Data-Driven Modelling and Prediction of Alloy Steel Properties using Fuzzy Systems with Special Emphasis on Type-2 and Quantum Fuzzy Sets

Raymond Muscat

Supervisor: Prof. Mahdi Mahfouf

Department of Automatic Control and Systems Engineering
The University of Sheffield

A thesis submitted for the degree of
Doctor of Philosophy

May 2020

Acknowledgements

I would like to express my appreciation towards all those who have supported me during my research studies.

First and foremost I would like to thank my supervisor, Professor Mahdi Mahfouf, for his professional guidance throughout the course of my PhD studies. His unwavering support and patience have been an inspiration to continue my research even at the most testing times.

I thank the Department of Automatic Control and Systems Engineering for awarding me a scholarship to pursue my studies and all the staff for their help and support. I also thank the members of the Intelligent Systems Lab, particularly Olusayo for the long discussions and sharing of ideas on various subjects.

A word of thanks also goes to all those I met at the Catholic Chaplaincy throughout the four years I spent in Sheffield, thanks to whom I found a home away from home.

I also thank my friends, both in Sheffield, especially George and the other housemates for the cups of tea we shared together, and in Malta, Daniel, Robert, Joe Borg and Joe Attard, for their sustained encouragement.

Finally, I am very grateful to my parents, Emanuel and Victoria, and my sister Anne Marie for their love, continuous support and understanding.

Abstract

Testing of materials in industry is very important to ensure product quality and safety. However, such tests are usually very time-consuming and expensive to perform. The motivation for this thesis is to address these challenges by proposing the combination of techniques from machine learning and fuzzy logic to the materials property prediction problem, to assist the pursuit of producing materials with specific required properties.

The first part of the research aims at designing a modelling architecture to deal with the imbalanced data relating to the production of rails. The modelling techniques are based on Support Vector Machines (SVMs). Results show that SVMs are sensitive to class imbalance. Subsequently, an internal class imbalance learning method (through a Biased Fuzzy SVM) and an external class imbalance learning method (data under-sampling of the majority class) are applied to the data. The performance of the techniques when implemented on the under-sampled dataset is better, while in both cases the inclusion of a fuzzy membership improves the performance of the SVM. Fuzzy C-Means (FCM) Clustering is analysed for reducing the number of support vectors of the Fuzzy SVM model, concluding it is effective in reducing model complexity without any significant performance deterioration.

The second part of the research deals with modelling Charpy impact data of heat-treated steel to predict Charpy energy. This is a challenging modelling problem because although the test is governed by a specific standard, several sources of disturbance give rise to uncertainty in the data. The data are also multidimensional, sparsely distributed and the relation between the variables and the output is highly nonlinear. A neuro-fuzzy modelling framework is employed which uses a new type of membership function, the Quantum membership function. Results are encouraging, with further investigation necessary to better understand Quantum membership functions and the effect that Quantum intervals have when modelling highly uncertain data. The framework is further improved using Interval Type-2 Fuzzy Sets and optimised using a genetic algorithm. The prediction result is 0.72% better than the best result found in the literature when modelling the same dataset.

Table of Contents

List of Figures

List of Tables

Abbreviations

AI	Artificial Intelligence
APF	Atomic Packing Factor
BCC	Body Centred Cubic
BNN	Bayesian Neural Network
BOS	Basic Oxygen Steelmaking
DEC	Different Error Costs
EAF	Electric Arc Furnace
FCC	Face Centred Cubic
FCM	Fuzzy C-Means
FLS	Fuzzy Logic System
FN	False Negative
FOU	Footprint of Uncertainty
FP	False Positive
FSVM	Fuzzy Support Vector Machine
GA-NN	Genetic Algorithm Neural Network
GC	Generalised Centroid
GMM	Gaussian Mixture Model
GrC	Granular Computing

HCP	Hexagonal Close Packed
IT2	Interval Type-2
KM	Karnik-Mendel
LMF	Lower Membership Function
MLP	Multilayer Perceptron
NDT	Non-Destructive Testing
NF	Neural-Fuzzy
NN	Neural Network
RBF	Radial Basis Function
RMSE	Root-Mean-Square Error
SMOTE	Synthetic Minority Over-sampling Technique
SVM	Support Vector Machine
SVR	Support Vector Regression
TN	True Negative
TP	True Positive
TSK	Takagi-Sugeno-Kang
UMF	Upper Membership Function

Chapter 1

Introduction

There are several manual experimental tests used in industry to verify that materials are produced to the required standards. Such tests are usually very time-consuming and expensive to perform. However, these tests are very important to ensure product quality and safety.

The testing of materials was greatly improved during the 19th century with the development of new steelmaking technologies. It is also closely associated with the rapid expansion of the global railway network and the accompanying areas of engineering science through the need for locomotives and supporting structures [85]. During these early times, testing provided a convenient and effective way of understanding the load-carrying capacity and critical fracture stress of a component.

Materials testing was further improved with the increased use of metals for construction. This required a better understanding of the characteristics and behaviour of the materials operating under various conditions.

One of the tests employed to test materials is the Charpy impact test. It is used to evaluate the toughness of a material and to measure the resistance of a material to brittle fracture. The importance of the Charpy test is evident when one considers the extent to which brittle fracture of metals can be dangerous for a structure.

A dramatic example of this happened during World War II when a large number of Liberty ships developed fatigue cracks while some ships literally split in half. It was later concluded that these failures were due to a number of factors including welding defects, design flaws (such as square hatches that acted as points of stress concentration where cracks could form), but also the embrittlement of steel due to the cold sea water and the effects of the ductile-to-brittle transition (which is determined from the Charpy test) [65].

A fairly recent area of computer science is the application of Artificial Intelligence (AI) to assist real life applications. In fact, technology has transformed our lives and AI has become an integrated part of many things that we use every day, sometimes without us even noticing.

Some of the areas using AI include: optimised supply chains, automatic fault detection, customer experience models, smart products or appliances, self-driving cars, smart homes, robotics, software assistants and games.

It is claimed that AI models are capable of predicting outcomes more accurately than humans while continuously learning. A question that the academic and industrial communities tried to answer is: Could the application of artificial intelligence techniques to the material property prediction and testing problem be a solution to assist the production of materials with the required properties?

1.1 Research Aim and Objectives

The aim of this research is to study the applicability of novel machine learning algorithms to the materials property prediction problem.

The main objectives of the project are:

- To build a model that classifies whether steel rails are good or bad. The model will be based on techniques derived from Support Vector Machines (SVMs) and will need to address the high data imbalance between the classes.
- To develop a modelling framework to predict the Charpy energy of heat-treated alloy steel. The framework will be based on the Adaptive-Network-based Fuzzy Inference System (ANFIS) architecture and make use of the Quantum function as a fuzzy membership function.
- To enhance the developed modelling framework by integrating Interval Type-2 Fuzzy Sets with the Quantum membership function.

1.2 Thesis Subject Area

In this respect, this research will focus on the intersection of three key subject areas as illustrated in Figure 1.1.

Increasing computing power and growth in available data have enabled the application of machine learning and data modelling techniques to complex systems [84, 19, 70, 34]. Machine learning is a subset of AI which uses algorithms to detect patterns in data. These algorithms learn from the data and are then used to make predictions. The two most common scenarios where machine learning is used are classification and regression. This ‘black box’ modelling can also assist in decision-making and to derive insights into complex systems.

Many real-world problems include some level of uncertainty. Among other factors, such uncertainty may arise from errors in measurement or due to the inherent complexity of the system. Fuzzy models intrinsically account for these by using IF-THEN rules to represent input-output relations. These linguistic rules and the use of fuzzy sets also increase the interpretability of the model, making it more human-readable [15, 73].

This interdisciplinary approach to modelling can be applied to materials science. Data in this domain are usually multi-variable, without explicit understanding of how the variables interact with each other. Therefore, modelling can extract useful knowledge from data, and be employed to predict complex material properties such as strength, toughness and fatigue.

1.3 Thesis Outline

The thesis is organised as follows:

- Chapter 2 provides an overview of the steelmaking process, heat treatment of steel and property testing of materials. The Charpy impact dataset is also introduced along with literature related to this dataset.

- Chapter 3 introduces the second dataset used in this thesis, the rail manufacturing data. The SVM techniques used to classify this imbalanced dataset are detailed and the results obtained are presented.

- Chapter 4 proposes the use of a new type of fuzzy membership function, the Quantum membership function, in a fuzzy logic framework. A detailed description of the

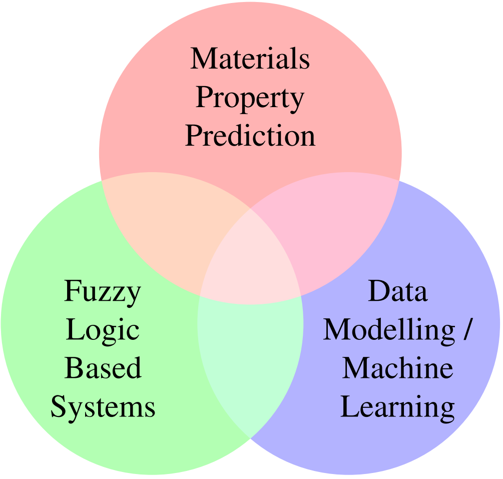

Materials Property Prediction
Fuzzy Logic Based Systems
Data Modelling / Machine Learning

Figure 1.1 Research areas for thesis

suggested modelling architecture is given, together with the results of the framework when applied to the Charpy impact energy dataset.

- Chapter 5 extends the modelling framework discussed in Chapter 4 to include Type-2 Fuzzy Sets. This was done to investigate the uncertainty handling capabilities of the model.

- Chapter 6 concludes the thesis and offers some recommendations for future work.

It should be noted that the modelling and analysis throughout this project were carried out in MATLAB.

