \; e_i \longrightarrow 0, \qquad \frac{e_i^2}{1+\gamma e_i^2} \longrightarrow e_i^2.$$

When the point-wise error e_i is large, then

$$\text{as } \; e_i \longrightarrow \pm\infty, \qquad \frac{e_i^2}{1+\gamma e_i^2} \longrightarrow \frac{1}{\gamma}.$$

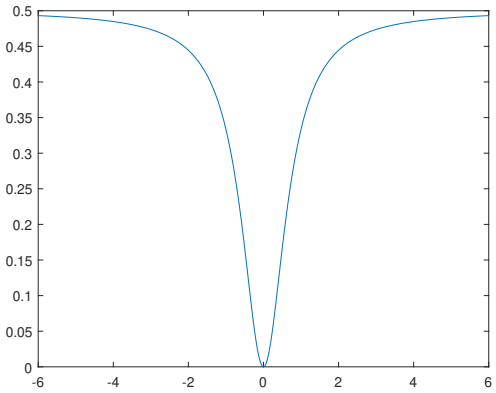


Fig. 7.5 The squared transformed least-squares error function for $\gamma = 2$ showing an asymptotic upper bound of $\frac{1}{2}$ as the magnitude of errors increase.

When the point-wise errors are small, the function behaves like a standard least squares operator. As the point-wise errors get large, their individual contribution to the error function E_3 is asymptotically bounded by $1/\gamma$. Therefore, the extent to which any given point-wise error can effect the position of the fitting curve is limited by choosing an appropriate value for γ. A plot of the error function $[G(e)]^2$ for $\gamma = 2$ is shown in figure 7.5.

7.3 Implementation of Model 2: Case Study

So far, we have introduced the TLS approach and demonstrated its performance in dealing with outliers. However, we intend to evaluate the performance of this method by applying it

to real data. Therefore, the gasoline prediction problem discussed in the previous chapter will be used for this purpose. The same input-output structure of the proposed ANFIS of the gasoline forecasting introduced in Section (6.3.2) will be utilized (see Fig. 6.4). There will be three input variables, i.e., capacity, temperature, and the number of cars, which represents the affecting factors; and the output variable will be the predicted consumption rates. The data of the input variables will be modified to contain outliers in order to explore the effectiveness of using the TLS on ANFIS performance.

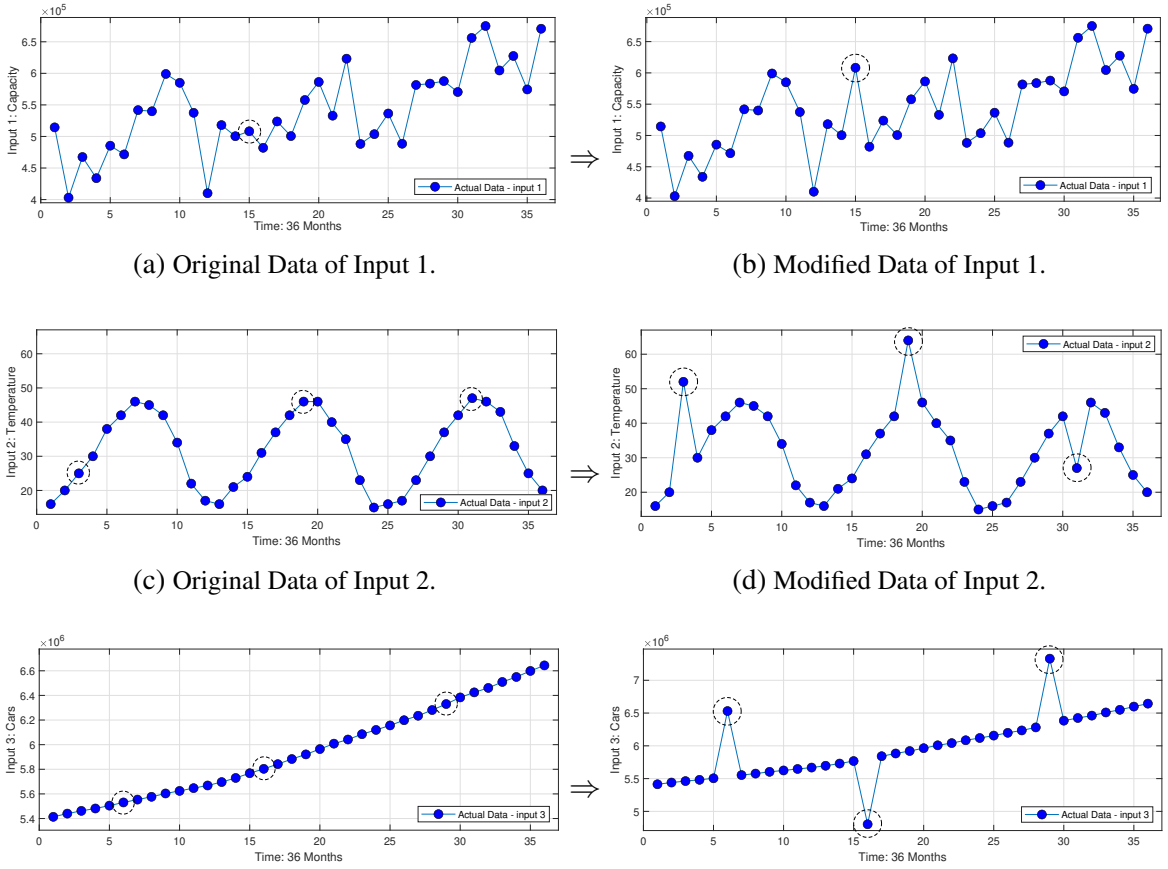
7.3.1 Modified Data

As a test case, we are using the same data that been provided for the first proposed model shown in Section (6.3.3) of the previous chapter. The historical data of four variables (i.e. three inputs and one output) have been supplied. Each variable has thirty-six data samples representing the monthly recorded value covering a three years time period from 2015 to 2017. The data set of the first input (capacity) is to be modified to contain one outlier. Whereas the second (temperature) and third (number of cars) data sets are to be modified to contain three outliers in each set. This modification on the original data set will create new sets that can be used to validate the TLS proposed model. Figures (7.6a to 7.6f) shows the original and modified data of the three inputs, respectively.

7.3.2 Outlier Mitigation Performance

In this section, we intend to apply our second proposed model using the TLS approach for modelling the modified gasoline data. As mentioned before, modelling the data using interpolation means that that the number of unknown parameters (i,e. degrees of freedom) is equal to the number of original data points. So, we end up with a square system of linear equations. In other words, all the data points are to be considered as *RBF centres*. Thus, we produce a fitting curve that passes through all the data points, interpolating the original data (see Fig. 7.1), including the outliers.

When using TLS, we restrict the linear system by reducing the number of degrees of freedom (RBF centres) to a specific number (usually much less than the number of data points). This will produce a sufficient curve fitting that maintains the underlying shape of the data and ignores the effect of outliers. Accordingly, choosing the right number of centres is to be considered as the primary key to gaining a good TLS fitting model. Therefore, it is vital to experiment with all the possible numbers of centres by trial and error until we achieve the best fit.



(a) Original Data of Input 1.

(b) Modified Data of Input 1.

(c) Original Data of Input 2.

(d) Modified Data of Input 2.

(e) Original Data of Input 3.

(f) Modified Data of Input 3.

Fig. 7.6 Modified Data of all Inputs.

Due to the nature of our data, we are classifying the data sets of the three inputs into two groups in order to explore the possibility of applying the TLS and its performance. The criteria of this classification will depend on the data pattern.

- **Smooth pattern:** when the data has a continuity in its trend (linearly or non linearly), then its pattern can be considered as a smooth curvature. One of the advantages of this type of data is that it is easy to identify the outliers, even by eye. Accordingly, looking at the inputs data, the temperature and number of cars, both can be classified as a steady pattern. The TLS has been applied to both inputs to re-produce an expanded fitting curve that keep the underlying trend of the original data and mitigates the outliers. Here we intend to expand the 36 data samples (months) into 1095 (days). The trial and error principle has been applied to find the best number of RBF centres that can produce the best TLS fitting results. Figure (7.7) shows a comparison between interpolation using direct RBFs (black line) and approximation using TLS (blue dashed line) of the second

input (temperature). Here we have used seventeen RBF centres in order to obtain the best TLS fits.

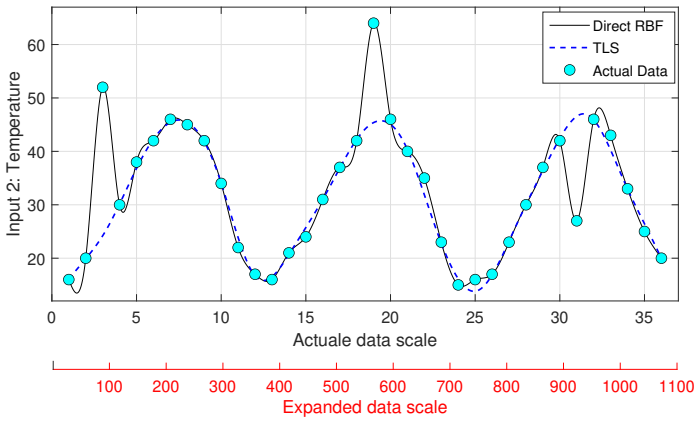


Fig. 7.7 Comparison between TLS (17 Centres) and interpolating RBFs showing fitting performance for the second input (Temperature).

Figure (7.8) compares the use of direct RBF interpolation with the TLS approximation for the third input (number of cars). Here we used five centres for the TLS curve. It is not difficult to see how the TLS outperforms the direct RBFs in mitigating the effect of the outliers. The resultant fitting curves keep to the underlying trend of the original data and ignore the outliers in both data sets, i.e. temperature and the number of cars.

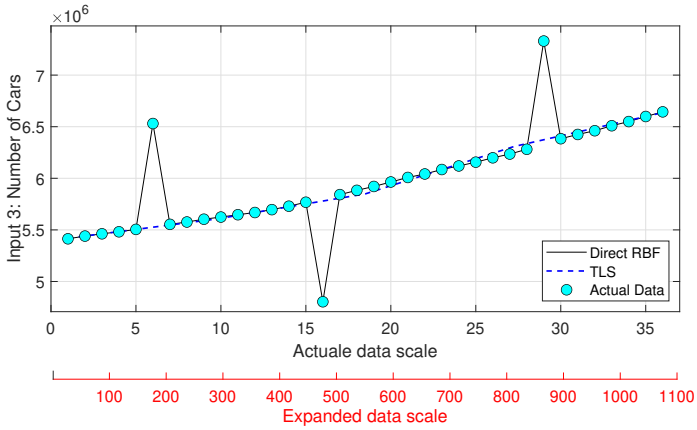
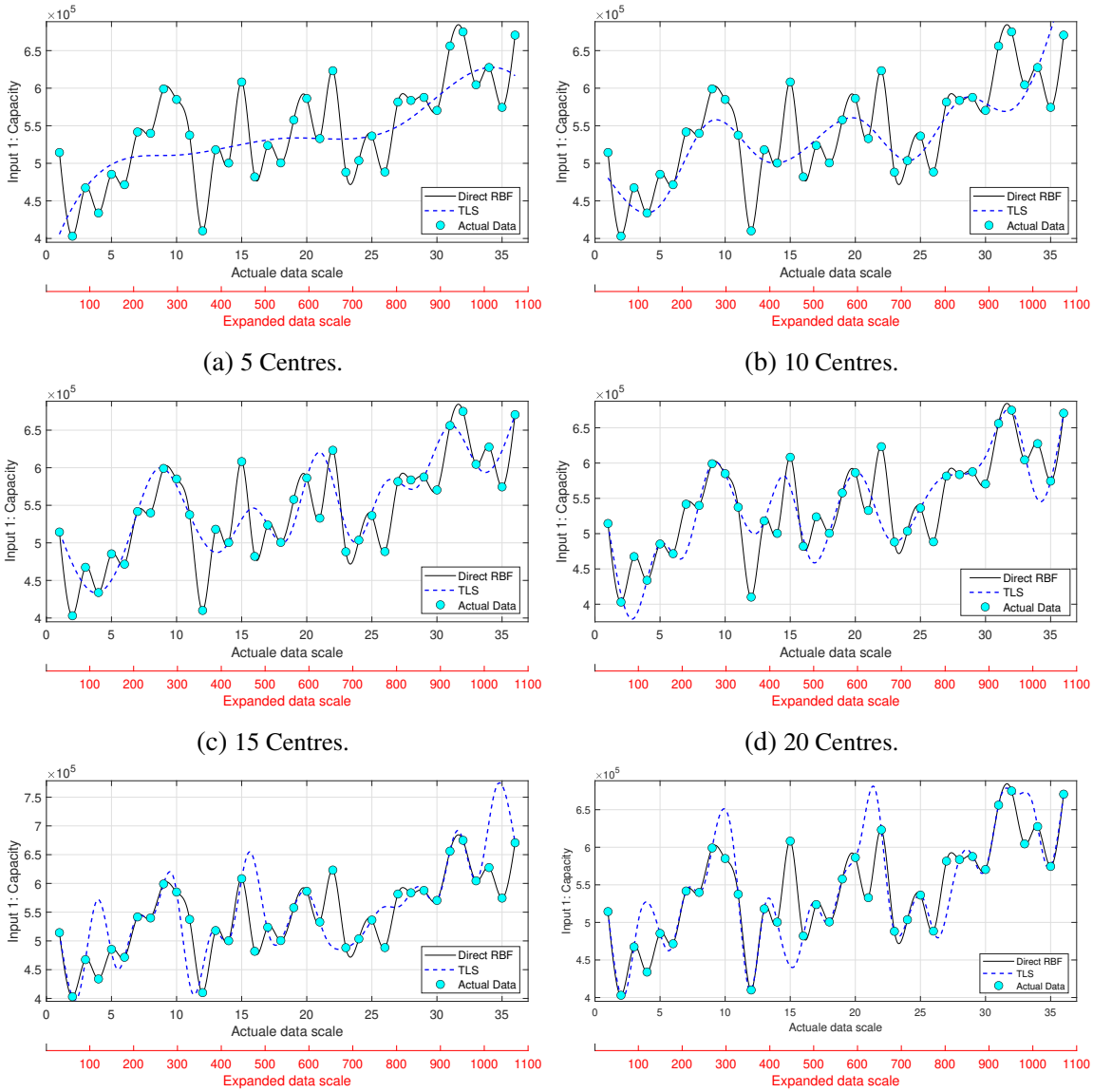


Fig. 7.8 Comparison between TLS (5 Centres) and Direct RBFs fitting performance for the third input (Number of Cars).

- **Random pattern:** when the data is erratic, disorderly, or the curve is not following a smooth trend, then its pattern can be considered as random. In other words, if the data has notable randomness built into it then it will be difficult to explain the underlying

pattern. Accordingly, the data of the third input (capacity) can be considered as random pattern data. Referring to the Figures (7.6a and 7.6b), although the capacity data has been modified to include one outlier, it is very difficult to distinguish this anomalous value from the rest of the data points. That is due to the pattern of the data which almost has built-in randomness and so notably fluctuates. Modelling this type of data using TLS did not provide the desired fit. Even after trying all possible numbers of centres, we could not find a good fit that keept the underlying shape of the original data and ignored the outlier. In Figures (7.9a to 7.9f), we show evidence of the defect in dealing with the random data when using TLS. Six different values for the centres are illustrated in these figures. In all cases, the TLS has failed to provide a good approximation model.



(a) 5 Centres.

(b) 10 Centres.

(c) 15 Centres.

(d) 20 Centres.

(e) 25 Centres.

(f) 30 Centres.

Fig. 7.9 Comparison between SLS (with different numbers of centres) and Direct RBFs fitting performance for the first input (Capacity).

7.3.3 Empirical Results

In this section, we are presenting the results of investigations carried out on the same gasoline prediction problem discussed in the previous chapter. We intend to use the same data supplied, which had four variables. Three inputs. i.e. capacity, temperature, and the number of cars as well as one output, i.e. consumption. Each data set contains 36 data points which represent three years' samples on a monthly basis. As discussed in the previous section, the TLS can not deal with some specific shape of the data. When the data has a smooth distribution, then

outliers present themselves in a way that is easier to detect and mitigate. When the underlying trend in the data is erratic or oscillatory, then outliers may not appear to be any different from the rest of the points. In this case, it would be difficult to gain good performance form the TLS approach, and it will not be the right choice for approximation. This has been adequately explained in an example of using one hypothetical outlier at the capacity data. Therefore, we intend to use the original data (no modification or outliers) for two variables, i.e., the output (consumption) and first input (capacity). In contrast, the data of the second and third variables will be modified in which to have three outliers in each data set, as showed in the previous section. In other words, the performance of TLS as the expansion technique as well as outliers mitigation will be examined using all supplied data sets, which includes outliers inserted into the temperature and number of cars data. Table (7.1) shows the data structure that will be utilized to explore the performance of TLS in improving the ANFIS prediction accuracy.