1.4 Key Contributions and Publications

- Created a novel learning machine by combining the properties of an SVM with a fuzzy vector penalty term and further modifying the objective function to include a Different Error Costs (DEC) class imbalance learning method. The novel DEC Fuzzy Support Vector Machine (FSVM) was applied to the classification of good and rejected rails from sample experimental data. The results obtained were compared with those obtained from classical models (SVM and FSVM) with the DEC-FSVM demonstrating better sensitivity.When the DEC-FSVM was compared to the models built using an external imbalance learning method (under-sampling of the majority class), the latter showed a better sensitivity.

- Successfully identified clustering as an effective way to facilitate the reduction of support vectors in an SVM while maintaining satisfactory performance. The results obtained by [10, 94] were also reconfirmed, where model training times were reduced.

- Successfully applied the Quantum membership function modelling framework to a regression problem for the first time using the Charpy impact data.

- Compared the performances of the results obtained from the Quantum membership function modelling framework with 5 other models whose results were published in the literature. The results when applied to the Charpy impact data show that the best results were obtained by the Genetic Algorithm Neural Network (GA-NN) Ensemble method closely followed by the result obtained using the Quantum membership function modelling framework.

- The Quantum membership function modelling framework was enhanced using Type-2 Fuzzy Sets with the model being optimised using a genetic algorithm. The results were again compared to those in the literature, this time the Type-2 Quantum membership function fuzzy model demonstrating the best overall result.

This research work led to the following publications:

- Muscat, R., Mahfouf, M., Zughrat, A., Yang, Y., Thornton, S., Khondabi, A., and Sortanos, S. (2014). Hierarchical Fuzzy Support Vector Machine (SVM) for Rail Data Classification. IFAC Proceedings Volumes (19th IFAC World Congress), 47(3), 10652–10657.

- Muscat, R. and Mahfouf, M. (2016). Predicting Charpy Impact Energy for Heat-Treated Steel using a Quantum-Membership-Function-based Fuzzy Model. IFAC-PapersOnLine (17th IFAC Symposium on Control, Optimization and Automation in Mining, Mineral and Metal Processing – MMM 2016), 49(20), 138–142.

Chapter 2

Background and Literature Review

2.1 The Steelmaking Process

While iron artefacts date back to around 2000 BC, there is evidence that iron was smelted even earlier in various parts of the world. Apart from iron being the fourth most abundant element on the earth's crust, these early civilisations discovered the usefulness of this metal. As an alloy of iron, steel is the most widely used and recycled material on earth. It continues to transform our lives in sectors such as construction, transport and space travel, energy generation and distribution, machinery in various industries, specialised instruments in the health sector, and country infrastructure.

In Britain, iron making methods were improved during the 800 years of Roman occupation which ended in 383 AD. During this period, natural draught for smelting was enhanced by using foot-operated bellows. However, after the Romans, this technological knowledge was lost, with evidence that only primitive methods were used until the 14th century [58].

During the 17th century, the cementation process was used to convert wrought iron into stiffer blister steel, which could then be forged and hammered into various grades. In 1740 Benjamin Huntsman who worked as a clockmaker in Doncaster, used a crucible method to produce steel for clock springs which was more consistent and had better properties. This process contributed to the high reputation of Sheffield steels [58].

Charcoal was used to smelt iron until 1709 when Abraham Darby operated a blast furnace in Coalbrookdale using coke. This resulted in ironworks being moved from woodlands to coal mine sites over the next half a century. The next important improvement in blast furnace operation was when James Neilson introduced the hot blast in Glasgow in 1828 [58].

Pig iron was converted to wrought iron by dry puddling, a process which Henry Cort invented in 1783. In 1854, James Nasmyth patented the 'wet puddling' method which involved passing steam through molten iron. This led to Henry Bessemer inventing a similar process in 1855 [58].

Further developments on the Bessemer process included the use of basic instead of acidic furnace linings in the Basic Bessemer or Thomas Converter, and the use of pure oxygen instead of air in the steelmaking process [58, 57].

2.1.1 Blast Furnace

The blast furnace remains an important part of the steel production route, and is usually the main component of a complex plant. It is a tall vertical shaft of about 40 metres, made of steel and lined with refractory bricks, which uses carbon, mainly in the form of coke, to reduce iron from its oxide ores. A schematic view of such a furnace is illustrated in Figure 2.1 [67].

The first stage in the steelmaking process is the preparation of the charge for the blast furnace which involves converting the ores to sinter. This is done by crushing the ores and heating them using hot gases to remove moisture and other volatile impurities. Sintered iron ore, coke and limestone are then charged in layers through the top of the furnace. Coke serves two functions, as a fuel and providing carbon monoxide when burnt, which is the main reducing agent. The main reduction equation for the iron oxides inside the furnace is:

$$Fe_2O_3 - 3\,[O] \longrightarrow 2\,Fe \qquad (2.1)$$

The blast furnace aims at producing molten iron and therefore the temperature is the critical parameter of its operation. For tapping the iron and slag in their molten state, the temperature in the hearth of the furnace must be higher than 1500 °C. Hot-blast air, hydrocarbons, as well as natural gas and powdered coal, are blown through the tuyeres at the bottom of the structure, reducing the iron ore and producing slag and pig iron, also called hot metal, containing about 4% Carbon [51, 67].

2.1.2 Desulfurisation

Sulphur adversely affects the structure and mechanical properties of steel, unless good machinability is required [71]. Lime, calcium carbide and magnesium are used to treat hot metal from the blast furnace, where the sulphur chemically reacts with these reagents and is then removed as slag [24].

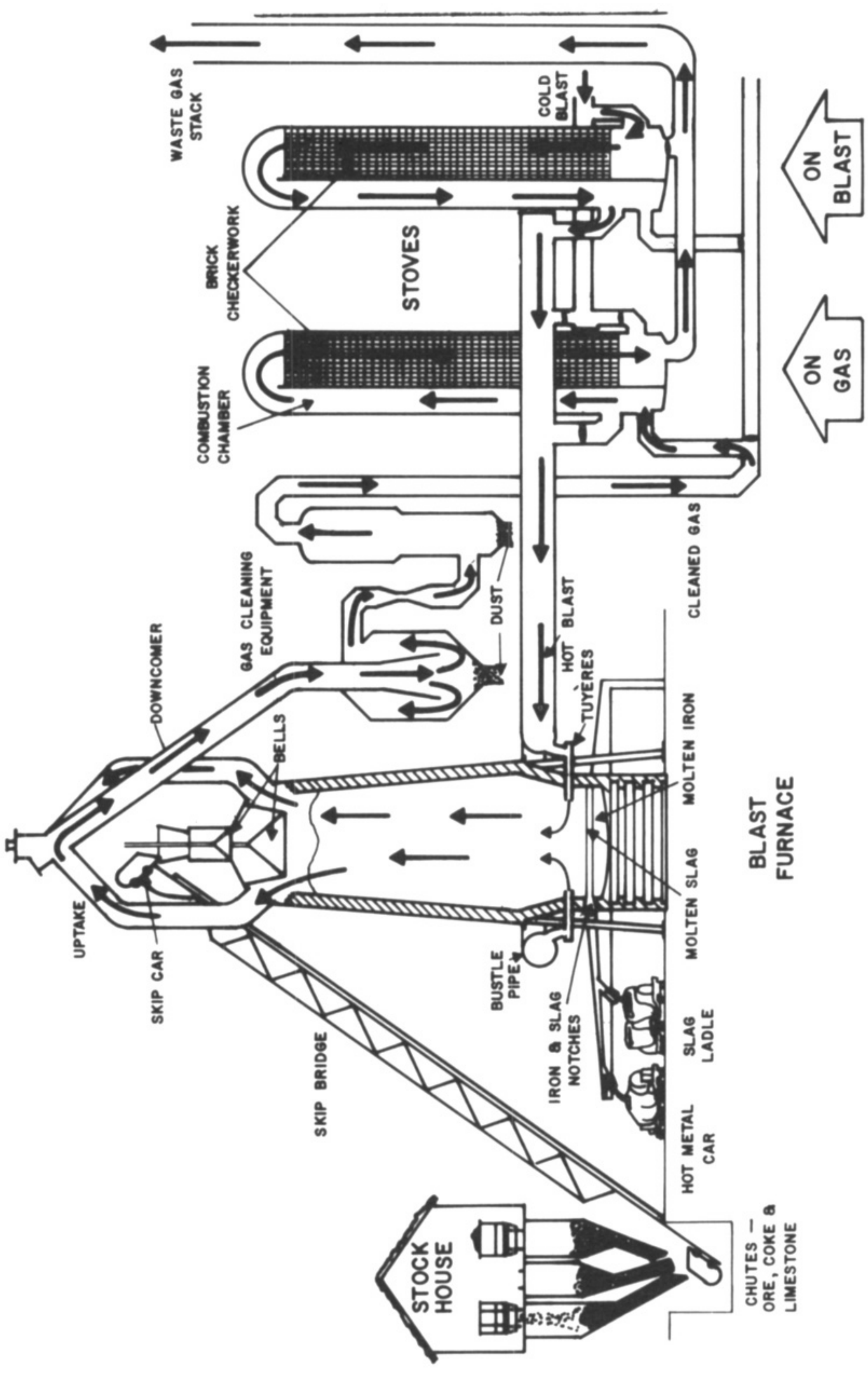


Figure 2.1 Representation of a blast furnace plant [67]

2.1.3 Steelmaking

The hot metal is transferred to a Basic Oxygen Steelmaking (BOS) converter where high-velocity pure oxygen is blown, oxidising the carbon and other impurities (Figure 2.2a). An alternative production route involves scrap metal and directly reduced iron (also known as sponge iron) being refined in an Electric Arc Furnace (EAF) (Figure 2.2b). According to [92], in 2016, around 25% of the 1630 million tons of steel in the world was produced in EAFs.

2.1.4 Secondary Steelmaking

The tight specifications of produced steel are adjusted during secondary steelmaking (or ladle metallurgy). This might include stages in various ladles for deoxidation, alloying, heating and degassing [49].

2.1.5 Continuous Casting

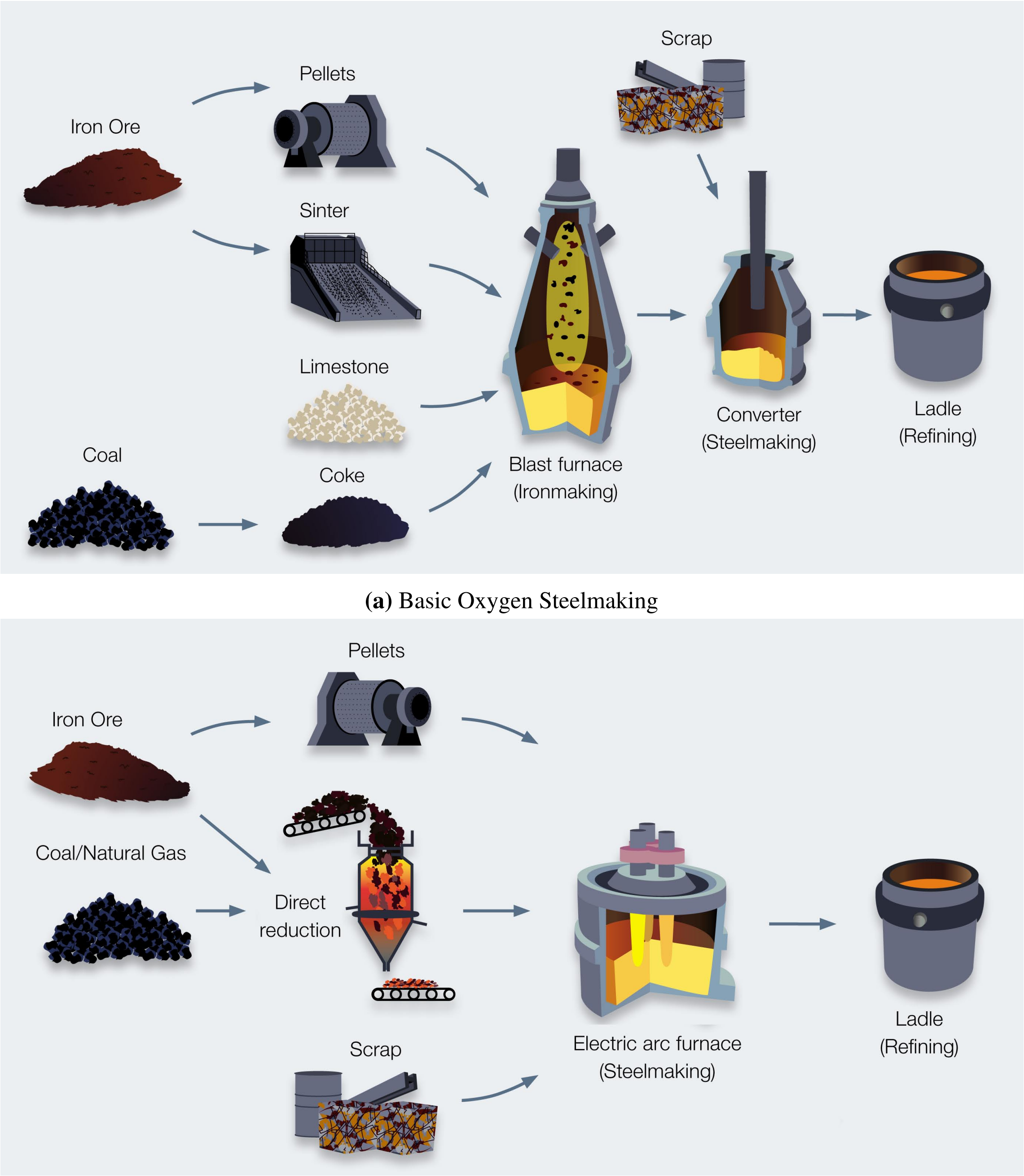
Traditionally, steel used to then be cast into ingots, before being milled into finished products. To increase efficiency, it is nowadays more commonly poured directly from the ladles into a tundish as illustrated in Figure 2.3, from which it flows into a mold and is continuously cast while being cooled down and milled into several required products [49].

2.2 Heat Treatment

2.2.1 Metallic Structure

The structure of a substance affects its properties. In a gas, the kinetic energy of the particles allows them to move in a random way, diffuse and fill the available space. Gases are poor conductors of heat and electricity. The particles in a liquid also fill the container in which they are placed but they form a surface. Their kinetic energy is less than the particles of a gas, there are some forces between them but they are still able to move freely within the liquid. Particles in a solid are closely bound together. This gives a solid its structure as the particles are arranged in a regular pattern and not randomly [17].

Metals are held together by strong metallic bonding where positive ions are surrounded by a ‘sea’ of electrons. These delocalised electrons come from the outer shell of the metal atoms and their negative charge attracts the positively charged ions and binds the metal nuclei together. Generally, metals have high melting points and they are good conductors of heat and electricity [17].



(a) Basic Oxygen Steelmaking

(b) Electric Arc Furnace Steelmaking

Figure 2.2 Steelmaking routes (adapted from [91])

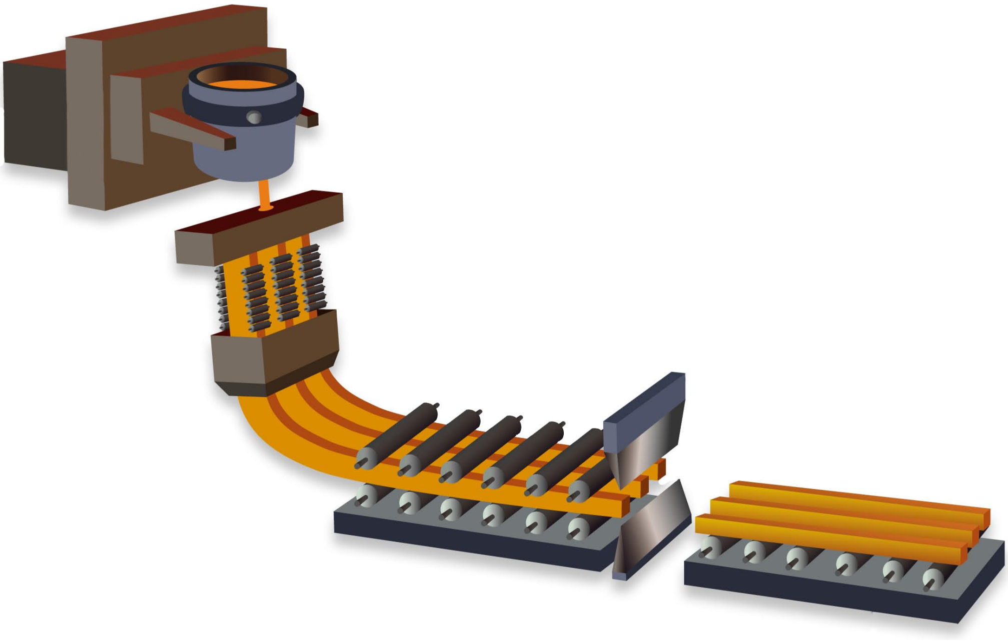


Figure 2.3 Continuous casting (adapted from [91])

Solid metals are said to be crystalline. During solidification from a liquid, nucleation takes place where the crystal structure starts to form. The unit cell is the smallest unit that retains the size, shape and atomic arrangement of the crystal lattice. The unit cell is therefore geometrically repeated throughout the whole material. The nuclei form dendrites, eventually continuing to grow into grains of the metal [9].

Although metals are strongly bonded they do not have directed bonds such that particle layers can slide over each other. In the metallic lattice, this relative movement of layers is called ‘slip’. This allows metals to be hammered or drawn, metallic properties that are called malleability and ductility. Two important methods of preventing slip by strengthening metals are reducing the crystal grain size and alloying. Alloying is the combination of two or more metals, or a metal and a non-metal, changing the properties of the original material. This changes a metal’s strength and its resistance to corrosion and wear and is very important practically. To a large extent, the properties of an alloy depend on the packing arrangement of the particles within it [31].

There are three packing arrangements for the unit cell of metals and their alloys. These are: Face Centred Cubic (FCC), Body Centred Cubic (BCC) and Hexagonal Close Packed (HCP). Apart from their geometrical shape, these arrangements differ from each other in their coordination number, which is the number of particles touching a single particle, and the atomic packing factor (APF), which is an approximation of the space occupied by the particles [9]. These properties are shown in Table 2.1.

Table 2.1 Properties of packing arrangements [78]

	FCC	BCC	HCP
Coordination Number	12	8	12
Atomic Packing Factor	74%	68%	74%

2.2.2 Iron-Carbon Phase Diagram

Pure iron has a BCC structure below 912 °C. This form of iron is called α-iron or ferrite. This changes to an FCC structure between 912 °C and 1394 °C and is called γ-iron or austenite. Beyond 1394 °C the crystal structure changes back to BCC and δ-iron or δ-ferrite is formed. The melting temperature of iron is 1538 °C.

These transformations are better explained through a phase diagram. A phase diagram, also called an equilibrium diagram, shows the various phases of a system with respect to the controllable variables temperature, pressure and composition, considering slow heating or cooling.

Iron is usually presented as a binary phase diagram with carbon (Figure 2.4), which is its most important alloying element. In a binary phase diagram, the temperature and composition of the alloy are variable while the pressure is constant. When alloying, one element's atoms can either substitute the other element's atoms, or stay between the atoms in the crystal lattice.