Table 7.1 Data structure for examining TLS performance in improving the ANFIS prediction accuracy.

Variables	**Title**	**Data**	**Outliers**	**Expansion method**	**RBF Centres**
Output	Consumption	Original	—-	Direct RBF	36
Input 1	Capacity	Original	—-	Direct RBF	36
Input 2	Temperature	Modified	3	TLS-RBF	16
Input 3	Num. of Cars	Modified	3	TLS-RBF	5

It can be noticed that the number of RBF centres for the output (consumption) and the first input (capacity) data sets are equal to the number of original data samples. This is due to the use of the direct RBF technique as the expansion method which uses 36 RBF centres that are equal to the number of data points when interpolating. As explained, this will provide a square system of linear equations. Consequently, the expanded data will pass through all the original data points. This means when the data does not contain outliers then we use the direct RBF as the expansion method. In contrast, when the data contains outliers, such as input two (temperature) and input three (num. of cars), then the TLS is to be utilized. The number of RBF centres is to be reduced to sixteen and five, respectively. These restricted centres provided a better approximation fitting that keeps to the underlying trend of the original data and ignores the outliers. In such a case, both expansion techniques are to be used in solving this type of problem. Therefore, we will use the term **Hybrid Expansion** to express a common solving strategy that includes both expansion methods working simultaneously.

As a test case, we intend to apply our second proposed model, shown in Section (7.2.1), into the ANFIS validation models listed in Table (6.2) in chapter (6). However, we choose to use the structure of one experiment from each ANFIS model rather than use all introduced experiments (see Tables 6.3, 6.5, & 6.7). These three experiments have been selected as they provided the optimum results out of twenty-four experiments (one of eight in each ANFIS model). Note that these experiments were originally solved using only the direct RBF method. Whereas in this chapter, we are intending to expand the data by applying the hybrid RBF approximation into the same ANFIS structure as the selected optimum experiments. In other words, we are using the TLS (where it is needed) combined with the direct RBF to approximate the data. Table (7.2) shows the details of the data scope, and the expansion and scaling methods that will be used for each experiment.

Table 7.2 Structure of selected experiments.

Model	**Exp.**	**Expanded Data**		**Expansion**			**Scaling**
		Scope	Samples	Method	RBF	Coefficients	
ANFIS 1	8	Weekly	156	Hybrid	Mul	$\rho_1 = 0.01, \rho_2 = 0.2$	Z-score
ANFIS 2	12	Weekly	156	Hybrid	Mul	$\rho_1 = 0.02, \rho_2 = 0.2$	Chebyshev
ANFIS 3	20	Daily	1095	Hybrid	Mul	$\rho_1 = 0.01, \rho_2 = 0.2$	Chebyshev

Here we applied the hybrid expansion (i.e. TLS-RBF and Direct-RBF) into experiments eight, twelve, and twenty, rather than use Direct-RBF only. Again, we used the hybrid expansion due to the outliers detected in the data sets of input 2 and 3 only.

In order to validate our proposed model, we intend to solve all three experiments explained in Table (7.2) using two expansion techniques. First, we apply the Direct RBF method to all data sets (including the noisy data) to solve the three ANFIS models, then we solve the same data sets using the hybrid expansion. By applying these two methods into the same data sets we are showing how the TLS has improved the prediction accuracy of ANFIS.

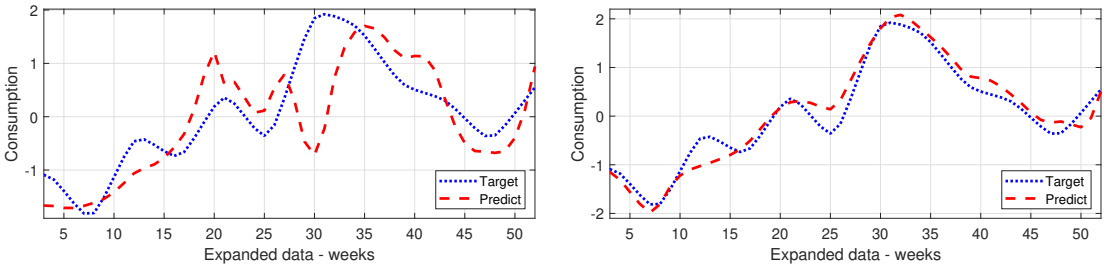
Table (7.3) demonstrates the results of solving the gasoline problem using our second proposed model. As mentioned, the results shown in this table represents the ANFIS outputs of three experiments. Each experiment has been examined using two expansion techniques. It also illustrates the performance improvement of each experiment based on the difference between the results of both expansion methods. Looking at RMSE, NRMSE and R^2 values, the results of all experiments showed that the hybrid RBF has significantly outperformed the direct RBF when the data are noisy.

Table 7.3 Results of solving various experiments using the Direct and Hybrid RBF expansion techniques.

Model	Exp.	Direct RBF Expansion			Hybrid Expansion			Performance
		RMSE	NRMSE	R^2	RMSE	NRMSE	R^2	
ANFIS 1	8	0.7102	0.1901	0.6054	0.2334	0.0625	0.9593	↗ 35%
ANFIS 2	12	0.8837	0.4419	0.2829	0.3483	0.1742	0.7619	↗ 48%
ANFIS 3	20	0.8116	0.4058	0.2243	0.1824	0.0912	0.9461	↗ 72%

Considering the results of experiment eight, it is not difficult to notice that using the hybrid RBF expansion has decreased the RMSE by 0.4768. Also, the NRMSE has been reduced by 0.1276 from 0.1901 using the direct RBF to 0.0625 using the hybrid RBF expansion. Moreover, applying the hybrid RBF improved the prediction performance considerably increasing by nearly 35% compared to the direct RBF. Similarly, experiment twelve shows a significant decrease in both RMSE and NRMSE with values of 0.5354 and 0.2677, respectively. The performance of this experiment increased by nearly 48% when using the hybrid RBF as an expansion method for our proposed model.

Figure (7.10) shows the convergence between the targeted and predicted values of the validation data for experiment eight. Figure (7.10a) represents the results of using the direct RBF as an expansion method. Figure (7.10b) elucidates the results of using the hybrid expansion (i.e. TLS-RBF for inputs 2 & 3, and Direct-RBF for the output and input 1).



(a) Direct RBF expansion.

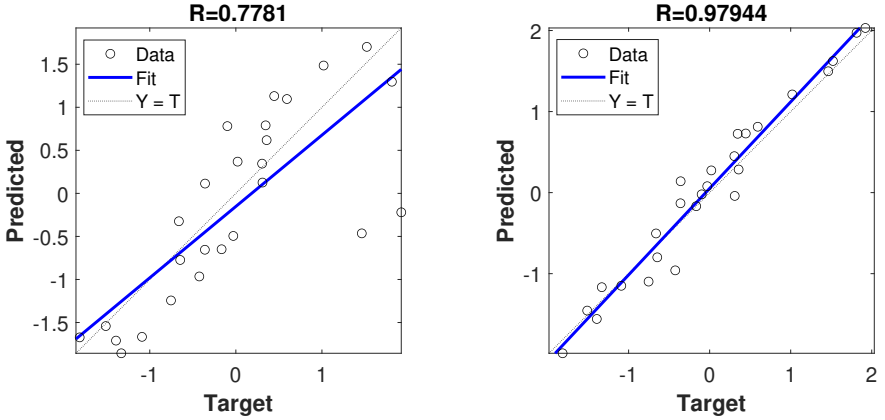
(b) Hybrid RBF expansion.

Fig. 7.10 Targeted and predicted gasoline consumption for validation data of experiment 8: (a) Direct RBF expansion (b) Hybrid RBF expansion.

The graphs clearly showed that substantial improvement could be obtained when using hybrid RBF. Looking at Figure (7.10b), it is not difficult to see how the prediction curve is showing a small deviation from the underlying trend of the targeted data curve. Whereas,

using the direct RBF (Fig. 7.10a) provides an evident deviation between the two curves. The behaviour of the predicted data has failed to converge with the target data in most areas.

Figure (7.11) demonstrates the regression scatter plots of experiment eight using both expansions (direct and hybrid). It shows up to what level the model can fit the data. Figure (7.11a) clearly illustrates how nearly 39.5% of the variation of the data (equal to R = 0.7781) has not been captured by the model when using direct RBF expansion. In contrast, Figure (7.11b) demonstrates how nearly 96% of the variability (equal to R = 0.97944) is explained by the model using the hybrid RBF expansion.



(a) Direct RBF expansion.

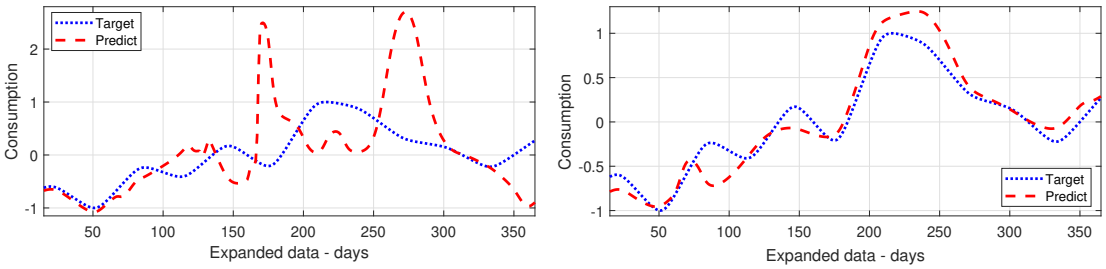
(b) Hybrid RBF expansion.

Fig. 7.11 Regression scatter plot of the targeted and predicted gasoline consumption for validation data of experiment 8: (a) Direct RBF expansion (b) Hybrid RBF expansion.

Experiment twenty showed the highest rate of improvement in performance. Using the direct RBF expansion to solve this model provided a fragile prediction scenario. The error measures represented by the RMSE and NRMSE emphasised very high values of (0.8116) and (0.4058), respectively. Therefore, the performance of this experiment decreased to an unacceptable level to reach 22%. However, using the hybrid RBF expansion has outperformed its rival. Looking at row-wise results of experiment twenty in Table (7.3), it can be noticed that the RMSE and NRMSE shows a significant decrease of 0.6292 and 0.3146, respectively. Consequently, the performance of this experiment has been increased dramatically, by nearly 72% to reach 94.61%.

Figure (7.12) shows a comparison of the convergence between the targeted and predicted values using both expansion methods. Referring to Figure (7.12a), it is not difficult to see how the predicted values showed considerable lack of convergence when using the direct RBF as an expansion method. In other words, the predicted values are in high variation and far from the targeted data. In contrast, Figure (7.12b), shows how well the model can predict

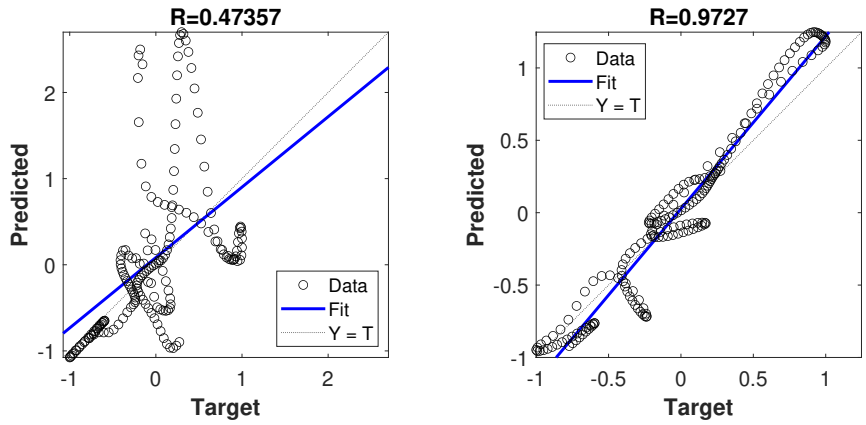
when using the hybrid RBF expansion. The curve of the predicted values is very close to the target values with an evident rate of convergence.



(a) Direct RBF expansion. (b) Hybrid RBF expansion.

Fig. 7.12 Targeted and predicted gasoline consumption for validation data of experiment 20: (a) Direct RBF expansion (b) Hybrid RBF expansion.

Figure (7.13) confirms the capability of the model to fit the data. The plots of the regression scatter shows how well the model captures the variation of the data. Figure (7.13a) represents the regression scatter plot of the proposed model using the direct RBF expansion. We can notice that nearly 22% of the variability (equal to R = 0.47357) is explained by the model. This can be considered as a meager value of the coefficient of determination R^2. Whereas, nearly 95% of the variability (equal to R = 0.9727) is explained by the model using the hybrid RBF expansion (Fig. 7.13b).



(a) Direct RBF expansion. (b) Hybrid RBF expansion.

Fig. 7.13 Regression scatter plot of the targeted and predicted gasoline consumption for validation data of experiment 20: (a) Direct RBF expansion (b) Hybrid RBF expansion.

7.3.4 Model Validation

This section illustrate alternative data sets that have been utilized to validate the TLS approach. A case study formed from a real-life problem of natural gas consumption forecasting presented by [17]. The data used in this case study have been taken directly from their work. The authors designed their ANFIS model to include some input variables which includes Gas Price, GDP, Inflation Rate, Population, Unemployment Rate, and one output representing Gas consumption.

They collected 36 samples of actual data per input. We used this data to evaluate the effectiveness of the TLS approach. Analysing the original data showed that two data sets contain outliers. The second input variable (GDP) contains 3 outliers, and the forth input variable (Population) contain one outlier. We employed the TLS approach to mitigate the existing outliers when generating the expanded data. The results showed an outstanding improvement of almost 20% of the prediction accuracy when compared to the author's original results using the standard ANFIS model.

7.3.5 Discussions and Conclusions

This chapter illustrated a mathematical solution method to the outliers problem. We have developed a robust outliers mitigation model that can help in accurately modelling data containing noise. This model has been shown to provide a better fit for ANFIS optimization. We have shown the difference between applying the RBFs directly as an interpolating technique to expand the data, and the use of SLS and TLS as approximation expansion techniques. The RBFs (Linear, Cubic, and Multiquadric) have been used as the basis functions for all mentioned techniques. The real-life problem of predicting the consumption of petroleum products introduced in Chapter (6) has been utilized as a test case for the second proposed model in this chapter. As explained, the gasoline problem comes with four variables, i.e. three inputs and one output. The data sets of the output and the first input variables were identical to the original data provided in Chapter (6). In comparison, the data sets of the second and third input variables were modified to include some outliers in order to investigate the robustness of the proposed model. We decided to select one experiment from each ANFIS validation model introduced in Chapter (6). A total of three experiments with three different structures have been carried out and solved using two expansion techniques. Firstly, we used the Direct RBF expansion to interpolate all data sets (including the ones containing outliers). Secondly, we used a Hybrid RBF expansion, i.e. using Direct RBF interpolating when there are no outliers and TLS-RBF approximation when there were outliers. By reviewing all the discussion and results showed in this chapter, we can conclude the following:

1. When the data have noise or outliers, interpolation is not the right choice for modelling. In fact, applying the direct RBF to the noisy data showed an abysmal prediction accuracy.

2. Standard least-squares (SLS) approximation can perform better than the interpolation approach when the data contains outliers. However, the SLS fitted curves were notably influenced by the outliers.

3. Transformed Least-Squares (TLS) showed an outstanding performance in mitigating outliers when approximating a smoothness trend (linearly or non linearly) data set.

4. When the data is erratic, fluctuating in its nature, the TLS has failed to provide the desired fit in modelling this type of data.

5. In all experiments, the results showed that the hybrid RBF has significantly outperformed the direct RBF when the data are noisy.

6. When care is taken, expanding the data using a more sophisticated model such as TLS, the model can perform even better with an order of magnitude difference in its error measures and prediction accuracy.

Chapter 8

ANFIS Optimization as a Fuzzy Expert System: Proposed Model 3

8.1 Introduction

The Encyclopedia Britannica defines an Expert System as:

"A computer program that uses artificial-intelligence methods to solve problems within a specialized domain that ordinarily requires human expertise."

This chapter introduces a tool, developed by the author, for optimizing ANFIS. Based on the definition above, it is sensible to define this as a Fuzzy Expert System (FES). It is clear from the definition above that an expert system can be thought of as a computer system that can simulate the human expert's capability of decision-making. As a division of Artificial Intelligence (AI), expert systems are utilized to solve complex problems by applying inference and knowledge in addition to traditional coding procedures. In other words, expert systems represent smart programs that use knowledge and reasoning to deal with characterised problems that usually depend on human expertise for their solutions. Accordingly, expert systems can consist of two subsystems, the Knowledge Base (KB) and the inference engine. The rules and facts are to be represented by the knowledge base, while the inference engine infers new facts by applying the rules into the existing facts [58] [71].