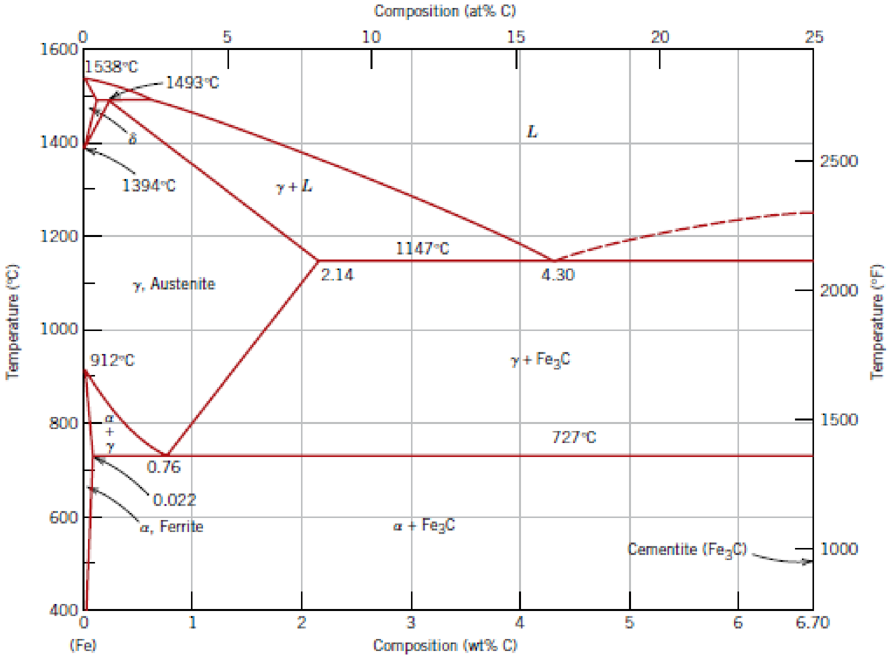


Figure 2.4 Iron-Carbon phase diagram [9]

In the case of iron and carbon, carbon forms an interstitial solution in between the iron atoms. The percentage weight of carbon soluble in iron also depends on the crystal structure. At low temperatures, the α-ferrite BCC structure has relatively small interstitial positions (maximum solubility of 0.022% carbon at 727 °C). At higher temperatures, FCC has larger interstitial positions for austenite (maximum solubility of 2.14% carbon at 1147 °C) [9].

As shown in the diagram, carbon changes the arrest points (or transformation temperatures) of the phases. At 4.3% carbon and 1147 °C, the liquid alloy solidifies to austenite and cementite (Fe_3C). This is called the eutectic point and the alloys formed below this point are called cast irons. The point at 0.76% carbon and 727 °C is called the eutectoid and is very important for engineering steels. During cooling austenite is transformed into α-iron and cementite. Steels with less than 0.76% carbon are called hypoeutectoid while those with a higher percentage of carbon are called hypereutectoid.

There may be other elements present in the steels apart from carbon. Some of them, such as manganese, nickel, chromium and vanadium, are used to control the steel properties, while others, such as sulphur and phosphorus, are classed as impurities and removed during steelmaking and ladle metallurgy processes [18].

The microstructure of steel depends on the carbon content, composition and the rate of cooling. Therefore phase diagrams, along with isothermal transformation diagrams (or temperature, time, percentage transformation diagrams) and continuous cooling transformation diagrams, are very important for the design of heat treating procedures. Phase transformations of various structures, such as pearlite, bainite, spheroidite and martensite, occur depending on nucleation, grain growth and diffusion rates. These structures are obtained using procedures such as normalising, annealing, quenching and tempering.

2.3 Steel Properties and Testing

Mechanical properties of steel are physical properties which describe the material when a load is applied. Some of the key properties in steel specification are stiffness, hardness, strength, ductility, fatigue and toughness. This section will describe the standard tests that are used to simulate loading of a material in a real application. These tests are performed by varying a number of factors related to the load and the surrounding environment. The load may be tensile, compressive or shear, and the magnitude of the force may vary constantly with time or fluctuate. It can be applied over a short or long period of time. The temperature can also be an important factor. The standardised methods in which the tests are performed ensure reproducible and consistent interpretation of results.

2.3.1 Tensile Test

One of the most common tests used to guarantee the quality of a product is the tensile test. This determines how steel reacts when a force is applied to it in tension. Figure 2.5 shows a diagram of the apparatus used. During the test a specimen of standard dimensions is stretched apart at a constant rate while measuring the load and the extension. These are converted to engineering stress (which is the force divided by the original cross-sectional area) and engineering strain (which is taken as the ratio of the extension to the original length), and a stress-strain curve is obtained.

As shown in Figure 2.6, the specimen exhibits a linear relationship between stress and strain until the elastic limit is reached. The slope of the linear portion approximates Young's Modulus. Beyond the elastic limit, starting at the yield point, the specimen is plastically deformed. During the phase of plastic deformation internal structural changes occur and several bands, known as Luders bands, are formed at points of stress concentration. The point of maximum stress is called the ultimate tensile strength, point at which the so-called 'necking' starts to form in the specimen. After the ultimate tensile strength the stress reduces until the fracture point. The area under the nonlinear portion of the graph is related to the energy absorbed during deformation and is therefore an indication of the toughness of the material [30].

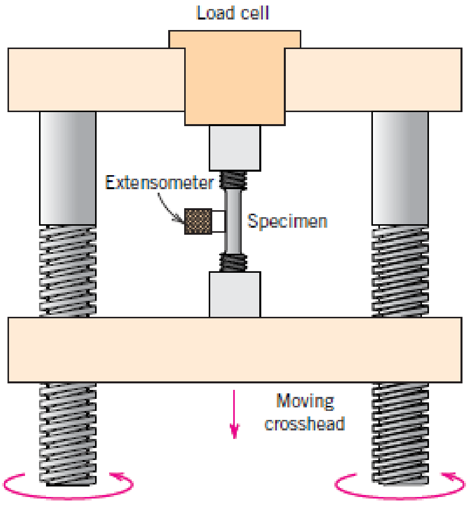


Figure 2.5 Schematic of the apparatus for the tensile test [9]

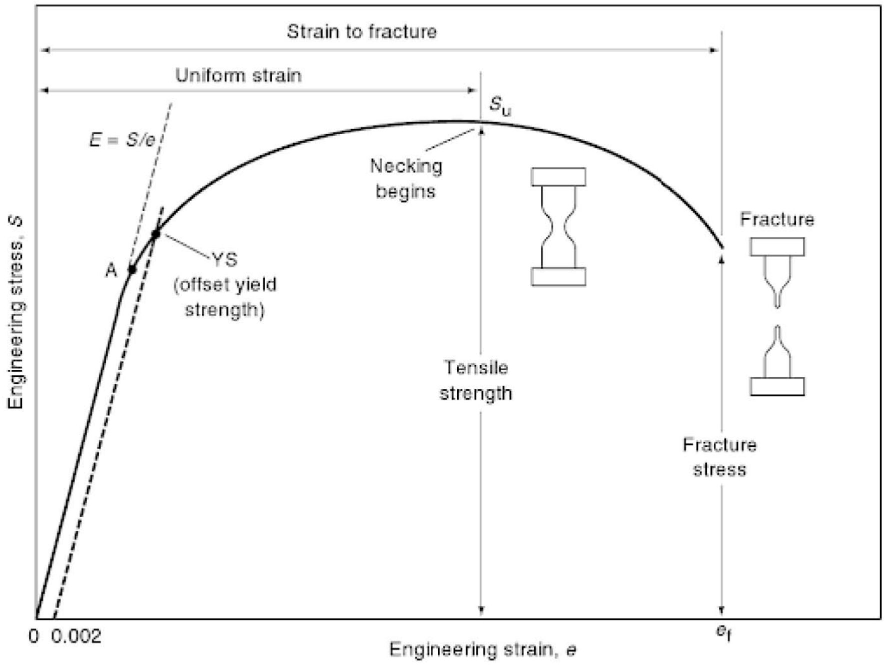


Figure 2.6 An example of a Stress-Strain curve [59]

2.3.2 Hardness Test

Hardness is a property of the material that describes its resistance to being deformed. Early hardness tests compared scratch resistance on a scale called the Mohs scale. More recently, quantitative techniques were developed which analyse a material's resistance to indentation. There are several tests used to determine hardness such as the Brinell, Vickers and Rockwell tests. The hardness is related to the ratio between a fixed load and the surface area of the indentation formed on a flat surface of the material. The different tests differ mainly in the type of indentor used [1].

2.3.3 Charpy Impact Test

The Charpy impact test is a standard test used to measure the impact energy (also referred to as notch toughness) absorbed by a material during fracture. Figure 2.7 shows a schematic diagram of the apparatus used in the test while Figure 2.8 depicts the notched test specimen which is supported horizontally in the anvil. The notch provides a point of stress concentration within the specimen and improves the reproducibility of the results. The absorbed energy is computed by working out the potential energy lost by a pendulum through breaking the specimen. Results from tests performed at different temperatures are used to determine the ductile-to-brittle transition temperature of materials. Figure 2.9 shows various impact energy curves for steel for varying carbon content. At higher temperatures, the impact energy absorbed is high, corresponding to a ductile fracture mode. Over a comparatively narrow temperature range the impact energy reduces rapidly and the low impact energy values correspond to a brittle fracture mode.

Although the test is governed by a standard test procedure, several variables influence the test result repeatability [9, 56]. In fact, through convention, the test is performed on three specimens at the same temperature and the results are averaged. However, the test is still susceptible to a number of uncertainties as outlined in [47, 81], giving rise to erratically distributed data. The sources of disturbance can be grouped as follows:

- Specimen (e.g. notch geometry, inhomogeneous distribution of atoms during the early stages of nucleation, duplex grain structures including both coarse and fine grains lead to inconsistent energy distribution, chemical composition)
- System (e.g. machine stiffness and friction, calibration settings)
- Environment (e.g. ambient and specimen temperatures)
- Procedure (e.g. human error)

When combined with a highly sparse data distribution, this suggests that modelling Charpy impact test data is a challenging task.

2.4 The Impact Energy Data

The heat-treated steel Charpy impact dataset was provided by Tata Steel Europe. During previous work [83], the data was cleaned and pre-processed with a metallurgist providing expert knowledge throughout this process.

2.4.1 Data Collection

The data were collected from different sites which had process variations depending on the steel product produced. The variables in the database had measurement tolerances as indicated in Table 2.2 [83].

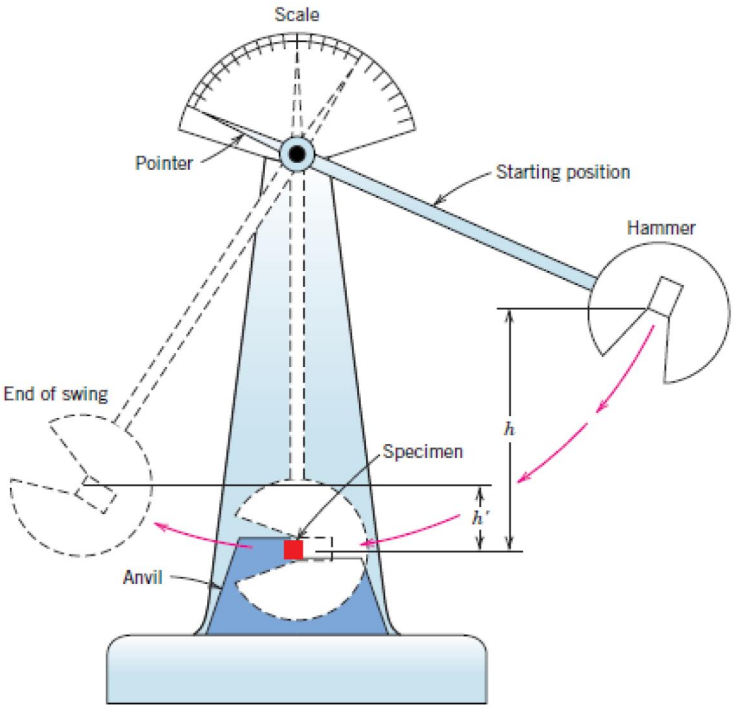


Figure 2.7 Setup of the Charpy impact test [9]

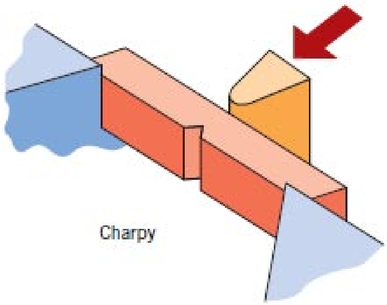


Figure 2.8 Charpy impact test specimen [9]

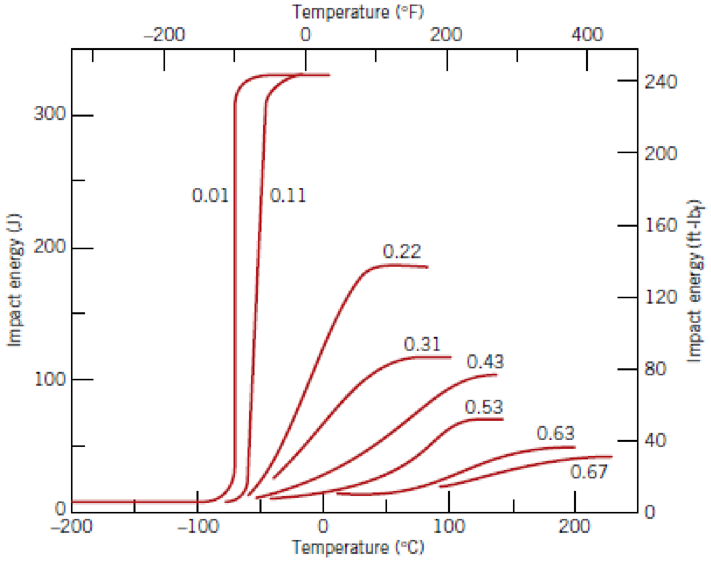


Figure 2.9 Impact energy curves for varying Carbon content [9]

2.4.2 Data Processing

Impact data in the database were available following both V-notch and U-notch test standards. However, since it is not possible to relate results from the different types of test, it was decided to retain the more numerous V-notch test samples.

The resulting dataset contains 1661 samples with each record consisting of 16 input variables and the Charpy energy as output.

The input variables can be grouped in three categories, namely chemical composition, heat treatment conditions and test parameters, as shown in Table 2.3. The variables' statistical properties are listed in Table 2.4. Figure 2.10 displays the sparse modelling data distribution, considering the distribution between the tempering and hardening temperatures as an example.

Table 2.2 Accuracy of measured variables

Variable	Tolerance
Composition	$0.004 \times (wt)^{-0.5}$
Hardening Temperature	$\pm$ 10 °C
Tempering Temperature	$\pm$ 5 °C
Size	$\pm$ 1 mm

Table 2.3 Test variables

Chemical Composition	Heat Treatment Conditions	Test Parameters
Carbon Silicon Manganese Sulphur Chromium Molybdenum Nickel Aluminium Vanadium	Hardening Temperature Cooling Medium Tempering Temperature	Test Depth Specimen Size Test Site Test Temperature

Table 2.4 Charpy impact data statistics

Variable	Units	Name	Range	Mean	Standard Deviation
Test Depth	mm	x_1	5.50 – 146.05	20.80	14.50
Size	mm	x_2	11 – 381	172.49	80.84
Coded Site	-	x_3	2 – 6	3.80	1.12
C	%	x_4	0.13 – 0.52	0.39	0.06
Si	%	x_5	0.11 – 0.38	0.25	0.03
Mn	%	x_6	0.41 – 1.75	0.84	0.22
S	%	x_7	(8.0e-4) – 0.05	0.02	8.9e-3
Cr	%	x_8	0.11 – 3.25	1.08	0.24
Mo	%	x_9	0.02 – 0.98	0.24	0.09
Ni	%	x_{10}	0.03 – 4.21	0.37	0.52
Al	%	x_{11}	(3.0e-3) – 0.05	0.03	4.8e-3
V	%	x_{12}	(1.0e-3) – 0.26	7.7e-3	0.02
Hardening Temperature	°C	x_{13}	810 – 980	864.02	15.47
Cooling Medium	-	x_{14}	1 – 3	2.77	0.52
Tempering Temperature	°C	x_{15}	190 – 730	647.19	49.92
Test Temperature	°C	x_{16}	(-59) – 23	-5.79	26.45
Charpy Energy	J	y	3.47 – 245.33	89.64	32.97

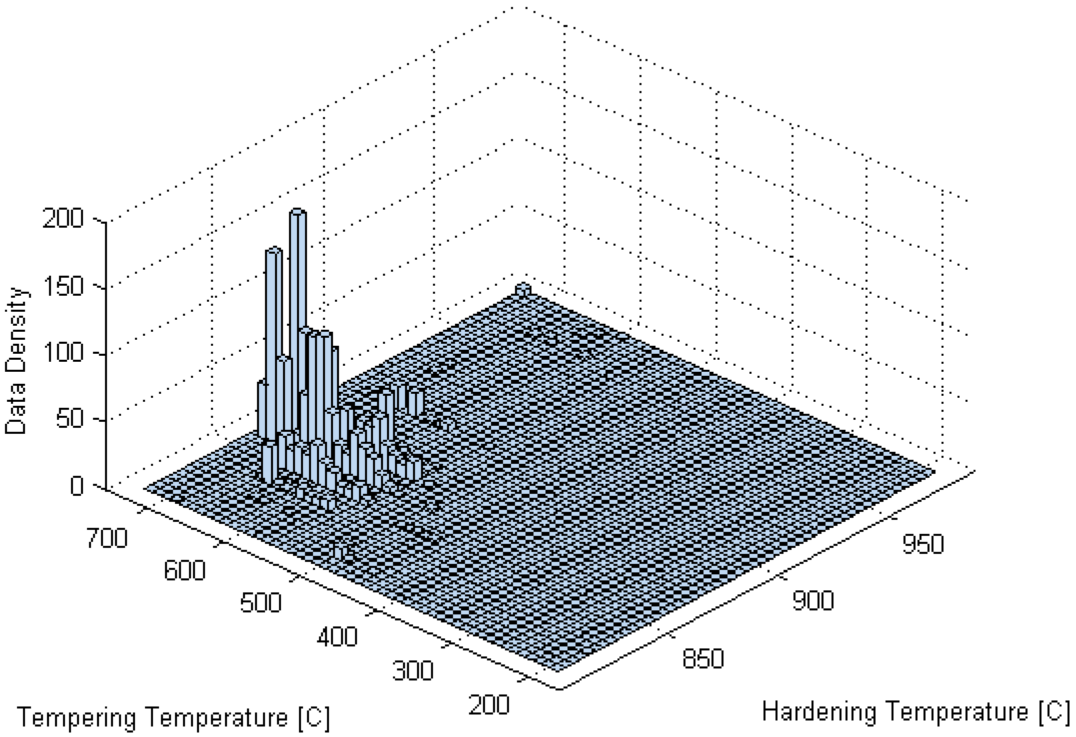


Figure 2.10 Sparse data distribution

2.5 Review of Charpy Impact Data Modelling

This section reviews the various techniques that have been applied to model the dataset being investigated. There are several reasons for which modelling may be useful. These include:

- Prediction of process or product parameters
- Process or product optimisation
- Model-based process control
- Fault detection and quality inspection

Tenner [83] employed an ensemble model made up of 10 Multilayer Perceptron (MLP) neural networks, with each network consisting of 11 neurons in 1 hidden layer.

Panoutsos and Mahfouf [62] used granular computing as a basis for Gaussian membership functions with the width being a function of granule cardinality and size. The neuro-fuzzy structure based on centre of gravity defuzzification, product inference rule and singleton fuzzy output space was optimised using an adaptive back-error-propagation algorithm.

Chen and Linkens [12] investigated neural-fuzzy modelling with knowledge-based modification through incorporation of fuzzy rules.

Panoutsos and Mahfouf [63] further analysed granular computing, using it for data preprocessing. A granular neural-fuzzy ensemble network was elicited, consisting of 13 multiple granularity models (of between 6 and 18 granules).

Mahfouf, Yang and Zhang [48] employed a Bayesian neural network to model the data, with this work focusing of error modelling using a Gaussian Mixture Model (GMM). The prediction confidence interval was further assessed in [95, 97].