In contrast, when the problem under consideration contains uncertainty or subjectivity, then the fuzzy logic is to be employed as an inference mechanism rather the Boolean logic. In other words, the knowledge base of the system will be obtained depending on the human operator's knowledge which can be recorded in two forms, linguistic information and numerical information. Therefore, it can be named as fuzzy knowledge base. Before the emanation of neuro-fuzzy techniques, only the linguistic information was used in building the fuzzy systems in most problems. Later on, with learning algorithms, the nero-fuzzy approaches (such as ANFIS) has appeared. Thus, the numerical information represented by the historical data (input-output data pairs) is to be used to refine the MFs systematically. Mimicking a human expert using linguistic information combined with numerical data to fine-tune the membership functions will result in the Fuzzy Expert System (FES). Thus, the general form of the expert system become more specific as a fuzzy expert system [108] [134] [135].

ANFIS models represent one of the effective models of fuzzy expert systems. It has two main input categories, i.e. the numerical data sets and the fuzzy knowledge base. The numerical data sets represent the historical data that can be collected from different sources. While the fuzzy knowledge base (MF's type and number) is to be obtained based on the knowledge often provided by human experts.

This means, ANFIS models are to be developed intuitively by human intervention. Nonetheless, the knowledge base proposed by experts does not necessarily represents the only one that can provide the best solution. For sure, there are several other combinations of MFs (types and numbers) that may provide a better prediction performance of ANFIS. Therefore, in this work, we aimed to proposed an Adaptive Neuro-Fuzzy Inference Expert System (ANFIES) that can look at all the possible scenarios of the fuzzy knowledge bases. This will give us the ability to find the best fuzzy knowledge base (MFs types and number combinations) of the ANFIS inputs. Consequently, we gain the optimal prediction performance. However, in this chapter, we are presenting a tool of solving ANFIS as a fuzzy expert system rather than discussing the expert system approaches.

8.2 Development of Model 3

8.2.1 Proposed Fuzzy Expert System Model

In the previous two chapters (6 & 7) we introduced two proposed models. Each model contains two main parts in its structure, i.e. the pre-processing (re-sampling) model at layer 0 and ANFIS procedures in layers (1 to 5) (see Figures 6.1 & 7.3). We have developed a data expansion model and outlier mitigation model, respectively, as data optimization models. These two models have been applied into three unique fuzzy knowledge base (MFs type and number showed in Table 6.2) that been specified prior to the run of ANFIS. In this chapter, we present a tool that complements what was previously achieved. We are proposing a model that can obtain the optimal solution by investigating all the possible combinations of the fuzzy knowledge base structures at layer 1, rather than using one specific structure.

In this section, we are presenting the development of the third proposed model. In MATLAB, there are eleven membership functions that can be utilized for each input variable as part of the ANFIS solving procedure (see Section 3.4.2). However, we intend to employ five diversified MFs types out of the eleven MFs to examine the ANFIES. These MFs are the Triangular (trimf), Trapezoidal (trabmf), Gaussian (gaussmf), Generalized Bell (gbellmf), and Difference Sigmoidal (dsigmf).

Assume that we have an ANFIS model that contains (n) input variables and one output. There are five types of MFs that can be assigned to each input variable (one at a time). Each MF can be represented by 2, 3 or 4 intervals (number of MFs) for every single input variable. Using all these MFs (types and number) will provide the total number of fuzzy knowledge base combinations, i.e. (TKB).

These combinations represent all possible fuzzy knowledge bases that can be used to solve ANFIS models. Looking at Figure (8.1) which illustrates all the possible scenarios of combinations that can be used as a KB for ANFIS. The total number of knowledge bases can be created as follows:

$$TKB = K_1 \times K_2, \tag{8.1}$$

where K_1 represents the MF's types combinations, and K_2 is the MF's number combinations. K_1 can be obtained by multiplying the total number of MF's types (TMF_{types}) used for each input variable. It can be extracted using the following formula:

$$K_1 = \prod_{i=1}^{n} TMF_{types_i} \tag{8.2}$$

K_2 can be calculated by multiplying the total number of MFs intervals (TMF_{num}) used for each MF in each input variable. The following formula can be used to find K_2:

$$K_2 = \prod_{i=1}^{n} TMF_{num_i} \tag{8.3}$$

In this chapter, all equations and derived expressions have been developed by the author. Moreover, the application tool presented in this chapter has been entirely created from scratch using the MATLAB app designer.

For example, assume that we have an ANFIS model that contains three input variables, and we are using two types of MFs (i,e. trimf and trapmf) for the first and second input variables, respectively. While the third input variable will use three types of MFs (i,e. trimf, trapmf and gaussmf). Thus, the total number of used types of MFs will be as follows:

$$K_1 = \prod_{i=1}^{3} TMF_{types_i} = 2 \times 2 \times 3 = 12$$

This will provide a (12×3) MF_{type} matrix, as highlighted in Table (8.1). In contrast, each type of MF in each input variable can be represented by two options (i.e. either 2 or 3 MFs number for each type in each input variable). For instance, if we choose the (trimf) as the type of MF for the first input variable, then it can be either (2 trimfs) or (3 trimfs). In order to find the total combination of MFs number, we use the following:

$$K_2 = \prod_{i=1}^{3} TMF_{num_i} = 2 \times 2 \times 2 = 8$$

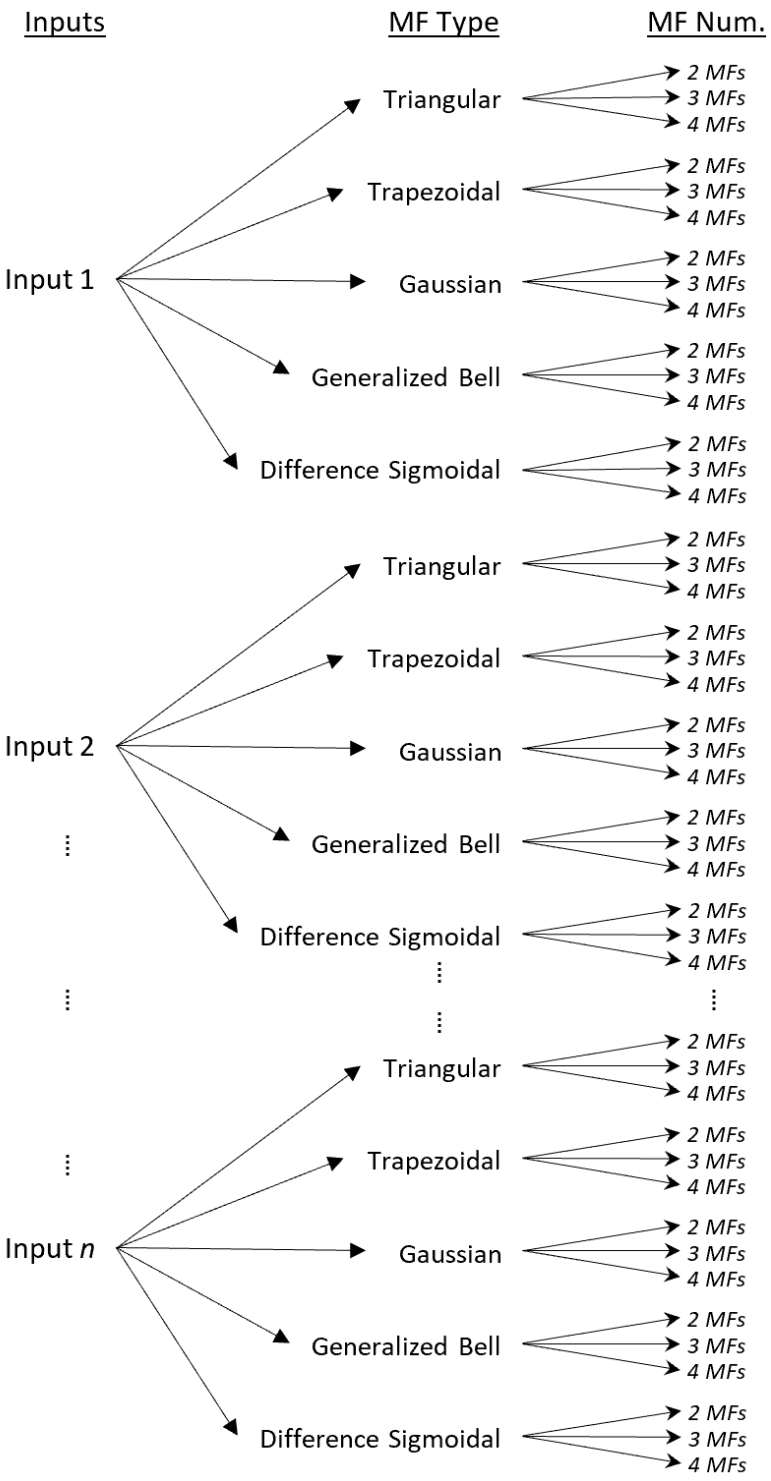


Fig. 8.1 KB combinations of proposed expert ANFIS.

This will provide a (3×8) MF_{num} matrix, as highlighted in Table (8.1). The total fuzzy knowledge bases (TKB) that will result from the combinations of MF_{type} and MF_{num} can be

calculated as follows:

$$TKB = K_1 \times K_2 = 12 \times 8 = 96$$

Table (8.1) shows all the fuzzy knowledge base combinations of the previous example. Each KB_j listed in this table (where $j = 1$ to 96) is composed of two sets of information. Thus, a set of types of MFs (located at the MF_{type} matrix) is to be combined with a corresponding set of number of MF (located at the MF_{num} matrix). For instance, the structure of KB_{54} (highlighted) results from the intersection of the seventh combination of the MF_{type} matrix and the sixth combination of the MF_{num} matrix. This means when we are at KB_{54} the fuzzy knowledge base will compose of "3 trapmf" for input 1, "2 trimf" for input 2, and "3 trimf" for input 3. And so forth.

These fuzzy knowledge bases are to be employed by the ANFIES to represent the main structure of layer (1) at all the ANFIS models. In other words, each knowledge base is unique, and represents one ANFIS model out of 96 models. All these models are to be solved and compared to each other via its results in order to find the best ANFIS that outperforms all other models.

Table 8.1 Example of KB combination.

				MF_{num} Matrix							
				Comb.	1	2	3	4	5	6	7	8
				Input1	2 MFs	2 MFs	2 MFs	2 MFs	3 MFs	3 MFs	3 MFs	3 MFs
				Input2	2 MFs	2 MFs	3 MFs	3 MFs	2 MFs	2 MFs	3 MFs	3 MFs
	MF_{type} Matrix			Input3	2 MFs	3 MFs	2 MFs	3 MFs	2 MFs	3 MFs	2 MFs	3 MFs
Comb.	Input 1	Input 2	Input 3									
1	trimf	trimf	trimf		KB_1	KB_2	KB_3	KB_4	KB_5	KB_6	KB_7	KB_8
2	trimf	trimf	trapmf		KB_9	KB_{10}	KB_{11}	KB_{12}	KB_{13}	KB_{14}	KB_{15}	KB_{16}
3	trimf	trimf	gaussmf		KB_{17}	KB_{18}	KB_{19}	KB_{20}	KB_{21}	KB_{22}	KB_{23}	KB_{24}
4	trimf	trapmf	trimf		KB_{25}	KB_{26}	KB_{27}	KB_{28}	KB_{29}	KB_{30}	KB_{31}	KB_{32}
5	trimf	trapmf	trapmf		KB_{33}	KB_{34}	KB_{35}	KB_{36}	KB_{37}	KB_{38}	KB_{39}	KB_{40}
6	trimf	trapmf	gaussmf		KB_{41}	KB_{42}	KB_{43}	KB_{44}	KB_{45}	KB_{46}	KB_{47}	KB_{48}
7	trapmf	trimf	trimf		KB_{49}	KB_{50}	KB_{51}	KB_{52}	KB_{53}	KB_{54}	KB_{55}	KB_{56}
8	trapmf	trimf	trapmf		KB_{57}	KB_{58}	KB_{59}	KB_{60}	KB_{61}	KB_{62}	KB_{63}	KB_{64}
9	trapmf	trimf	gaussmf		KB_{65}	KB_{66}	KB_{67}	KB_{68}	KB_{69}	KB_{70}	KB_{71}	KB_{72}
10	trapmf	trapmf	trimf		KB_{73}	KB_{74}	KB_{75}	KB_{76}	KB_{77}	KB_{78}	KB_{79}	KB_{80}
11	trapmf	trapmf	trapmf		KB_{81}	KB_{82}	KB_{83}	KB_{84}	KB_{85}	KB_{86}	KB_{87}	KB_{88}
12	trapmf	trapmf	gaussmf		KB_{89}	KB_{90}	KB_{91}	KB_{92}	KB_{93}	KB_{94}	KB_{95}	KB_{96}

8.2.2 Development of a Tool for Optimizing the Proposed Model

A tool for optimization has been programmed using MATLAB in order to get better performance. We designed and developed a specific application that gives the user an intuitive input-output environment. Figure (8.2) illustrates the coding flowchart of the proposed fuzzy expert system. It shows the sequence of all the interactive procedures for the developed tool. These procedures include all the inputs, interactive selection options, solving process, and outputs. Thus,

- **Inputs:** represented by:
  - Input-output data sets.
  - Training- testing data splitting (numbers or percentages).
  - Number of target extended data.
  - Number of RBF centres when using TLS.
  - Modified Multiquadric RBF curvature parameters.
  - Number of designated epochs.
- **Interactive selection options:** represented by:
  - Selection of the data range (Actual, scaled and expanded).
  - Selection of scaling method (Chebyshev or Zero-mean).
  - Selection of expansion method ((Direct RBF or TLS).
  - Selection of the basis function (Linear RBF, Cubic RBF, or Modified Multiquadric RBF).
  - Selection of MF's types and number of each input.
- **Solving process** represented by:
  - Expansion process.
  - Scaling process.
  - Generating MF's types matrices.
  - Generating MF's number matrices.
  - Generating fuzzy knowledge base matrix.
  - Performing ANFIS solving algorithms.
  - Finding optimal solution.

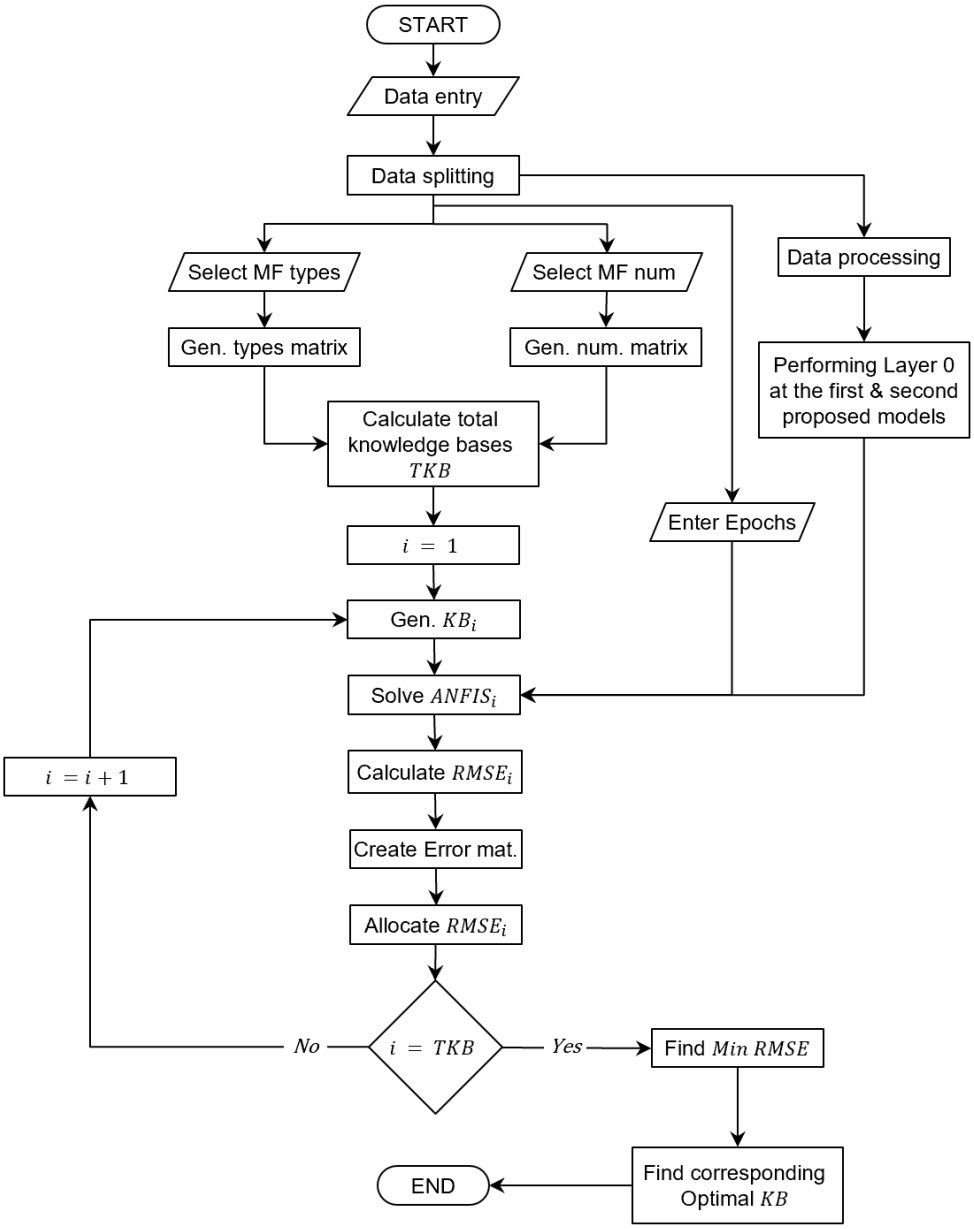


Fig. 8.2 Flowchart of the proposed expert ANFIS.

- **Outputs:** represented by:
  - Total number of tested models.
  - Optimal ANFIS model (best fuzzy knowledge base).
  - Optimal epoch number.
  - Minimum RMSE and normalized RMSE for the training and testing fitting data.

- Plotting of the original data, expanded data (Direct RBF and TLS), predicted data, initial and trained membership functions, errors, regressions, 3D surfaces, and histograms.

The proposed tool of the expert ANFIS has been developed by using the App Designer in MATLAB. It provides a friendly user interface application which makes dealing with the expert system easier and convenient. The application has been built with an intelligent interactive environment which allows it to change the interface components automatically according to the number of inputs extracted from the entered data sets. Figure (8.3) shows a screenshot of the designed application. In order to show all components with its full options, we have chosen data set samples containing six inputs and one output. Therefore, it can be noticed that six input panels, one for each input, appears in Figure (8.3). The results of solving the model will appear in the same interface of the application. It will show the evaluation error criteria and the fitting plot for both training and testing data sets.

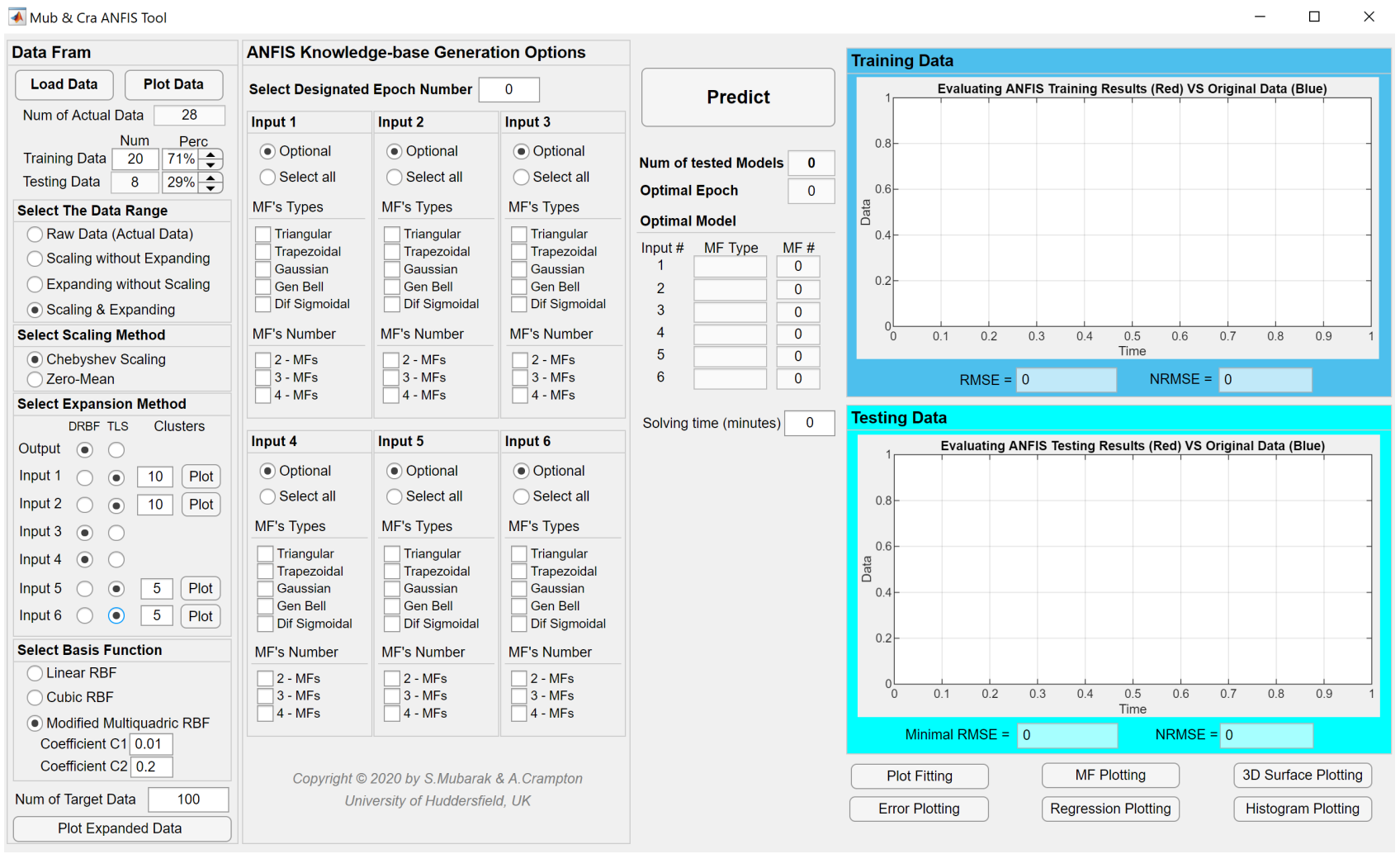


Fig. 8.3 Interface screen shot of the designed optimization tool.

8.3 Implementation of Model 3

8.3.1 Expert System Solving Procedures

In this section, we intend to validate our proposed ANFIES using the gasoline prediction problem introduced in Chapter (6). The provided data sets showed in Section (6.3.3) will be used as the input-output data for the ANFIES. These data represent three input variables and one output variable (see Section 6.3.2). These data sets included thirty-six data samples (i.e. the monthly recorded values of three years) with no outliers. In other words, the number of data samples is small and will be less than the number of total parameters (premise and consequent) in all cases of ANFIS's knowledge base. Therefore, the DRBF (linear, Cubic, and Modified Multiquadriq) will be used to expand the data. The expanded data are to be in the forms of weekly (156 samples) or daily (1095 samples) to replace the monthly samples. These expansion methods are to be combined with two scaling methods (i.e. Chebyshev and Zero-mean) to solve ANFIES. Combining all these methods represents the pre-processing (re-sampling) model (layer 0 at Fig. 7.3). Using all the possible combinations of the previous methods to process the data sets into dual time frames (i.e. weekly and daily) will produce twelve data structures. Each data structure will be solved individually using our proposed ANFIES.

As mentioned in the previous section, the fuzzy knowledge base matrix is to be built using either two, three or four MFs (3 numbers) for each MF type out of (5 types). These MFs types and numbers are to be used for each input variable individually. The maximum number of MFs types combinations can be extracted using Equation (8.2) as follows:

$$K_1 = \prod_{i=1}^{3} TMF_{types_i} = 5 \times 5 \times 5 = 125.$$

This will produce a (125×3) MF_{type} matrix in the case when the model uses all five MFs in each input. In contrast, The maximum possible MF_{num} combinations can be calculated using Equation (8.3) as follows:

$$K_2 = \prod_{i=1}^{3} TMF_{num_i} = 3 \times 3 \times 3 = 27.$$

This will produce a (3×27) MF_{num} matrix. Consequently, the fuzzy knowledge base will be in the form of (125×27) matrix, and the total number of knowledge bases can be calculated

using Equation (8.1) as follows:

$$TKB = K_1 \times K_2 = 125 \times 27 = 3375.$$

This represents the maximum number of fuzzy knowledge bases that can be extracted if all MF_{type} and MF_{num} are applied to all inputs. These fuzzy knowledge bases are to be used to solve the proposed ANFIES for each data structure.

8.3.2 Empirical Results

In this section, we are presenting the results of solving the gasoline prediction problem using the third proposed ANFIES model. As mentioned, this problem has three inputs, i.e. capacity, temperature, and the number of cars as well as one output (consumption). We aim to use the proposed ANFIES model to find the best fuzzy knowledge base among all the possible knowledge bases. In other words, we intend to find the optimal ANFIS model, which provides the minimum prediction errors. The ANFIES will be applied to all twelve data structures resulting from combining the scaling and expansion methods. This will allow us to validate the robustness of our model by applying it to various forms of data sets.

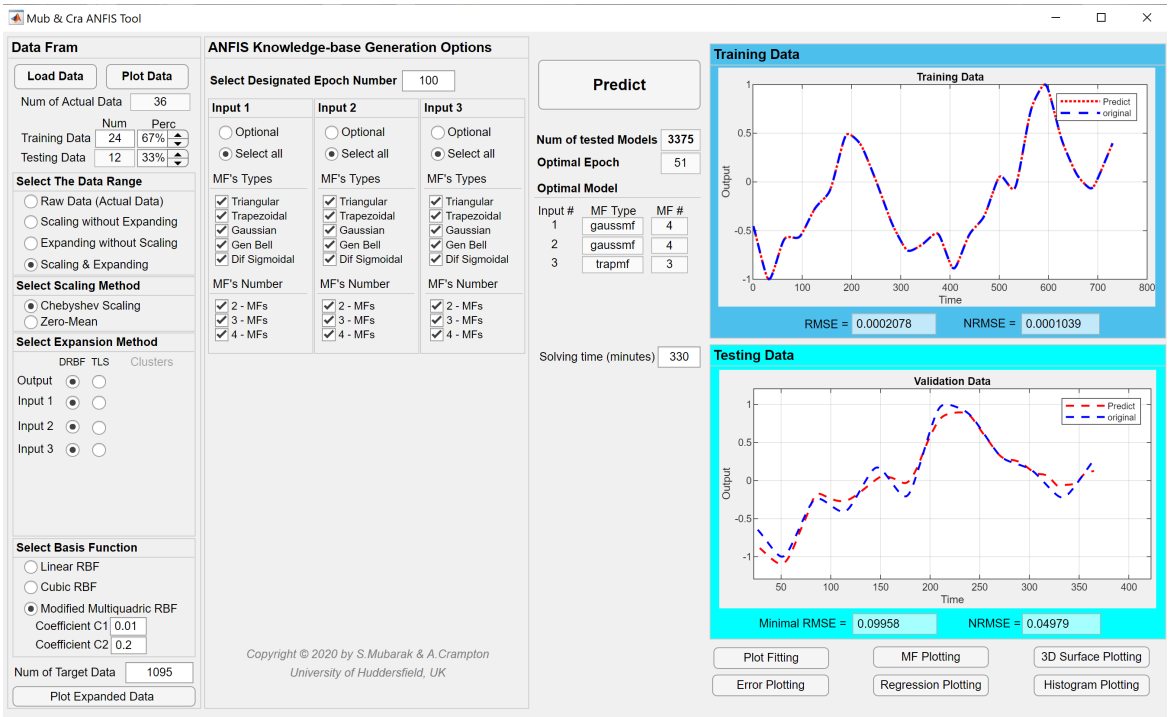


Fig. 8.4 Screen shot for the optimization tool after being run.

By way of explanation, assume that we are solving one of the data structures mentioned in the previous section using our proposed application. This will allow us to select various combinations of MFs types and numbers (usually selecting all). Thus, the application will extract the optimal solution by comparing the results of different ANFISs. Figure (8.4) shows a screenshot of the created application. It can be noticed how the application's interface has shown the tick-box options for three inputs only. This is due to the interactive interface modifying the panel's components automatically according to the input data sets. In contrast, the data structure shown in Figure (8.4) combines the Chebyshev scaling and the modified Multiquadric expansion with ($\rho_1 = 0.01$ & $\rho_1 = 0.2$). The data are to be expanded into daily samples (1095). The maximum number of epochs was chosen to be up to 100 epochs. The results of solving this ANFIES using the previous options took 330 minutes to obtain all possible models and achieves the optimal ANFIS.

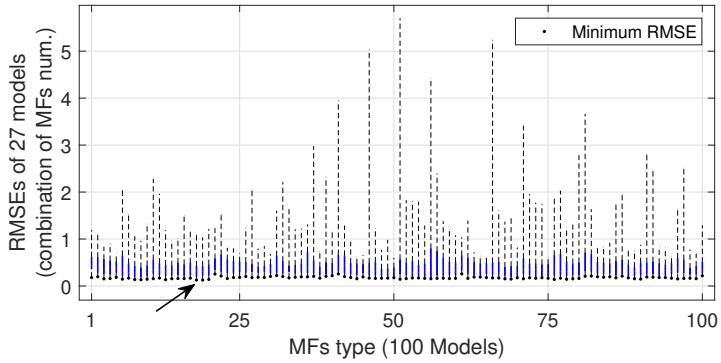
Similarly, we intend to apply our proposed ANFIES to all twelve data structure mentioned. Table (8.2) illustrates the results of solving all data structures using the direct RBF as an expansion method. All the runs were using 100 epochs. The first six rows show the results of the weekly expanded data followed by the daily data. As discussed, we are testing a number of different data sets (data structures). That means each data structure will have its own properties and characteristics. Therefore, we will not follow the comparison strategy between the solved models when analysing the overall results.

It is not difficult to see that the proposed ANFIES has provided impressive results in all cases. Looking at the NRMSE, we can find that all error values were less than 0.08. In fact, in some cases, the NRMSE reached 0.05 or even less. In contrast, the coefficient of determination R^2 values showed how well the model could capture the variation in the underlying data. Again, in all cases, the results of R^2 indicates superior fitting performance. The performance values ranged from 84% to 97%. Nearly 50% of the tested models showed a performance value of more than 94%. Overall, a total of (37,800) models have been tested in order to find the optimal ANFIS model for different scenarios of data scaling and expansion of the gasoline problem.

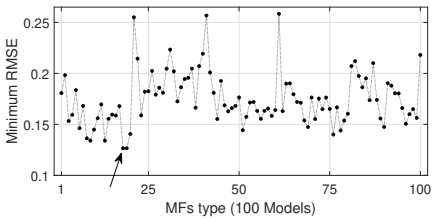
Figures (8.5 to 8.16) shows the plots of all cases. Each figure contains two sub-figures. From the left, we have a box plot for all elements of the RMSE matrix for each model. Then we have the plot of the minimum RMSEs of all MF types combinations on the right. Here we used a box plot to show the range of errors that accompanies each model. The y-axis represents the 27 values of RMSEs corresponding to each MF types combination in the x-axis. From the results of each case, we can indicate the minimum RMSE of each model. These RMSEs were extracted and plotted as the second plot in each figure.

Table 8.2 Expert ANFIS results.

Data Structure							KB of the Optimal Models							
	Data		Expansion			Tested	Input 1		Input 2		Input 3			
Set	Scope	Samples	Method	RBF	Scale	KBs	Type	Num.	Type	Num.	Type	Num.	NRMSE	R^2
1	Weekly	156	Direct	Lin	Ch	2,700	trimf	3	dsigmf	4	gaussmf	3	0.0633	94.47%
2	Weekly	156	Direct	Cub	Ch	2,700	trimf	3	dsigmf	4	dsigmf	3	0.0723	91.99%
3	Weekly	156	Direct	Mul	Ch	2,700	trimf	4	dsigmf	3	gbellmf	3	0.0506	97.06%
4	Weekly	156	Direct	Lin	Z-s	3,375	gbellmf	4	gaussmf	4	trimf	3	0.0688	89.26%
5	Weekly	156	Direct	Cub	Z-s	3,375	trimf	3	dsigmf	4	trimf	3	0.0739	91.03%
6	Weekly	156	Direct	Mul	Z-s	3,375	gaussmf	3	gbellmf	4	gaussmf	2	0.0580	94.98%
7	Daily	1095	Direct	Lin	Ch	3,375	dsigmf	2	gbellmf	4	gaussmf	4	0.0549	94.18%
8	Daily	1095	Direct	Cub	Ch	2,700	trimf	4	dsigmf	4	trapmf	3	0.0680	92.56%
9	Daily	1095	Direct	Mul	Ch	3,375	gaussmf	4	gaussmf	4	trapmf	3	0.0498	96.04%
10	Daily	1095	Direct	Lin	Z-s	3,375	gbellmf	3	gaussmf	3	trapmf	3	0.0719	84.83%
11	Daily	1095	Direct	Cub	Z-s	3,375	gbellmf	2	gbellmf	4	trapmf	3	0.0712	92.01%
12	Daily	1095	Direct	Mul	Z-s	3,375	gaussmf	3	gaussmf	4	trimf	3	0.0570	96.07%

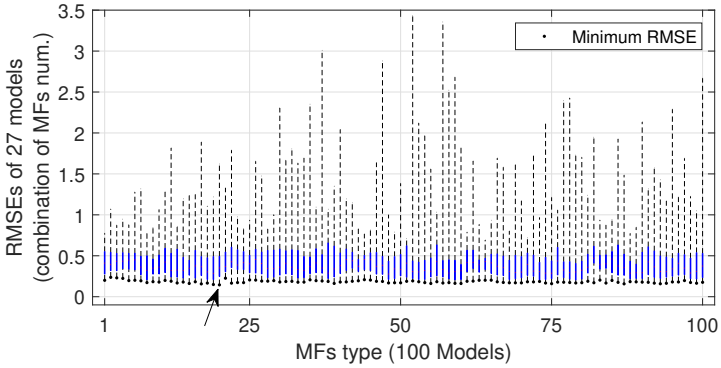


(a) Box-plot of the results of 2700 ANFIS models.