Zhang and Mahfouf [102] analysed hierarchical fuzzy modelling for training data selection, modelling and optimisation. The multi-objective optimisation improved accuracy and interpretability by removing redundant rules (using confidence and support measures) and sets, and merging similar ones.

Yang, Mahfouf and Panoutsos [96] used a genetic algorithm to optimise a neural network structure with parameters from the final population providing an ensemble model. The genetic algorithm optimised the number of neurons, activation function, training algorithm and data pre-processing/normalising method.

Rubio-Solis and Panoutsos investigated neutrosophic sets in [80, 74], in conjunction with granular computing neural-fuzzy modelling in the former, and for fuzzy uncertainty assessment of Radial Basis Function (RBF) networks in the latter. In [75], Interval Type-2 Fuzzy Sets were applied to RBF networks.

In the ensuing chapters, modelling results will be compared to the results obtained in the studies reviewed in this section.

2.6 Summary

Firstly, this chapter explained the various stages in the steelmaking process. The next section discussed heat-treating procedures, which give steel the required microstructure properties. Tests that are used to determine the mechanical properties of steel were then outlined. In the final sections, the Charpy impact dataset was presented and research that makes use of this dataset was reviewed.

The next chapter will present the second dataset used in this research, the rail quality data, and then focus on classification techniques from SVMs to classify this binary dataset.

Chapter 3

Fuzzy Support Vector Machine Modelling

3.1 Introduction

Cost management along with production efficiency have become very important at every industry level. Investments in production lines allow companies to improve efficiency and reduce manufacturing costs while quality control procedures improve process output. Modelling techniques are increasingly being employed to understand the interaction and influence of input variables on the process.

Advances in computer processing power, together with the vast amounts of available data, have encouraged the application of machine learning techniques to different real world problems in an attempt to extract useful knowledge from the available information. Pattern classification is a supervised machine learning method in which a labelled set of data points is used to train a model which is then used to classify new test examples. Classifier performance is commonly evaluated by its accuracy. However, this metric does not correctly value the minority class in an imbalanced data set and as a result the trained model tends to be biased towards the majority class [89]. Many data sets from real world problems are inherently imbalanced and therefore appropriate measures need to be taken to ensure that important information due to the minority class is correctly represented by the classifier.

This chapter will discuss some techniques based on SVMs for modelling materials related data [87]. Specifically, a modelling architecture will be applied to a dataset from the testing of rails produced by Tata Steel Europe.

3.2 Rail Manufacturing Data

At Tata Steel Europe, a precisely controlled rail production line, whose sub-processes are indicated in Figure 3.1, produces high quality rails.

During steelmaking, the desired steel chemical composition is achieved. Continuous casting produces steel blooms which are reheated in a furnace and rolled into rail sections. Non-Destructive Testing (NDT) maintains steel integrity, avoids imperfections, and therefore improves product safety such that the rails meet the required standards. Several techniques are employed to inspect the rails for metallurgical and surface flaws, and also to ensure that strict dimensional tolerances are met. Instrumentation systems measure, monitor and control the process variables to improve productivity and product quality.

The dataset obtained from the Tata Steel Europe rail route extended over a two year production period. A process expert provided knowledge about the data which approximated 65,000 records and up to 200 variables. The data was pre-processed with the variables reduced to 70 to make it more manageable for use in modelling.

3.2.1 Variable Selection

Although development in processing power motivated data-driven modelling, data dimensionality still remains an important issue. Reducing the dimensionality improves the performance of the predictor by improving generalisation, and provides faster and more cost-effective predictors by simplifying models and reducing training times [27]. One of the ways to reduce dimensionality is through variable selection. Therefore, variable selection was performed in the work by Yang et al. [98], where 39 inputs were selected from the 70 input variables using

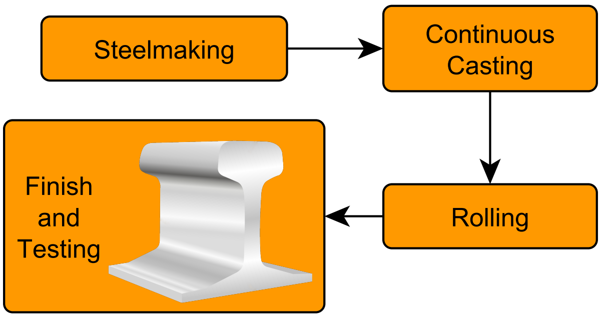


Figure 3.1 Railway production route

an iterative forward selection algorithm. This employed a three-layer MLP neural network as the performance evaluator.

3.2.2 Data Imbalance

The dataset consistsed of 3000 samples which were separated into 60% for training and 40% testing data. The dataset is highly imbalanced with only 6.77% of the data corresponding to rejected rails (Figure 3.2). Details regarding the nature of input variables in the dataset could not be disclosed, so as to protect sensitive information about the rail manufacturing process.

3.3 Theory

3.3.1 Support Vector Machine

The SVM algorithm was proposed by Vapnik and Chervonenkis in the 1960s. It was further developed in 1992 by Boser, Guyon and Vapnik [7] who applied the kernel trick to allow the classification of linearly non-separable data. The soft margin classifier was proposed by Cortes and Vapnik in 1995 [14], allowing some data misclassification when using an SVM.

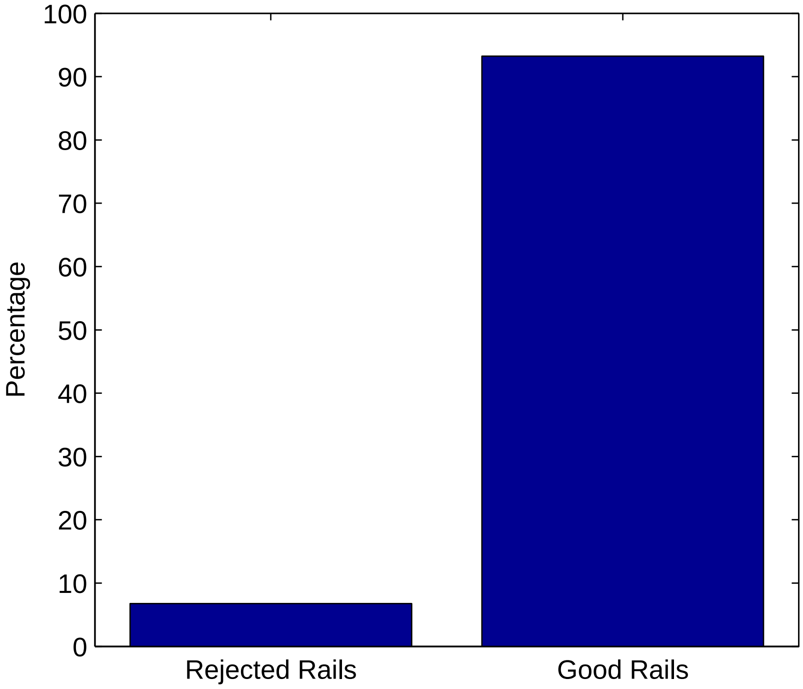


Figure 3.2 Rail quality showing data imbalance

The theory for SVMs is discussed in this subsection [8, 16, 29, 23, 87]. The SVM is a supervised machine learning algorithm. Its aim is to find the optimal hyperplane which separates data from two classes with the maximum margin of separation.

Let the linearly separable training data be labelled $\{\boldsymbol{x}_i, y_i\}$, $y_i \in \{-1, 1\}$, $\boldsymbol{x}_i \in \mathbb{R}^d$, where each input $\boldsymbol{x}_i$ has d dimensions and is in one of two classes y_i.

The hyperplane that separates the two classes can be described as follows:

$$\boldsymbol{w}^T\boldsymbol{x} + b = 0 \tag{3.1}$$

where $\boldsymbol{w}$ is normal to the hyperplane and b is a bias.

Suppose that all the training data satisfy the following constraints:

$$\boldsymbol{w}^T\boldsymbol{x}_i + b \geq +1 \quad \text{for} \quad y_i = +1 \tag{3.2}$$

$$\boldsymbol{w}^T\boldsymbol{x}_i + b \leq -1 \quad \text{for} \quad y_i = -1 \tag{3.3}$$

The inequalities can be combined into the following:

$$y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1 \geq 0 \quad \forall i \tag{3.4}$$

The points for which (3.2) holds with the equality lie on the hyperplane $H_1 : \boldsymbol{w}^T\boldsymbol{x}_i + b = +1$. Its perpendicular distance from the origin is $|1 - b|/||\boldsymbol{w}||$.

Similarly, the points for which (3.3) holds with the equality lie on the hyperplane $H_2 : \boldsymbol{w}^T\boldsymbol{x}_i + b = -1$, whose perpendicular distance from the origin is $|-1 - b|/||\boldsymbol{w}||$.

The points on these parallel hyperplanes on either side of the separating hyperplane are called support vectors and the distance between them is called the margin, such that:

$$\text{margin} = \frac{|(1 - b) - (-1 - b)|}{||\boldsymbol{w}||} = \frac{2}{||\boldsymbol{w}||} \tag{3.5}$$

Therefore, maximising the margin is equivalent to minimising the norm of $\boldsymbol{w}$. The problem is formulated as a convex optimisation as follows:

$$\min \tfrac{1}{2}||\boldsymbol{w}||^2 \quad \text{s.t.} \quad y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1 \geq 0 \quad \forall i \tag{3.6}$$

The problem is switched to a Lagrangian formulation. Thus, positive Lagrange multipliers, α_i, are introduced and the primal Lagrangian is as follows:

$$L_p = \tfrac{1}{2}||\boldsymbol{w}||^2 - \sum_i \alpha_i \left[y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1\right] \tag{3.7}$$

Taking the partial derivatives with respect to $\boldsymbol{w}$ and b and setting them equal to zero:

$$\begin{aligned} \frac{\partial L_p}{\partial \boldsymbol{w}} = 0 \quad &\Rightarrow \quad \boldsymbol{w} = \sum_i \alpha_i y_i \boldsymbol{x}_i \\ \frac{\partial L_p}{\partial b} = 0 \quad &\Rightarrow \quad \sum_i \alpha_i y_i = 0 \end{aligned} \tag{3.8}$$

Substituting back into the primal form of the Lagrangian, the dual formulation can be obtained:

$$L_d = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j \tag{3.9}$$

This is maximised with respect to α_i, subject to the constraints $\alpha_i \geq 0 \; \forall i$ and $\sum_i \alpha_i y_i = 0$. An important property is that the training data only appears in the form of dot products between vectors.