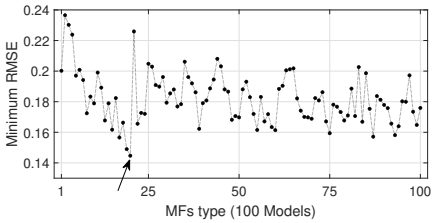


(b) Minimum RMSEs.

Fig. 8.5 Expert ANFIS outputs of the data structure number (1) in Table (8.2).

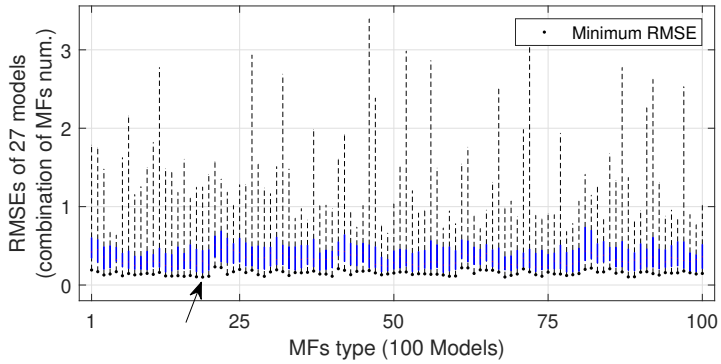


(a) Box-plot of the results of 2700 ANFIS models.

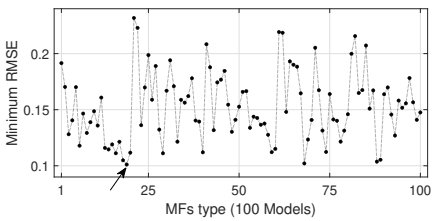


(b) Minimum RMSEs.

Fig. 8.6 Expert ANFIS outputs of the data structure number (2) in Table (8.2).

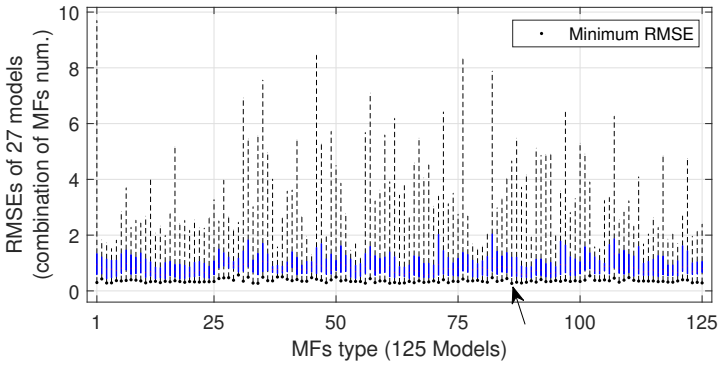


(a) Box-plot of the results of 2700 ANFIS models.

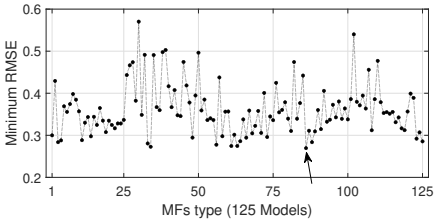


(b) Minimum RMSEs.

Fig. 8.7 Expert ANFIS outputs of the data structure number (3) in Table (8.2).

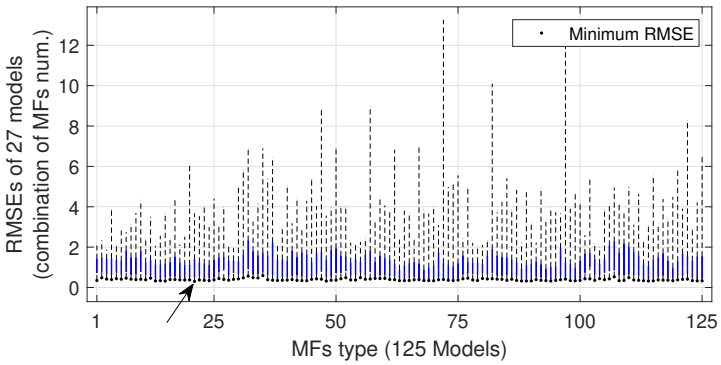


(a) Box-plot of the results of 3375 ANFIS models.

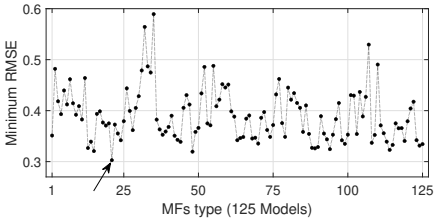


(b) Minimum RMSEs.

Fig. 8.8 Expert ANFIS outputs of the data structure number (4) in Table (8.2).

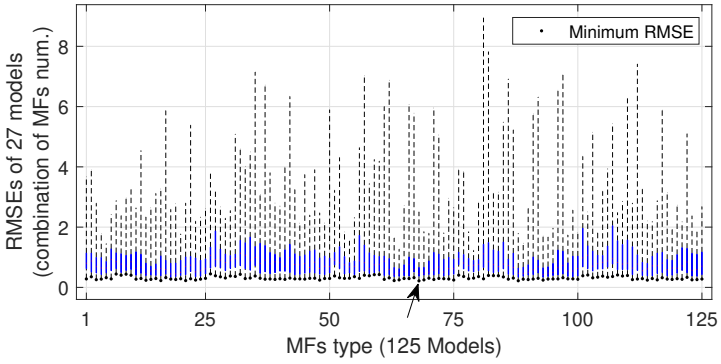


(a) Box-plot of the results of 3375 ANFIS models.

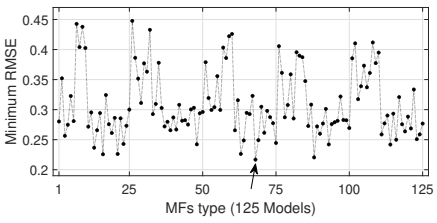


(b) Minimum RMSEs.

Fig. 8.9 Expert ANFIS outputs of the data structure number (5) in Table (8.2).

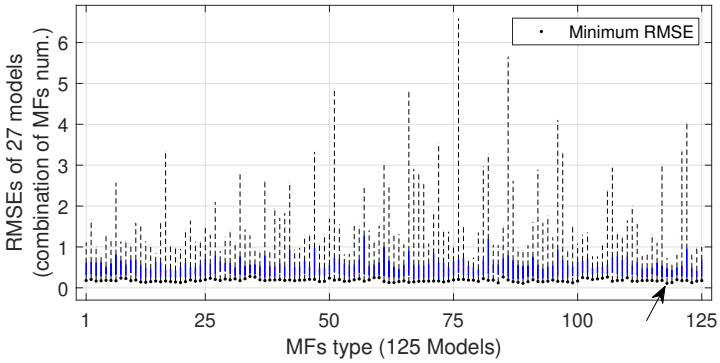


(a) Box-plot of the results of 3375 ANFIS models.

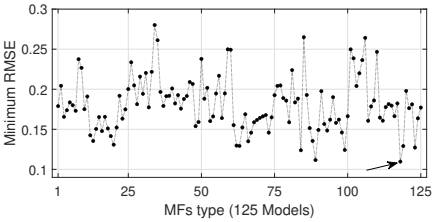


(b) Minimum RMSEs.

Fig. 8.10 Expert ANFIS outputs of the data structure number (6) in Table (8.2).

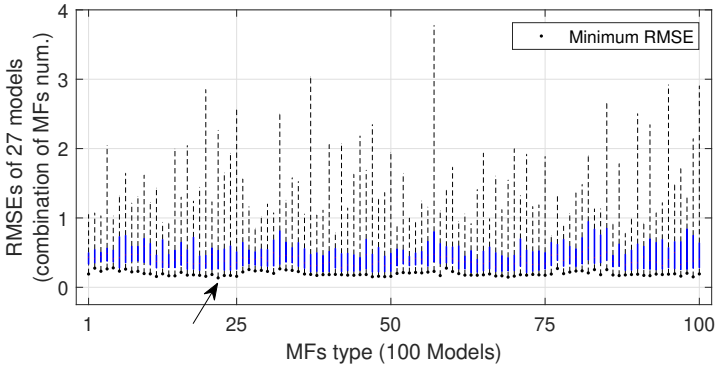


(a) Box-plot of the results of 3375 ANFIS models.

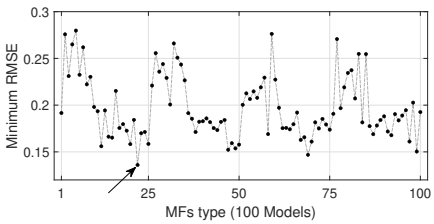


(b) Minimum RMSEs.

Fig. 8.11 Expert ANFIS outputs of the data structure number (7) in Table (8.2).

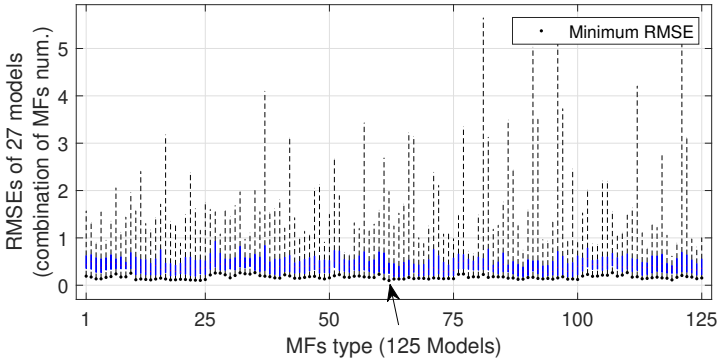


(a) Box-plot of the results of 2700 ANFIS models.

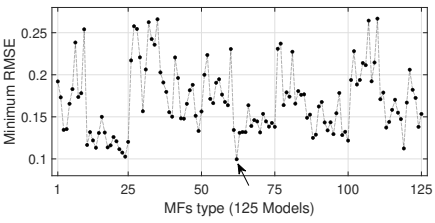


(b) Minimum RMSEs.

Fig. 8.12 Expert ANFIS outputs of the data structure number (8) in Table (8.2).

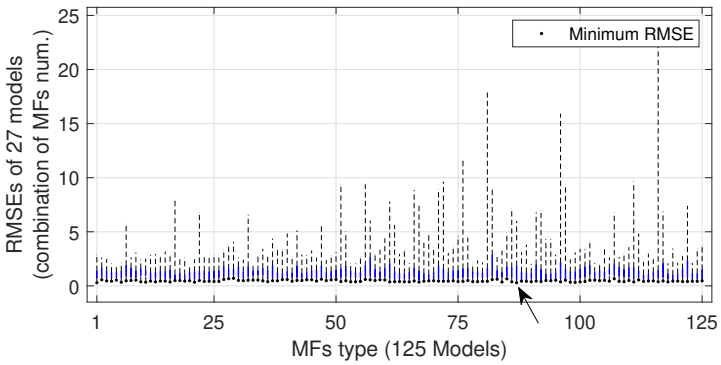


(a) Box-plot of the results of 3375 ANFIS models.

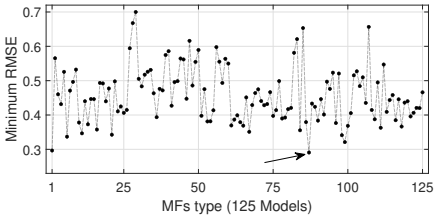


(b) Minimum RMSEs.

Fig. 8.13 Expert ANFIS outputs of the data structure number (9) in Table (8.2).

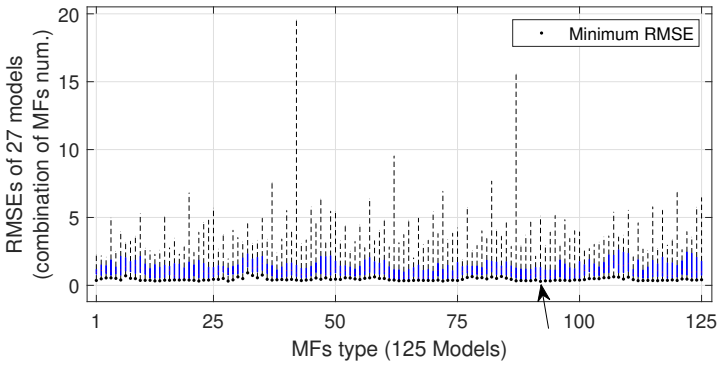


(a) Box-plot of the results of 3375 ANFIS models.

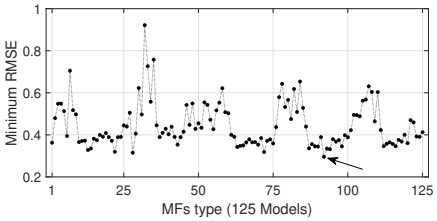


(b) Minimum RMSEs.

Fig. 8.14 Expert ANFIS outputs of the data structure number (10) in Table (8.2).

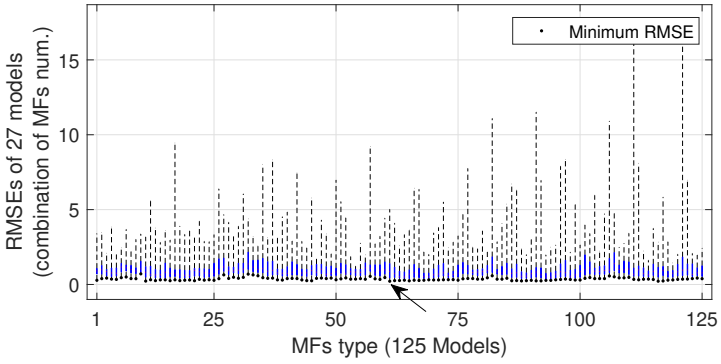


(a) Box-plot of the results of 3375 ANFIS models.

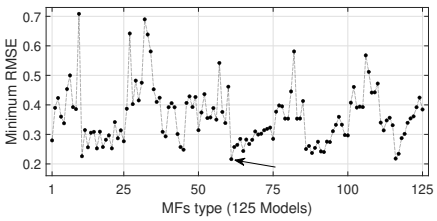


(b) Minimum RMSEs.

Fig. 8.15 Expert ANFIS outputs of the data structure number (11) in Table (8.2).



(a) Box-plot of the results of 3375 ANFIS models.



(b) Minimum RMSEs.

Fig. 8.16 Expert ANFIS outputs of the data structure number (12) in Table (8.2).

8.3.3 Discussions and Conclusions

This chapter illustrated the development process of the proposed expert ANFIS model and the MATLAB application tool that been constricted to optimize the prediction performance of ANFIS models. We proposed an ANFIES as a fuzzy expert system to solve causal forecasting problems. The proposed model works by investigating all available fuzzy knowledge base combinations in order to find the best model and optimal solution. In other words, we introduced a new mechanism that examines all possible MFs combinations (types and numbers) for known data sets rather than using one specific MF's structure. This can help the experts to overcome the gaps that may occur at the design stage of the knowledge base.

In order to show (validate) the efficiency of our proposed expert ANFIS model, we employed five MFs types out of the eleven MFs available in MATLAB (see section 3.4.2). Each one of the utilized MF has been represented by three different intervals (i.e. either 2,3, or 4 MFs) at a time. These MF's types and numbers are to be assigned to all input variables of ANFIS during the solving process. The duty of the proposed expert system, is to examine these combinations, and provide the best one.

At this stage, We have limited the ANFIS's knowledge bases structure into specific types and numbers of the MFs in order to reduce the model complexity to some extend. In other words, simplifying the model complexity will result in reducing the mathematical computations as well as the application running time. Consequently, we examined the mechanism of the proposed ANFIES model using bounded knowledge bases. However, the rest of the MFs types can be added in the future work. The gasoline prediction problem in Chapter (6) has been used as a test case for the third proposed model in this chapter.

By proposing this expert system and designing this application, we are trying to provide the researchers with various options that can be used to solve ANFIS problems. In fact, we know that each problem has its own characteristics, variables and data, which may be unique and not similar to any other problem. Therefore, we seek to obtain a tool that provides the researchers with various mathematically robust approaches in order to deal with different types of problems. By reviewing all the discussion and results shown in this chapter, we can conclude the following:

1. Using the proposed ANFIES model has contributed to obtaining better results. This has been achieved by investigating all possible MFs combinations in order to reach the best knowledge base; and provides the optimal ANFIS prediction model.

2. Solving time is different from one problem to another. It depends on various factors that can affect the timings. For instance, the data type, the data size, number of inputs, and the knowledge base structure (MFs types and numbers).

PART 4

Conclusions

Chapter 9

Conclusions and Future Work

In this chapter, we are introducing the thesis conclusions and the main research contributions. After that, the future work related to the research area can be concluded and given in order of priority.

9.1 Conclusions

In this work, we have addressed the problem of enhancing the prediction accuracy of a non-standardized operational research optimization model for solving the aggregate production planning problem. The main novelty and contribution of the present work is to propose methods to solve a complex prediction model in circumstances where the data are scarce, poorly scaled, and potentially noisy. Moreover, the complexity of the model can raise due to the fuzziness, uncertainty, fluctuation, and non-linearity extant in the problem nature. This research focused on proposing a mathematical model formed from robust pre-processing data techniques such as interpolation, approximation, and scaling, combined with expert ANFIS.

In chapter (6) we demonstrated novel modifications to the interpolation and approximation approaches in dealing with data scarcity for ANFIS problems. We developed a combined model that can deal with the rarity of data in order to improve prediction accuracy and model performance. This has been achieved by adding a pre-processing layer to ANFIS in order to replace the original discrete data by carefully optimised continuous data. The pre-processing layer composing the expansion and scaling models works sequentially. We used radial basis function (RBF) interpolation approaches (such as, linear, Cubic, and Multiquadric) as a data expansion model. Moreover, we used Chebyshev intervals and zero mean normalisation as scaling methods.

We have seen that when data is scarce, fitting continuous models to the data and re-sampling can provide richer information on which to build a model. In presenting this approach, we chose radial basis functions as our fitting models due to their sound theoretical properties; guaranteeing solvability, preserving the shape of the data, and having minimum variance to reduce over-fitting. In applying this technique we highlighted difficulties in fitting data near boundaries where over-smoothing can be a significant problem - particularly at turning points. We overcame this problem by modifying the multiquadric basis function to allow shape-preserving near-interpolation to take place within the convex hull of the original data set.

We have proposed three different approaches to process the data expansion providing more flexibility in determining the closest expansion pattern to the original problem. For instance, if we are dealing with a problem that needs the expanded data to be shape-preserving, then we use the linear approximation. In this case, the original data points are to be connected

by a straight line. In other words, the linear expansion can be considered as the best shape-preserving function because it appears in the form of straight lines that passes through every single point of the original data when it has applied. This method should be adopted when no former knowledge about the curvature of the underlying data is known.

When using multiquadric RBFs, we were able to reduce the amount of over-smoothing at turning points; which we demonstrated can cause significant problems. We noted, however, that this came at a cost of accurately interpolating the data. The increased control needed over the fitting function meant that we could produce only near-interpolation. That said, the control of the shape parameters could be tweaked to balance the smoothness properties with the fidelity of the fitting function. The cubic RBF interpolation showed considerable over-smoothing at the turning points, though it still passed through every point of the original data while interpolating. According to the nature of the problem under consideration, a trade-off can take place between either using a multiquadric RBF, which provides smooth near interpolation without over fitting; or applying cubic RBF that gives exact interpolation with some over-smoothing. Though it could be argued that both cubic RBF and multiquadric RBF can be used to fit interpolating curves, the cubic RBF is proven to have minimum variance properties which make it the most desirable continuous interpolating curve to use.

A further consideration in this chapter was to explore the impact of two different types of data scaling method on the accuracy of the final prediction model. We have found, on many occasions, significant problems when trying to accurately train fuzzy systems on poorly scaled data. Often, data can be drawn from a range of distributions where the data values can vary considerably between sets. We have seen from our test case, for example, how this can be problematic when training models to interpret data that have significantly different orders of magnitude. In fact, for the oil/fuel industry data, we were not able to train a satisfactory model without first normalizing the data. This was evident in our case study where the difference between temperature and the number of cars was in the order of 10^5. Most researchers believe that normalizing (or standardizing) the data can bring significant improvements in the training stages, and so we investigated two ways that this might be achieved. Though many pieces of research stated that normalization had taken place, few actually demonstrated its impact on the accuracy of the final models. Therefore, in this work, we looked at *Chebyshev scaling* and *zero-mean scaling (z-scores)*, specifically.

In chapter (7), we introduced our second proposed pre-processing data model represented by the use of Transformed Least Squares (TLS) as an outlier and noise mitigation solution method. We have classified the noise into two categories, i.e. Gaussian noise (modest or inconsiderable noise) and outliers (large discrete deviations from the underlying trend); in some cases, it can be a mixture of both. We were conscious of the great difficulties that can

arise from modelling with noisy data sets, which represents the biggest problem that can be posed in the data expansion stage of the modelling process. Many artificial intelligence approaches are greatly affected by noise and are often ill-equipped to deal with it. Least squares is known to be effective if the data is Gaussain in nature alone, but few can cope with outliers or even a mixture of both. It was therefore necessary to deal with the noisy data problem prior to the ANFIS optimisation stages for the predictions. Gaussian noise can often be dealt with very effectively by relaxing the interpolation conditions and choosing a subset of the data as RBF centres (possibly using k-means). This forces the models to approximate, rather than interpolate the data; following a standard least-squares approach to determine the fitting coefficients.

By relaxing the interpolation requirements of the fitting function, an approximating function can often smooth out the effect of the noise and pick up the underlying trend reliably. However, dealing with outliers is a greater challenge. For this reason, we took the opportunity to demonstrate the implementation of a method for mitigating the effect of outliers on the fitting curve by asymptotically bounding their contribution to the error function using a modified least-squares approach. We have found that using the RBFs (linear, Cubic, and Multiquadric) as the basis for our approximation functions for the TLS approach, can be an effective method in dealing with mixed noise distributions.

Additionally, in chapter (8), we developed An Adaptive Neuro-Fuzzy Inference Expert System (ANFIES) that can simulate all possible solutions inferred from different fuzzy knowledge base structures (MF types and numbers). We found that by letting the ANFIES search for the best ANFIS model, it provided superior results than the use of expert knowledge alone. Furthermore, we have developed an application tool using the MATLAB App Designer in order to provide a comprehensive and convenient solution environment for users which allows them to choose from various options, which includes all proposed models (i.e., pre-processing and expert system).

It is not the intention of this research to claim that ANFIS is the best approach to model this kind of problem. Nor are we saying that this approach to improving ANFIS modelling for scarce/sparse or poorly scaled data is better than any other approach. However, we are claiming that there is a need to derive useful models, for a vast range of problems, for which very little data is available and that this is a real problem not adequately addressed in the literature.

This work has clearly shown that where the quantity and scale of data is a problem, there are effective ways of dealing with it and the problem presented in this research helps us to demonstrate these methods. Even if the expansion method is a simple linear interpolation of the data, then significant improvement can be found. We believe that the approaches in this

thesis can easily be applied to similar problems encountered in artificial neural networks - particularly multi-layer perceptron networks and radial basis function networks.

9.2 Limitations of the Study

It is important to recognise that there are some assumptions and limitations associated with this work. Firstly, the data expansion methods will not fill gaps where data is missing. For example, if we were using data expansion to model a function $f(x)$ on a given range of x, but where data was missing for a large portion of the abscissa domain, we could not hope to recover the missing trend accurately. However, due to the properties of the cubic RBF, we can have confidence that the curve will remain fairly flat over the region of missing data. This would at least ensure boundedness on the reproduced values. A further limitation, when considering the TLS method for mitigating outliers is that this will only be effective where clear outliers can be detected. Where the underlying trend is naturally fluctuating or chaotic, the TLS curve may attempt to remove or smooth out portions of curve that are of interest and contribute to the prediction in a valid away. Also, with regard to the constructed application tool, when the model complexity increases, the solution times can take considerably longer due to the complex calculations and optimization process of the ANFIS algorithm. It can also be the case that solution times are not consistent between similar looking problems. Many factors can affect this such as: data type, data size, number of inputs, and the knowledge base structure (MFs types and numbers). Finally, all of the algorithms developed in this research make extensive use of Matlab's ANFIS library. The Matlab language that contains the ANFIS solver algorithm, i.e., the FIS generating functions (genfis), are not available yet for other applications such as Python. This has proved difficult when trying to extend the functionality of the system to incorporate techniques such as k-fold cross validation for example.

9.3 Future Work

In this work, we introduced a comprehensive, novel solution method for a particular problem, i.e., the impact of data scarcity and outliers in solving neuro-fuzzy problems; which represents a significant and challenging problem in other areas of optimization problems as well. We believe that what we have presented in this work represents the first step in dealing with such problems. However, different ideas, modifications, examination, and experiments have been left for the future due to lack of time. From our point of view, we can deliver the future development of the proposed model in five ways.

1. We are planning to investigate the improvement in performance that we may gain by employing different solution techniques and learning algorithms. This can be achieved by extending our proposed model. One of these techniques is k-fold cross-validation. This technique works on dividing the data sets into k subsets (so-called folds) that are equal in size. One set is to be considered as the validation subset, and the remaining (k-1) is to be the training subsets in which to interchange iteratively. This will allow the ANFIS model to process all the given data sets as training and validation subsets sequentially. In other words, the k-fold cross-validation technique may improve the ANFIS performance, and make it converge more reliably by flipping the training and testing subsets; which gives ANFIS the ability to train with all given data sets.

   However, in order to facilitate this additional work it will be necessary to consider developing the ANFIS solution in a different programming language. The reason for this is due to the way in which Matlab's ANFIS functions work. It is not currently possible to restart the training process from a pre-trained parameter set. The current Matlab implementation requires an initial FIS to be created before ANFIS optimises the model on the given data. However, the implementation of cross-validation requires the parameter set found from a given fold to be fed into the start of the next fold, hence building on the quality of the fit to the previous data set. Re-initialising the parameter set for each fold would defeat the object. As there was not sufficient time to redeploy this algorithm in a different programming language, it was not possible to implement cross-validation in the current study.

   We have proposed an initial structure for the combined novel ANFIS-Cross-Validation model to be developed under the future framework. We did not have the opportunity to add it to this work due to the time limitation. Figure (9.1) illustrates the general structure of the future proposed model.

2. We may extend the proposed expert ANFIS model, and the created application tool, in the future work by including several approaches, algorithms, and tools. For example, the ANFIS generation process can employ the fuzzy c-means and subtractive clustering algorithms in addition to the grid partition algorithm used in this work. Furthermore, the Simulink software can be combined with our MATLAB App-designer to produce a comprehensive solution method and predictor. We believe that this will improve the capabilities of the application tool as well as the proposed model.

3. The RBFs data expansion models are still believed to be promising in the future study due to their features and properties of simplicity, flexibility, and robustness. Therefore, it could be interesting to use the pre-processing data expansion models (i.e. the direct

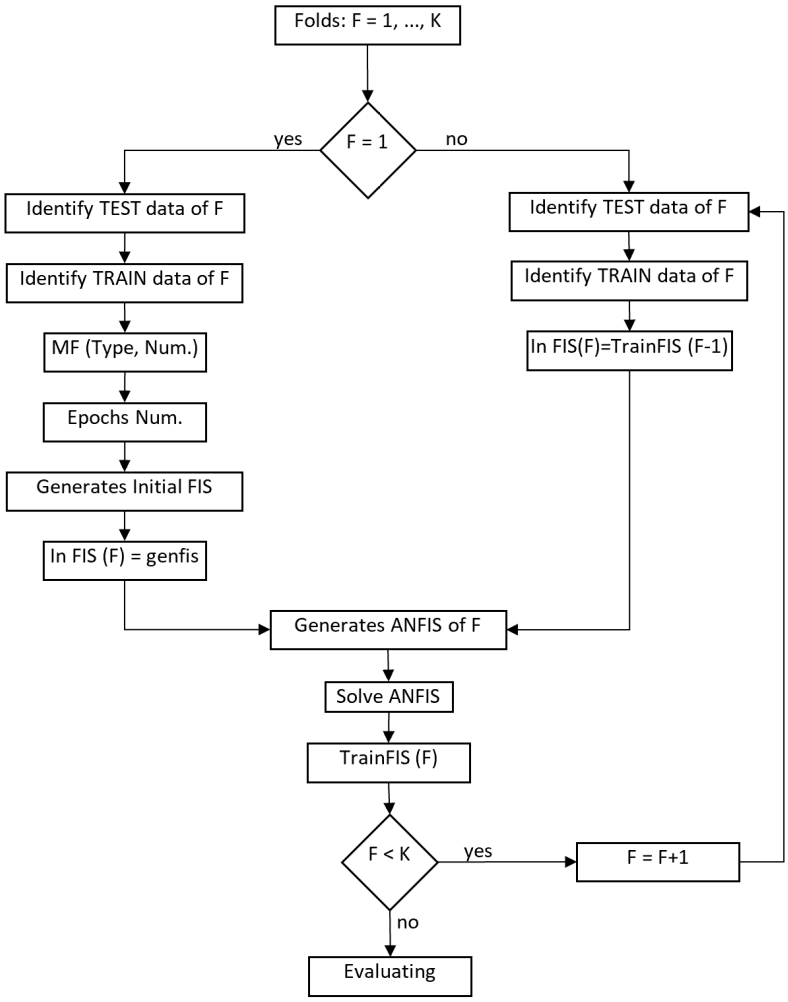


Fig. 9.1 Proposed model of using the K-Fold cross validation for ANFIS.

RBF interpolation and the TLS-RBF approximation) with other applications than ANFIS, where the data quality and quantity represents a key-factor of the optimization process. Multi-layer perceptrons and support vector machine regression are particular examples that could be considered.

4. As part of the future developments, the knowledge base options (i.e. MFs types and number) can be improved by adding more MF types to the application tool rather than using five types only. That will provide a broader range of input options, and could result in a more convenient solving environment to achieve more comprehensive outcomes.

5. Finally, we may explore the effectiveness of feeding the proposed system with a different range of input data and evaluating it by using the sensitivity analysis techniques.

Additionally, we can investigate the robustness of our proposed model by applying it into different domains.

References

[1] Abdipour, M., Younessi-Hmazekhanlu, M., Ramazani, S. H. R., and hassan Omidi, A. (2019). Artificial neural networks and multiple linear regression as potential methods for modeling seed yield of safflower (Carthamus tinctorius L.). *Industrial Crops and Products*, 127:185–194.

[2] Adedeji, P. A., Akinlabi, S., Madushele, N., and Olatunji, O. O. (2020). Hybrid adaptive neuro-fuzzy inference system (ANFIS) for a multi-campus university energy consumption forecast. *International Journal of Ambient Energy*, pages 1–10.

[3] Aengchuan, P. and Phruksaphanrat, B. (2018). Comparison of fuzzy inference system (FIS), FIS with artificial neural networks (FIS + ANN) and FIS with adaptive neuro-fuzzy inference system (FIS + ANFIS) for inventory control. *Journal of Intelligent Manufacturing*, 29(4):905–923.

[4] Ahanger, T. A., Tariq, U., Ibrahim, A., Ullah, I., and Bouteraa, Y. (2020). ANFIS-inspired Smart Framework for Education Quality Assessment. *IEEE Access*, pages 1–1.

[5] Ahmadlou, M., Karimi, M., Alizadeh, S., Shirzadi, A., Parvinnejhad, D., Shahabi, H., and Panahi, M. (2019). Flood susceptibility assessment using integration of adaptive network-based fuzzy inference system (ANFIS) and biogeography-based optimization (BBO) and BAT algorithms (BA). *Geocarto International*, 34(11):1252–1272.

[6] Al-Hmouz, A., Jun Shen, Al-Hmouz, R., and Jun Yan (2012). Modeling and Simulation of an Adaptive Neuro-Fuzzy Inference System (ANFIS) for Mobile Learning. *IEEE Transactions on Learning Technologies*, 5(3):226–237.

[7] Alizadeh, M., Rada, R., Jolai, F., and Fotoohi, E. (2011). An adaptive neuro-fuzzy system for stock portfolio analysis. *International Journal of Intelligent Systems*, 26(2):99–114.

[8] Alizadeh, R., Jia, L., Nellippallil, A. B., Wang, G., Hao, J., Allen, J. K., and Mistree, F. (2019). Ensemble of surrogates and cross-validation for rapid and accurate predic-

tions using small data sets. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 33(4):484–501.

[9] Amarbayasgalan, T., Pham, V. H., Theera-Umpon, N., and Ryu, K. H. (2020). Unsupervised Anomaly Detection Approach for Time-Series in Multi-Domains Using Deep Reconstruction Error. *Symmetry*, 12(8):1251.

[10] Amid, S. and Mesri Gundoshmian, T. (2017). Prediction of output energies for broiler production using linear regression, ANN (MLP, RBF), and ANFIS models. *Environmental Progress & Sustainable Energy*, 36(2):577–585.