3.3.2 The Non-Separable Case

Since no feasible solution is found when classifying non-separable data, the constraints are relaxed and a cost is introduced in the objective function. Positive slack variables, $\xi_i \geq 0 \; \forall i$, are included in the constraints (3.2) and (3.3) [14], such that:

$$\boldsymbol{w}^T \boldsymbol{x}_i + b \geq +1 - \xi_i \quad \text{for} \quad y_i = +1 \tag{3.10}$$

$$\boldsymbol{w}^T \boldsymbol{x}_i + b \leq -1 + \xi_i \quad \text{for} \quad y_i = -1 \tag{3.11}$$

The objective function is defined as follows:

$$\min\left(\tfrac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_i \xi_i\right) \quad \text{s.t.} \quad y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) - 1 + \xi_i \geq 0 \quad \forall i \tag{3.12}$$

where C is a positive penalty term to allow misclassification.

The dual form of the Lagrangian is the same as (3.9) with the constraints now being $0 \leq \alpha_i \leq C$ and $\sum_i \alpha_i y_i = 0$.

3.3.3 Nonlinear Support Vector Machine

When the relationship between inputs and output is nonlinear, the data is mapped to a higher dimensional space using the function $\boldsymbol{\varphi}(\boldsymbol{x})$ where a separating hyperplane can be found. However, since the algorithm only depends on the data through dot products, Boser, Guyon and Vapnik [7] proposed the kernel trick such that no computations are done explicitly with

$\boldsymbol{\varphi}(\boldsymbol{x})$. Therfore, for a kernel function,

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{\varphi}(\boldsymbol{x}_i) \cdot \boldsymbol{\varphi}(\boldsymbol{x}_j), \tag{3.13}$$

only $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ needs to be used in the training algorithm. An example of such a function is the RBF kernel:

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right) \tag{3.14}$$

The dual form of the Lagrangian is:

$$L_d = \sum_i \alpha_i - \frac{1}{2}\sum_i\sum_j \alpha_i\alpha_j y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{3.15}$$

$$\text{s.t.} \quad \sum_i \alpha_i y_i = 0 \quad \text{and} \quad 0 \leq \alpha_i \leq C \quad \forall i \tag{3.16}$$

The decision rule can also be expressed in terms of the kernel, such that:

$$f(\boldsymbol{x}) = sign\left(\sum_i \alpha_i y_i \boldsymbol{\varphi}(\boldsymbol{s}_i) \cdot \boldsymbol{\varphi}(\boldsymbol{x}) + b\right) = sign\left(\sum_i \alpha_i y_i k(\boldsymbol{s}_i, \boldsymbol{x}) + b\right) \tag{3.17}$$

where $\boldsymbol{s}_i$ are the support vectors.

3.3.4 Fuzzy Support Vector Machine

Apart from dealing with misclassification when the classes are not clearly defined, the SVM might need to take care of noise and outlying samples, to which it is very sensitive [28]. This can be done by providing a membership value, $0 < s_i \leq 1$, for each point in the training dataset and including this membership in the SVM objective function [44]:

$$\min\left(\tfrac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_i s_i\xi_i\right) \quad \text{s.t.} \quad y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1 + \xi_i \geq 0 \quad \forall i \tag{3.18}$$

Thus, if misclassified, an important point with a high membership value will add more weight to the objective function than a misclassified unimportant point.

The dual formulation of the Lagrangian is the same as in (3.15) with a change in the constraints:

$$\sum_i \alpha_i y_i = 0 \quad \text{and} \quad 0 \leq \alpha_i \leq s_i C \quad \forall i \tag{3.19}$$

As proposed by Lin and Wang [44], the FSVM can reduce the effect of outliers by using a fuzzy membership which is a function of the distance between a point and its class centre. Let the fuzzy membership be a function of the mean and radius of each class such that a point close to the centre has a high membership while a point far away from the centre has a low membership:

$$s_i = \begin{cases} 1 - |\boldsymbol{x}_+ - \boldsymbol{x}_i|/(r_+ + \delta) & \text{if } y_i = +1 \\ 1 - |\boldsymbol{x}_- - \boldsymbol{x}_i|/(r_- + \delta) & \text{if } y_i = -1 \end{cases} \tag{3.20}$$

where $\boldsymbol{x}_+$ is the mean and r_+ is the radius of class $y = +1$, $\boldsymbol{x}_-$ is the mean and r_- is the radius of class $y = -1$, $\delta > 0$ to avoid $s_i = 0$.

3.3.5 Support Vector Machines for Class Imbalance

Preliminary results showed that SVMs are sensitive to class imbalance. Batuwita and Palade [4] review methods in the literature used to reduce the problem that models are biased towards the majority class and have low performance on the minority class. These cover external methods such as data pre-processing and internal methods that make algorithmic modifications to the SVM algorithm.

An internal imbalance learning method was applied by modifying the SVM objective function and assigning two misclassification costs. This is referred to as DEC by Batuwita and Palade [4] and was originally proposed by Veropoulos et al. [88]. Thus C^+ is the misclassification cost for the positive (minority) class while C^- is the misclassification cost for the negative (majority) class. When combined with the fuzzy membership defined earlier, the following objective function is obtained:

$$\min\left(\frac{1}{2}||\boldsymbol{w}||^2 + C^+C \sum_{\{i|y_i=+1\}} s_i\xi_i + C^-C \sum_{\{i|y_i=-1\}} s_i\xi_i\right) \tag{3.21}$$

$$\text{s.t.} \quad y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1 + \xi_i \geq 0 \quad \forall i \tag{3.22}$$

The dual Lagrangian form, L_d, of this function is the same as in (3.15) and is maximised subject to the following constraints:

$$\textstyle\sum_i \alpha_i y_i = 0, \;\; 0 \leq \alpha_i^+ \leq s_i C^+ C, \;\; 0 \leq \alpha_i^- \leq s_i C^- C \quad \forall i \tag{3.23}$$

where α_i^+ and α_i^- represent the Lagrange multipliers of positive and negative samples respectively.

The ratio C^-/C^+ was set equal to the minority to majority class ratio as suggested in [2] such that the penalty for misclassifying minority class examples is higher.

It was subsequently decided to apply an external imbalance learning method by balancing the training data. The dataset obtained by Zughrat et al. [103] was used, where under-sampling of the majority class was performed to make the number of good rails equal to the rejected rail training examples.

3.3.6 Fuzzy C-Means Clustering

The number of support vectors when applying SVMs and FSVMs on the balanced data set was still very high leading to long training times and making parameter optimisation impractical and inefficient. Therefore Fuzzy C-Means (FCM) clustering was proposed as a way of reducing the number of support vectors, reducing training times and model complexity, and improving generalisation [94, 10].

The FCM clustering algorithm is an optimisation problem whereby the coordinates of the cluster centres need to be identified. The cost function to be minimised [5] is:

$$J(X,U,V) = \sum_{i=1}^{c}\sum_{k=1}^{N} \mu_{ik}^{m} \|\boldsymbol{x}_k - \boldsymbol{v}_i\|^2 \tag{3.24}$$

where $V = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_c]$ is the vector of cluster centres, $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N]$ represents the data samples, $U = [\mu_{ik}]$ is the fuzzy partition matrix of X, m is the weighting exponent, ${D_{ik}}^2 = \|\boldsymbol{x}_k - \boldsymbol{v}_i\|^2$ is the squared distance norm.

The minimisation of (3.24) is possible if and only if:

$$\mu_{ik} = \frac{1}{\sum_{j=1}^{c}\left(\frac{D_{ik}}{D_{jk}}\right)^{\frac{2}{m-1}}} \qquad 1 \le i \le c \quad 1 \le k \le N \tag{3.25}$$

$$\boldsymbol{v}_i = \frac{\sum_{k=1}^{N} \mu_{ik}^{m} \boldsymbol{x}_k}{\sum_{k=1}^{N} {\mu_{ik}}^{m}} \qquad 1 \le i \le c \tag{3.26}$$

The membership degree, μ_{ik}, is inversely proportional to the squared distance from the data points to the current cluster centres. Equation (3.26) gives $\boldsymbol{v}_i$ as the weighted mean of the data points, where the weights are the membership degrees. The FCM algorithm iterates through (3.25) and (3.26) to optimise the fuzzy partition matrix and cluster centres.

3.3.7 The Confusion Matrix

In binary classification accuracy is usually the metric used to assess a classifier's predictive performance. Using the confusion matrix of Table 3.1, accuracy can be defined as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{3.27}$$

Table 3.1 Confusion matrix

		Actual Rail Quality	
		Rejected	Good
Predicted Rail Quality	Rejected	True Positive (TP)	False Positive (FP)
	Good	False Negative (FN)	True Negative (TN)

Accuracy is the number of correct classifications from both classes. However, for an imbalanced dataset, a classifier may be skewed towards the majority class and therefore sensitivity and specificity offer a better description of the classifier's performance.