[11] Anderson, D. R., Sweeney, D. J., Williams, T. A., Camm, J. D., Cochran, J. J., Fry, M. J., and Ohlamann, J. W. (2013). *Quantitative Methods for Business*. Cengage Learning, Boston, 12 edition.

[12] Antholzer, S., Haltmeier, M., and Schwab, J. (2019). Deep learning for photoacoustic tomography from sparse data. *Inverse Problems in Science and Engineering*, 27(7):987–1005.

[13] Ashour, M. A. H., Al-Dahhan, I. A. H., and Hassan, A. K. (2020). Forecasting by Using the Optimal Time Series Method. In *IHIET 2020*, pages 148–154. Springer.

[14] Atsalakis, G. S., Protopapadakis, E. E., and Valavanis, K. P. (2016). Stock trend forecasting in turbulent market periods using neuro-fuzzy systems. *Operational Research*, 16(2):245–269.

[15] Azadeh, A., Asadzadeh, S., and Ghanbari, A. (2010). An adaptive network-based fuzzy inference system for short-term natural gas demand estimation: Uncertain and complex environments. *Energy Policy*, 38(3):1529–1536.

[16] Azadeh, A., Asadzadeh, S., Mirseraji, G., and Saberi, M. (2015a). An emotional learning-neuro-fuzzy inference approach for optimum training and forecasting of gas consumption estimation models with cognitive data. *Technological Forecasting and Social Change*, 91(2015):47–63.

[17] Azadeh, A., Zarrin, M., Rahdar Beik, H., and Aliheidari Bioki, T. (2015b). A neuro-fuzzy algorithm for improved gas consumption forecasting with economic, environmental and IT/IS indicators. *Journal of Petroleum Science and Engineering*, 133:716–739.

[18] Barak, S. and Sadegh, S. S. (2016). Forecasting energy consumption using ensemble ARIMA–ANFIS hybrid algorithm. *International Journal of Electrical Power & Energy Systems*, 82:92–104.

[19] Barrodale, I. (1968). L 1 Approximation and the Analysis of Data. *Applied Statistics*, 17(1):51.

[20] Beatson, R. K. and Bui, H. Q. (2007). Mollification formulas and implicit smoothing. *Advances in Computational Mathematics*, 27(2):125–149.

[21] Beigi, G., Ranganath, S., and Liu, H. (2019). Signed Link Prediction with Sparse Data: The Role of Personality Information. In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 1270–1278, New York, NY, USA. ACM.

[22] Benmouiza, K. and Cheknane, A. (2019). Clustered ANFIS network using fuzzy c-means, subtractive clustering, and grid partitioning for hourly solar radiation forecasting. *Theoretical and Applied Climatology*, 137(1-2):31–43.

[23] Berredjem, T. and Benidir, M. (2018). Bearing faults diagnosis using fuzzy expert system relying on an Improved Range Overlaps and Similarity method. *Expert Systems with Applications*, 108:134–142.

[24] Bilbao, I. and Bilbao, J. (2017). Overfitting problem and the over-training in the era of data. In *ICICIS 2017*, pages 173–177. IEEE.

[25] Bonakdari, H., Moeeni, H., Ebtehaj, I., Zeynoddin, M., Mahoammadian, A., and Gharabaghi, B. (2019). New insights into soil temperature time series modeling: linear or nonlinear? *Theoretical and Applied Climatology*, 135(3-4):1157–1177.

[26] Box, G. E. P., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2016). *Time Series Analysis: Forecasting and Control*. Wiley Series in Probability and Statistics, 5th edition.

[27] Boyacioglu, M. A. and Avci, D. (2010). An adaptive network-based fuzzy inference system (ANFIS) for the prediction of stock market return: The case of the Istanbul stock exchange. *Expert Systems with Applications*, 37(12):7908–7912.

[28] Buhmann, M. D. (2003). *Radial Basis Functions: Theory and Implementations*. Cambridge Monographs on Applied and Computational Mathematics.

[29] Callan, R. (1999). *The Essence of Neural Networks*. Prentice-Hall, Great Britain.

[30] Chen, G. and Pham, T. T. (2001). *Introduction to Fuzzy Sets , Fuzzy Logic and Fuzzy Control Systems*. CRC Press LLC, Florida.

[31] Chen, H., Zhang, Y., Chen, Y., Zhang, J., Zhang, W., Sun, H., Lv, Y., Liao, P., Zhou, J., and Wang, G. (2018a). LEARN: Learned Experts' Assessment-Based Reconstruction Network for Sparse-Data CT. *IEEE Transactions on Medical Imaging*, 37(6):1333–1347.

[32] Chen, J., Liu, S., Zhang, W., and Liu, Y. (2020). Uncertainty quantification of fatigue S-N curves with sparse data using hierarchical Bayesian data augmentation. *International Journal of Fatigue*, 134:105511.

[33] Chen, J.-F., Do, Q., Nguyen, T., and Doan, T. (2018b). Forecasting Monthly Electricity Demands by Wavelet Neuro-Fuzzy System Optimized by Heuristic Algorithms. *Information*, 9(3):51.

[34] Chen, J. F. and Do, Q. H. (2014). A cooperative cuckoo search-hierarchical adaptive neuro-fuzzy inference system approach for predicting student academic performance. *Journal of Intelligent and Fuzzy Systems*, 27(5):2551–2561.

[35] Chen, Z. and Cheng, L. (2020). Mitigating Outliers for Bayesian Mixture of Factor Analyzers. In *2020 IEEE 11th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pages 1–5. IEEE.

[36] Coppejans, M. (2003). Flexible but Parsimonious Demand Designs: The Case of Gasoline. *Review of Economics and Statistics*, 85(3):680–692.

[37] Crespo Marques, E., Maciel, N., Naviner, L., Cai, H., and Yang, J. (2019). A Review of Sparse Recovery Algorithms. *IEEE Access*, 7:1300–1322.

[38] Das, S., Datta, S., and Chaudhuri, B. B. (2018). Handling data irregularities in classification: Foundations, trends, and future challenges. *Pattern Recognition*, 81:674–693.

[39] Dewan, M. W., Huggett, D. J., Warren Liao, T., Wahab, M. A., and Okeil, A. M. (2016). Prediction of tensile strength of friction stir weld joints with adaptive neuro-fuzzy inference system (ANFIS) and neural network. *Materials & Design*, 92:288–299.

[40] Efendigil, T., Önüt, S., and Kahraman, C. (2009). A decision support system for demand forecasting with artificial neural networks and neuro-fuzzy models: A comparative analysis. *Expert Systems with Applications*, 36(3):6697–6707.

[41] Elena Dragomir, O., Dragomir, F., Stefan, V., and Minca, E. (2015). Adaptive Neuro-Fuzzy Inference Systems as a Strategy for Predicting and Controling the Energy Produced from Renewable Sources. *Energies*, 8(11):13047–13061.

[42] Emamgholizadeh, S., Moslemi, K., and Karami, G. (2014). Prediction the Groundwater Level of Bastam Plain (Iran) by Artificial Neural Network (ANN) and Adaptive Neuro-Fuzzy Inference System (ANFIS). *Water Resources Management*, 28(15):5433–5446.

[43] Fachini, F. and Lopes, B. (2019). Critical Bus Voltage Mapping using ANFIS with regards to Max Reactive Power in PV buses. In *2019 IEEE Milan PowerTech*, pages 1–6. IEEE.

[44] Feng, C., Liang, J., Song, P., and Wang, Z. (2020). A fusion collaborative filtering method for sparse data in recommender systems. *Information Sciences*, 521:365–379.

[45] Firat, M. and Güngör, M. (2010). Monthly total sediment forecasting using adaptive neuro fuzzy inference system. *Stochastic Environmental Research and Risk Assessment*, 24(2):259–270.

[46] Fuller, R. (1998). *Fuzzy Reasoning and Fuzzy Optimization*. Abo: Turku Centre for Computer Science, no.9 edition.

[47] Galperin, E. A. and Kansa, E. J. (2002). Application of global optimization and radial basis functions to numerical solutions of weakly singular Volterra integral equations. *Computers and Mathematics with Applications*, 43(3-5):491–499.

[48] Geiger, A., Liu, D., Alnegheimish, S., Cuesta-Infante, A., and Veeramachaneni, K. (2020). TadGAN: Time Series Anomaly Detection Using Generative Adversarial Networks.

[49] Gosasang, V., Chandraprakaikul, W., and Kiattisin, S. (2011). A Comparison of Traditional and Neural Networks Forecasting Techniques for Container Throughput at Bangkok Port. *The Asian Journal of Shipping and Logistics*, 27(3):463–482.

[50] Guan, J., Shi, D., Zurada, J. M., and Levitan, A. S. (2014). Analyzing Massive Data Sets: An Adaptive Fuzzy Neural Approach for Prediction, with a Real Estate Illustration. *Journal of Organizational Computing and Electronic Commerce*, 24(1):94–112.

[51] Gupta, P. K. and Hira, D. S. (2008). *Operations Research*. S. Chand & Company LTD., India.

[52] Han, J. and Kamber, M. (2006). *Data Mining : Concepts and Techniques*. Elsevier Science & Technology, USA, second edition.

[53] Hao, J., Kim, Y., Kim, T.-K., and Kang, M. (2018). PASNet: pathway-associated sparse deep neural network for prognosis prediction from high-throughput data. *BMC Bioinformatics*, 19(1):510.

[54] Hodge, V. J. and Austin, J. (2004). A Survey of Outlier Detection Methodologies. *Artificial Intelligence Review*, 22(2):85–126.

[55] Hosseinpour, M., Yahaya, A. S., Ghadiri, S. M., and Prasetijo, J. (2013). Application of Adaptive Neuro-fuzzy Inference System for road accident prediction. *KSCE Journal of Civil Engineering*, 17(7):1761–1772.

[56] Hsu, S. C. (2016). Fuzzy Time Series Customers Prediction: Case Study of an E-Commerce Cash Flow Service Provider. *International Journal of Computational Intelligence and Applications*, 15(04):1650024.

[57] Iranmanesh, H., Abdollahzade, M., and Miranian, A. (2012). Mid-term energy demand forecasting by hybrid neuro-fuzzy models. *Energies*, 5(1):1–21.

[58] Jackson, P. (1998). *Introduction to Expert Systems*. Addison-Wesley Longman Publishing Co.

[59] Jang, J.-S. R. (1991). Fuzzy Modeling Using Generalized Neural Networks and Kalman Filter Algorithm. In *AAAI-91*, pages 762–767. AAAI.

[60] Jang, J.-S. R. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3):665–685.

[61] Jang, J.-S. R., Sun, C. T., and Mizutani, E. (1997). *Neuro-Fuzzy and Soft Computing*. Prentice-Hall, Englewood Cliffs.

[62] Jenkinson, D., Crampton, A., Mason, J., Cox, M. G., Forbes, A., and Boudjemae, R. (2004). Parameterised approximation estimators for mixed noise distributions. In Ciarlini, P., Cox, M. G., Pavese, F., Rossi, G. B., and Richter, D., editors, *Advanced Mathematical and Computational Tools in Metrology VI (Series on Advances in Mathematics for Applied Sciences: Vol 66)*, pages 67–81. World Scientific, ISBN 978-981-238-904-6.

[63] Jones, A. H. S., Pranolo, A., Dianto, A., and Winiarti, S. (2018). Prediction of population growth using Sugeno and Adaptive Neuro-Fuzzy Inference System (ANFIS). In *IOP Conference Series: Materials Science and Engineering*, volume 403, page 012073.

[64] Kadam, A. K., Wagh, V. M., Muley, A. A., Umrikar, B. N., and Sankhua, R. N. (2019). Prediction of water quality index using artificial neural network and multiple linear regression modelling approach in Shivganga River basin, India. *Modeling Earth Systems and Environment*, 5(3):951–962.

[65] Kar, S., Das, S., and Ghosh, P. K. (2014). Applications of neuro fuzzy systems: A brief review and future outline. *Applied Soft Computing*, 15:243–259.

[66] Karaboga, D. and Kaya, E. (2019). Adaptive network based fuzzy inference system (ANFIS) training approaches: a comprehensive survey. *Artificial Intelligence Review*, 52(4):2263–2293.

[67] Kaveh, M., Rasooli Sharabiani, V., Amiri Chayjan, R., Taghinezhad, E., Abbaspour-Gilandeh, Y., and Golpour, I. (2018). ANFIS and ANNs model for prediction of moisture diffusivity and specific energy consumption potato, garlic and cantaloupe drying under convective hot air dryer. *Information Processing in Agriculture*, 5(3):372–387.

[68] Khademi, F., Jamal, S. M., Deshpande, N., and Londhe, S. (2016). Predicting strength of recycled aggregate concrete using Artificial Neural Network, Adaptive Neuro-Fuzzy Inference System and Multiple Linear Regression. *International Journal of Sustainable Built Environment*, 5(2):355–369.

[69] Khoshnevisan, B., Rafiee, S., Omid, M., and Mousazadeh, H. (2014). Prediction of potato yield based on energy inputs using multi-layer adaptive neuro-fuzzy inference system. *Measurement: Journal of the International Measurement Confederation*, 47(1):521–530.

[70] Kim, Y. H., Choi, M. J., Kim, E. J., and Song, J. W. (2019). Magnetic-Map-Matching-Aided Pedestrian Navigation Using Outlier Mitigation Based on Multiple Sensors and Roughness Weighting. *Sensors*, 19(21):4782.

[71] Kiritsis, D. (1995). A review of knowledge-based expert systems for process planning. Methods and problems. *The International Journal of Advanced Manufacturing Technology*, 10(4):240–262.

[72] Laptev, N., Yosinski, J., Li, L. E., and Smyl, S. (2017). Time-series Extreme Event Forecasting with Neural Networks at Uber. In *ICML 2017 Time Series Workshop*, Sydney, Australia.

[73] Li, D.-C., Wu, C., and Chang, F. M. (2005). Using data-fuzzification technology in small data set learning to improve FMS scheduling accuracy. *The International Journal of Advanced Manufacturing Technology*, 27(3-4):321–328.

[74] Li, D.-C., Wu, C.-S., Tsai, T.-I., and Chang, F. M. (2006). Using mega-fuzzification and data trend estimation in small data set learning for early FMS scheduling knowledge. *Computers & Operations Research*, 33(6):1857–1869.

[75] Lohani, A., Kumar, R., and Singh, R. (2012). Hydrological time series modeling: A comparison between adaptive neuro-fuzzy, neural network and autoregressive techniques. *Journal of Hydrology*, 442-443:23–35.

[76] MacAllister, A., Kohl, A., and Winer, E. (2020). Using high-fidelity meta-models to improve performance of small dataset trained Bayesian Networks. *Expert Systems with Applications*, 139:112830.

[77] Mahdavi, Z. and Khademi, M. (2012). Prediction of Oil Production with: Data Mining, Neuro-Fuzzy and Linear Regression. *International Journal of Computer Theory and Engineering*, 4(3):446–447.

[78] Mahmud, M. S. and Meesad, P. (2016). An innovative recurrent error-based neuro-fuzzy system with momentum for stock price prediction. *Soft Computing*, 20(10):4173–4191.

[79] Majdisova, Z. and Skala, V. (2017). Radial basis function approximations: comparison and applications. *Applied Mathematical Modelling*, 51:728–743.

[80] Majid, R. (2018). Advances in Statistical Forecasting Methods: An Overview. *Economic Affairs*, 63(4).

[81] Makridakis, S., Wheelwright, S. C., and Hyndman, R. J. (1998). *Forecasting: Methods and Applications*. John Wiley & Sons, Inc, 3rd edition.

[82] Mamdani, E. and Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7(1):1–13.

[83] Mashaly, A. F. and Alazba, A. A. (2018). Membership function comparative investigation on productivity forecasting of solar still using adaptive neuro-fuzzy inference system approach. *Environmental Progress & Sustainable Energy*, 37(1):249–259.

[84] Matyjaszek, M., Riesgo Fernández, P., Krzemień, A., Wodarski, K., and Fidalgo Valverde, G. (2019). Forecasting coking coal prices by means of ARIMA models and

neural networks, considering the transgenic time series theory. *Resources Policy*, 61:283–292.

[85] Mekanik, F., Imteaz, M. A., and Talei, A. (2016). Seasonal rainfall forecasting by adaptive network-based fuzzy inference system (ANFIS) using large scale climate signals. *Climate Dynamics*, 46(9-10):3097–3111.

[86] Mendel, J. M., Hagras, H., Tan, W.-W., Melek, W. W., and Ying, H. (2014). *Introduction to type-2 fuzzy logic control: theory and applications*. John Wiley & Sons.

[87] Mitrea, C. A., Lee, C. K. M., and Wu, Z. (2009). A Comparison between Neural Networks and Traditional Forecasting Methods: A Case Study. *International Journal of Engineering Business Management*, 1:11.

[88] Mollaiy-Berneti, S. (2016). Optimal design of adaptive neuro-fuzzy inference system using genetic algorithm for electricity demand forecasting in Iranian industry. *Soft Computing*, 20(12):4897–4906.

[89] Montgomery, D. C., Jennings, C. L., and Kulahci, M. (2008). *Introduction to Time Series Analysis and Forecasting*. Wiley Series in Probability and Statistics, New Jersey.

[90] Mordjaoui, M. and Boudjema, B. (2011). Forecasting and Modelling Electricity Demand Using Anfis Predictor. *Journal of Mathematics and Statistics*, 7(4):275–281.

[91] Munir, M., Siddiqui, S. A., Dengel, A., and Ahmed, S. (2019). DeepAnT: A Deep Learning Approach for Unsupervised Anomaly Detection in Time Series. *IEEE Access*, 7:1991–2005.

[92] Nadimi, V., Azadeh, A., Pazhoheshfar, P., and Saberi, M. (2010). An Adaptive-Network-Based Fuzzy Inference System for Long-Term Electric Consumption Forecasting (2008-2015): A Case Study of the Group of Seven (G7) Industrialized Nations: U.S.A., Canada, Germany, United Kingdom, Japan, France and Italy. In *2010 Fourth UKSim European Symposium on Computer Modeling and Simulation*, pages 301–305, Pisa, Italy. IEEE.

[93] Nahmias, S. (2009). Aggregate Planning. In *Production and Operations Analysis*, pages 124–168. McGraw- Hill, New York.

[94] Najah, A. A., El-Shafie, A., Karim, O. A., and Jaafar, O. (2012). Water quality prediction model utilizing integrated wavelet-ANFIS model with cross-validation. *Neural Computing and Applications*, 21(5):833–841.

[95] Nanda, J., Das, L. D., Choudhury, S., and Parhi, D. R. (2020). Revelence of Multiple Breathing Cracks on Fixed Shaft Using ANFIS and ANN. In *Innovative Product Design and Intelligent Manufacturing Systems*, pages 599–618.

[96] Nedjah, N. and Mourelle, L. d. M. (2005). *Fuzzy Systems Engineering: Theory and Practice*. Springer Science & Business Media.

[97] Nguyen, H. T., Walker, C. L., and Walker, E. A. (2018). *A First Course in Fuzzy Logic*. CRC Press LLC, forth edit edition.

[98] Nikitin, A. V. and Davidchack, R. L. (2019). Hidden Outlier Noise and its Mitigation. *IEEE Access*, 7:87873–87886.

[99] Nikolić, M., Šelmić, M., Macura, D., and Ćalić, J. (2020). Bee Colony Optimization metaheuristic for fuzzy membership functions tuning. *Expert Systems with Applications*, 158:113601.

[100] Noori, R., Khakpour, A., Omidvar, B., and Farokhnia, A. (2010). Comparison of ANN and principal component analysis-multivariate linear regression models for predicting the river flow based on developed discrepancy ratio statistic. *Expert Systems with Applications*, 37(8):5856–5862.

[101] Okwu, M. O. and Adetunji, O. (2018). A comparative study of artificial neural network (ANN) and adaptive neuro-fuzzy inference system (ANFIS) models in distribution system with nondeterministic inputs. *International Journal of Engineering Business Management*, 10:184797901876842.

[102] Panapakidis, I. P. and Dagoumas, A. S. (2017). Day-ahead natural gas demand forecasting based on the combination of wavelet transform and ANFIS/genetic algorithm/neural network model. *Energy*, 118:231–245.

[103] Pao, H.-T. (2008). A comparison of neural network and multiple regression analysis in modeling capital structure. *Expert Systems with Applications*, 35(3):720–727.

[104] Parvizi, B., Khanlarkhani, A., Palizdar, Y., and Farshad, A. (2019). Comparison of ANFIS and ANN modeling for predicting the behavior of a catalytic methane reformer. *Bulgarian Chemical Communications*, 51(2):190–199.

[105] Pozo-Pérez, J. A., Medina, D., Herrera-Pinzón, I., Heßelbarth, A., and Ziebold, R. (2017). Robust Outlier Mitigation in Multi-Constellation GNSS-based Positioning for Waterborne Applications. pages 1330–1343.

[106] Pratihar, D. K. (2015). *Soft computing: Fundamental and applications*. Narosa Publishing House, New Delhi, revised edition.

[107] Rajab, S. and Sharma, V. (2018). A review on the applications of neuro-fuzzy systems in business. *Artificial Intelligence Review*, 49(4):481–510.

[108] Rajabi, M., Hossani, S., and Dehghani, F. (2019). A literature review on current approaches and applications of fuzzy expert systems. *arXiv preprint arXiv:1909.08794*.

[109] Rezakazemi, M., Dashti, A., Asghari, M., and Shirazian, S. (2017). H2-selective mixed matrix membranes modeling using ANFIS, PSO-ANFIS, GA-ANFIS. *International Journal of Hydrogen Energy*, 42(22):15211–15225.

[110] Rivett, P. (1980). *Model building for decision analysis*. John Wiley & Sons.

[111] Runkler, T. (1997). Selection of appropriate defuzzification methods using application specific properties. *IEEE Transactions on Fuzzy Systems*, 5(1):72–79.

[112] Saber Iraji, M., Aboutalebi, M., Seyedaghaee, N. R., and Tosinia, A. (2012). Students Classification With Adaptive Neuro Fuzzy. *International Journal of Modern Education and Computer Science*, 4(7):42–49.

[113] Sajadi, S. M., Asadzadeh, S. M., Majazi Dalfard, V., Nazari Asli, M., and Nazari-Shirkouhi, S. (2013). A new adaptive fuzzy inference system for electricity consumption forecasting with hike in prices. *Neural Computing and Applications*, 23(7-8):2405–2416.

[114] Salleh, M. N. M., Talpur, N., and Talpur, K. H. (2018). A Modified Neuro-Fuzzy System Using Metaheuristic Approaches for Data Classification. In *Artificial Intelligence - Emerging Trends and Applications*. InTech.

[115] Schaback, R. (1995). Multivariate interpolation and approximation by translates of a basis function. *Series In Approximations and Decompositions*, 6:491–514.

[116] Sen, S., Sezer, E., Gokceoglu, C., and Yagiz, S. (2012). On sampling strategies for small and continuous data with the modeling of genetic programming and adaptive neuro-fuzzy inference system. *Journal of Intelligent & Fuzzy Systems*, 23(6):297–304.

[117] Siddique, N. and Adeli, H. (2013). *Computational intelligence synergies of fuzzy logic, neural networks and evolutionary computing*. John Wiley & Sons, New Delhi.

[118] Sinha, S., Vaidya, U., and Yeung, E. (2019). On Computation of Koopman Operator from Sparse Data. In *2019 American Control Conference (ACC)*, pages 5519–5524. IEEE.

[119] Sirabahenda, Z., St-Hilaire, A., Courtenay, S. C., and van den Heuvel, M. R. (2020). Assessment of the effective width of riparian buffer strips to reduce suspended sediment in an agricultural landscape using ANFIS and SWAT models. *CATENA*, 195:104762.

[120] Sivanandam, S. N., Sumathi, S., and Deepa, S. N. (2007). *Introduction to fuzzy logic using MATLAB*. Springer-Verlag Berlin Heidelberg.

[121] Skala, V. (2017). RBF Interpolation with CSRBF of Large Data Sets. *Procedia Computer Science*, 108:2433–2437.

[122] Smolik, M. and Skala, V. (2017). Large scattered data interpolation with radial basis functions and space subdivision. *Integrated Computer-Aided Engineering*, 25(1):49–62.

[123] Stojčić, M., Pamučar, D., Mahmutagić, E., and Stević, Ž. (2018). Development of an ANFIS Model for the Optimization of a Queuing System in Warehouses. *Information*, 9(10):240.

[124] Su, C. H. and Cheng, C. H. (2016). A hybrid fuzzy time series model based on ANFIS and integrated nonlinear feature selection method for forecasting stock. *Neurocomputing*, 205.

[125] Subramanian, J. and Simon, R. (2013). Overfitting in prediction models – Is it a problem only in high dimensions? *Contemporary Clinical Trials*, 36(2):636–641.

[126] Sugeno, M. and Kang, G. (1988). Structure identification of fuzzy model. *Fuzzy Sets and Systems*, 28(1):15–33.

[127] Svalina, I., Galzina, V., Lujić, R., and Šimunović, G. (2013). An adaptive network-based fuzzy inference system (ANFIS) for the forecasting: The case of close price indices. *Expert Systems with Applications*, 40(15):6055–6063.

[128] Taha, H. A. (2007). *Operations Research : An Introduction*. Pearson, Singapore, eighth edition.

[129] Taha, H. A. (2017). *Operations Research: An Introduction*. Pearson Education Limited, Malaysia, tenth edition.

[130] Takagi, T. and 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.

[131] Takwoingi, Y., Guo, B., Riley, R. D., and Deeks, J. J. (2017). Performance of methods for meta-analysis of diagnostic test accuracy with few studies or sparse data. *Statistical Methods in Medical Research*, 26(4):1896–1911.

[132] Talei, A., Chua, L. H. C., Quek, C., and Jansson, P. E. (2013). Runoff forecasting using a Takagi-Sugeno neuro-fuzzy model with online learning. *Journal of Hydrology*, 488:17–32.

[133] Talei, A., Chua, L. H. C., and Wong, T. S. (2010). Evaluation of rainfall and discharge inputs used by Adaptive Network-based Fuzzy Inference Systems (ANFIS) in rainfall–runoff modeling. *Journal of Hydrology*, 391(3-4):248–262.

[134] Tang, K. H. D., Md Dawal, S. Z., and Olugu, E. U. (2018). Integrating fuzzy expert system and scoring system for safety performance evaluation of offshore oil and gas platforms in Malaysia. *Journal of Loss Prevention in the Process Industries*, 56:32–45.

[135] Tavana, M. and Hajipour, V. (2019). A practical review and taxonomy of fuzzy expert systems: methods and applications. *Benchmarking: An International Journal*, 27(1):81–136.

[136] Vasileva-Stojanovska, T., Vasileva, M., Malinovski, T., and Trajkovik, V. (2015). An ANFIS model of quality of experience prediction in education. *Applied Soft Computing*, 34:129–138.

[137] Viharos, Z. and Kis, K. (2015). Survey on Neuro-Fuzzy systems and their applications in technical diagnostics and measurement. *Measurement*, 67:126–136.

[138] Wagner, H. M. (1972). *Principles of Operations Research: With Application to Managerial Decisions*. Prentice-Hall International, London.

[139] Wagner, P. D., Fiener, P., Wilken, F., Kumar, S., and Schneider, K. (2012). Comparison and evaluation of spatial interpolation schemes for daily rainfall in data scarce regions. *Journal of Hydrology*, 464-465:388–400.

[140] Wang, J. G. and Liu, G. R. (2002). A point interpolation meshless method based on radial basis functions. *International Journal for Numerical Methods in Engineering*, 54(11):1623–1648.

[141] Wang, J. S. and Ning, C. X. (2015). ANFIS based time series prediction method of bank cash flow optimized by adaptive population activity PSO algorithm. *Information (Switzerland)*, 6(3):300–313.

[142] Wang, X., Lin, J., Patel, N., and Braun, M. (2016). A Self-Learning and Online Algorithm for Time Series Anomaly Detection, with Application in CPU Manufacturing. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1823–1832, New York, NY, USA. ACM.

[143] Wang, X., Lin, J., Patel, N., and Braun, M. (2018). Exact variable-length anomaly detection algorithm for univariate and multivariate time series. *Data Mining and Knowledge Discovery*, 32(6):1806–1844.

[144] Wen, X., Feng, Q., Yu, H., Wu, J., Si, J., Chang, Z., and Xi, H. (2015). Wavelet and adaptive neuro-fuzzy inference system conjunction model for groundwater level predicting in a coastal aquifer. *Neural Computing and Applications*, 26(5):1203–1215.

[145] Werbos, P. (1974). *Beyond regression : new tools for prediction and analysis in the behavioral sciences*. Phd thesis, Harvard University.

[146] Wong, Y. J., Arumugasamy, S. K., Chung, C. H., Selvarajoo, A., and Sethu, V. (2020). Comparative study of artificial neural network (ANN), adaptive neuro-fuzzy inference system (ANFIS) and multiple linear regression (MLR) for modeling of Cu (II) adsorption from aqueous solution using biochar derived from rambutan (Nephelium lappaceum) pee. *Environmental Monitoring and Assessment*, 192(7):439.

[147] Xu, Q. and Liu, Z. (2019). Scattered Data Interpolation and Approximation with Truncated Exponential Radial Basis Function. *Mathematics*, 7(11):1101.

[148] Yaïci, W. and Entchev, E. (2016). Adaptive Neuro-Fuzzy Inference System modelling for performance prediction of solar thermal energy system. *Renewable Energy*, 86:302–315.

[149] Yang, H., Antonante, P., Tzoumas, V., and Carlone, L. (2020). Graduated Non-Convexity for Robust Spatial Perception: From Non-Minimal Solvers to Global Outlier Rejection. *IEEE Robotics and Automation Letters*, 5(2):1127–1134.

[150] Yang, Y., Chen, Y., Wang, Y., Li, C., and Li, L. (2016). Modelling a combined method based on ANFIS and neural network improved by DE algorithm: A case study for short-term electricity demand forecasting. *Applied Soft Computing Journal*, 49:663–675.

[151] YAO, X. (1993). Evolutionary Artificial Neural Networks. *International Journal of Neural Systems*, 04(03):203–222.

[152] Yaseen, Z. M., Ebtehaj, I., Bonakdari, H., Deo, R. C., Danandeh Mehr, A., Mohtar, W. H. M. W., Diop, L., El-shafie, A., and Singh, V. P. (2017). Novel approach for streamflow forecasting using a hybrid ANFIS-FFA model. *Journal of Hydrology*, 554:263–276.

[153] Yaseen, Z. M., Ghareb, M. I., Ebtehaj, I., Bonakdari, H., Siddique, R., Heddam, S., Yusif, A. A., and Deo, R. (2018). Rainfall Pattern Forecasting Using Novel Hybrid Intelligent Model Based ANFIS-FFA. *Water Resources Management*, 32(1):105–122.

[154] Yip, H.-l., Fan, H., and Chiang, Y.-h. (2014). Predicting the maintenance cost of construction equipment: Comparison between general regression neural network and Box–Jenkins time series models. *Automation in Construction*, 38:30–38.

[155] Yu, X. and Baek, S. J. (2017). Energy-Efficient Collection of Sparse Data in Wireless Sensor Networks Using Sparse Random Matrices. *ACM Transactions on Sensor Networks*, 13(3):1–36.

[156] Zadeh, L. (1971). Quantitative fuzzy semantics. *Information Sciences*, 3(2):159–176.

[157] Zadeh, L. (1975a). The concept of a linguistic variable and its application to approximate reasoning-I. *Information Sciences*, 8(3):199–249.

[158] Zadeh, L. (1975b). The concept of a linguistic variable and its application to approximate reasoning-II. *Information Sciences*, 8(4):301–357.

[159] Zadeh, L. (1975c). The concept of a linguistic variable and its application to approximate reasoning-III. *Information Sciences*, 9(1):43–80.

[160] Zadeh, L. A. (1965). Fuzzy Sets. *Information and Control*, 8:338–353.

[161] Zadeh, L. A. (1973). Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(1):28–44.

[162] Zadeh, L. A. (2008). Is there a need for fuzzy logic? *Information Sciences*, 178(13):2751–2779.

[163] Zadeh, L. A. and Aliev, R. A. (2018). *Fuzzy Logic Theory and Applications*. World Scientific.

[164] Zamani, H., Dehghani, M., Diaz, F., Li, H., and Craswell, N. (2018). SIGIR 2018 Workshop on Learning from Limited or Noisy Data for Information Retrieval. In *The*

41st International ACM SIGIR Conference on Research & Development in Information Retrieval - SIGIR '18, pages 1439–1440, New York, New York, USA. ACM Press.

[165] Zhou, Y., Guo, S., and Chang, F.-J. (2019). Explore an evolutionary recurrent ANFIS for modelling multi-step-ahead flood forecasts. *Journal of Hydrology*, 570:343–355.

[166] Zounemat-Kermani, M. and Teshnehlab, M. (2008). Using adaptive neuro-fuzzy inference system for hydrological time series prediction. *Applied Soft Computing Journal*, 8(2):928–936.

[167] Đokić, A. and Jović, S. (2017). Evaluation of agriculture and industry effect on economic health by ANFIS approach. *Physica A: Statistical Mechanics and its Applications*, 479:396–399.

[168] Şahin, M. and Erol, R. (2017). A Comparative Study of Neural Networks and ANFIS for Forecasting Attendance Rate of Soccer Games. *Mathematical and Computational Applications*, 22(4):43.