Considering the imbalanced rail dataset, the terms in Table 3.1 can be explained as:

- True Positive (TP): Rejected rails correctly classified as rejected rails
- True Negative (TN): Good rails correctly classified as good rails
- False Positive (FP): Good rails incorrectly classified as rejected rails
- False Negative (FN): Rejected rails incorrectly classified as good rails

Using these terms, sensitivity and specificity are defined as follows:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP}+\text{FN}} = \frac{\text{TP}}{\text{Number of Positives}} \tag{3.28}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN}+\text{FP}} = \frac{\text{TN}}{\text{Number of Negatives}} \tag{3.29}$$

3.4 Results

3.4.1 Modelling Imbalanced Dataset using the Support Vector Machine Techniques

A grid search was performed to optimise the parameters of the SVM and FSVM models as shown in Figure 3.3 and Figure 3.4 respectively. This was done to optimise the penalty term, C, and RBF kernel spread, σ. The results obtained, with a sensitivity of approximately 24% as indicated in Table 3.2, show that SVM techniques are sensitive to class imbalance with the result being that the models overfit the training data leading to poor generalisation.

3.4.2 Modelling using Class Imbalance Learning Methods

For the DEC-FSVM model, a grid search was performed for the parameters C and σ (Figure 3.5). Considering a model with the best sensitivity (without degrading specificity) using a C of 3,162 and σ of 251, an accuracy of 68.1%, sensitivity of 53.0% and specificity of 69.2% were obtained (Table 3.3).

Figure 3.6 and Figure 3.7 show the grid search results for the SVM and FSVM when applied on the balanced dataset. Models were chosen from the region with best sensitivity with C equal to 11,776 and σ equal to 46.45 for both models. The results are tabulated in Table 3.3 with the FSVM offering a 9.06% improvement in terms of sensitivity (Table 3.4). However, the ratio of support vectors to the number of training points was still very high at 0.952.

3.4.3 Fuzzy C-Means Clustering, Fuzzy Support Vector Machine

Clustering was performed on the balanced dataset which had 2877 points. Random initial cluster centres and a weighting exponent, m, of 2 were used for the FCM algorithm. Using the same values for the parameters C and σ, the average performance of 10 FSVM models was considered for every clustering level from 10% to 90% (with 10% having the least number of cluster centres resulting in the minimum number of training points). The models where tested on a separate unbalanced data set. Table 3.5 shows that the number of support vectors was reduced since clustering reduces the number of training points ($R_{sv/max}$). However, clustering grouped points with similar features and this allowed the SVM algorithm to further reduce the number of support vectors in relation to the number of training points available ($R_{sv/tr}$). Figure 3.8 indicates that classifier performance is generally a compromise between sensitivity and specificity. Figure 3.9 shows that after clustering and fuzzification, the SVM algorithm was able to build the model using 50% or less of the available training points ($R_{sv/tr}$). This highly reduced the model training time which is especially important for parameter optimisation procedures. Analysing performance, one has to keep in mind the data dimensionality and the high nonlinearities in the input-output relationship.

3.5 Summary

This chapter aimed at designing a modelling architecture to deal with the imbalanced data relating to the production of rails. The modelling techniques are based on SVMs which are not affected by local minima as they are mathematically based on the solution of a convex optimisation problem. Also, in some cases, SVM generalisation performance has

Table 3.2 Performance of SVM and FSVM models on imbalanced dataset

	SVM	FSVM
Sensitivity [%]	24.10	24.10
Specificity [%]	93.30	93.10
Accuracy [%]	88.50	88.30
Support Vector Ratio $(R_{sv/tr})$	0.24	0.25

Table 3.3 Model performance for class imbalance learning methods

Imbalance Learning Method	Internal	External	
	FSVM	SVM	FSVM
Sensitivity [%]	53.00	61.94	67.55
Specificity [%]	69.20	73.80	68.24
Accuracy [%]	68.10	73.06	68.20
Support Vector Ratio $(R_{sv/tr})$	0.98	0.95	0.95

Table 3.4 Performance comparison of SVM and FSVM models

	SVM	FSVM	Percentage Difference
Sensitivity [%]	61.94	67.55	+ 9.06
Specificity [%]	73.80	68.24	– 7.53
Accuracy [%]	73.06	68.20	– 6.65
Support Vector Ratio $(R_{sv/tr})$	0.948	0.952	+ 0.42

Table 3.5 FCM-FSVM with different clustering levels

Clustering	Sensitivity	Specificity	Accuracy	$R_{sv/tr}$	$R_{sv/max}$
[% of max. # pts.]	[%]	[%]	[%]		
10	71.31	50.60	51.89	0.43	0.04
20	69.46	57.08	57.80	0.50	0.10
30	69.73	57.71	57.44	0.51	0.15
40	68.30	60.10	60.61	0.48	0.19
50	68.21	60.78	61.25	0.49	0.24
60	68.75	60.71	61.21	0.47	0.28
70	66.09	63.31	63.48	0.46	0.32
80	66.08	63.37	63.54	0.42	0.34
90	66.37	63.75	63.91	0.43	0.39

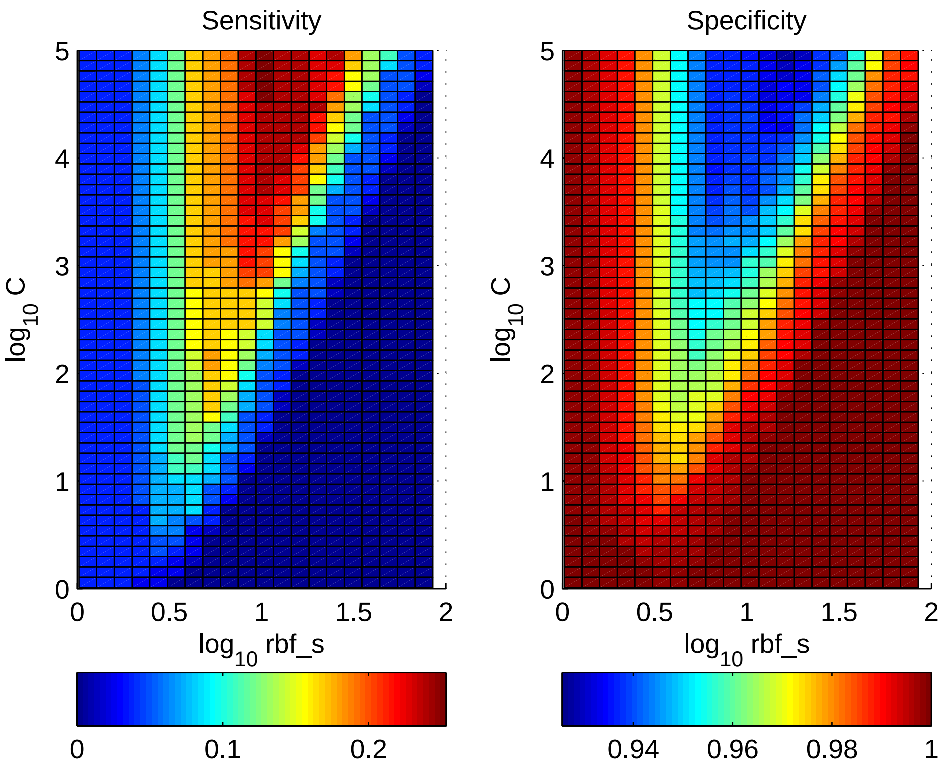


Figure 3.3 Grid search for SVM with RBF kernel

been shown to be better than that of other classification methods [8]. However, SVMs are sensitive to class imbalance. Therefore, an internal (FSVM with DEC) and external (data under-sampling) class imbalance learning methods were applied to the data. The performance of the techniques when implemented on the under-sampled dataset was better, while in both cases the inclusion of a fuzzy membership improved the performance of the SVM. Quadratic optimisation scales poorly with the number of data samples and therefore FCM clustering was proposed for reducing training time and model complexity, by reducing the number of support vectors of the FSVM model, concluding it is effective in reducing model complexity without any significant performance deterioration.

The following chapter will discuss the introduction of a new type of fuzzy membership function for materials property prediction of steel.

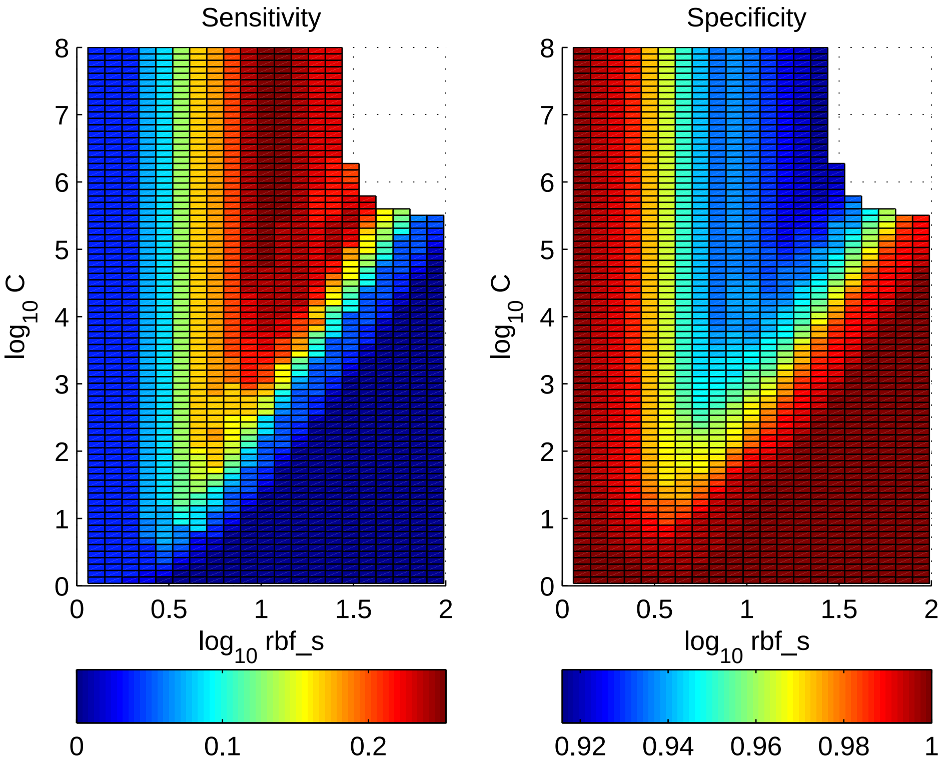


Figure 3.4 Grid search for FSVM with RBF kernel

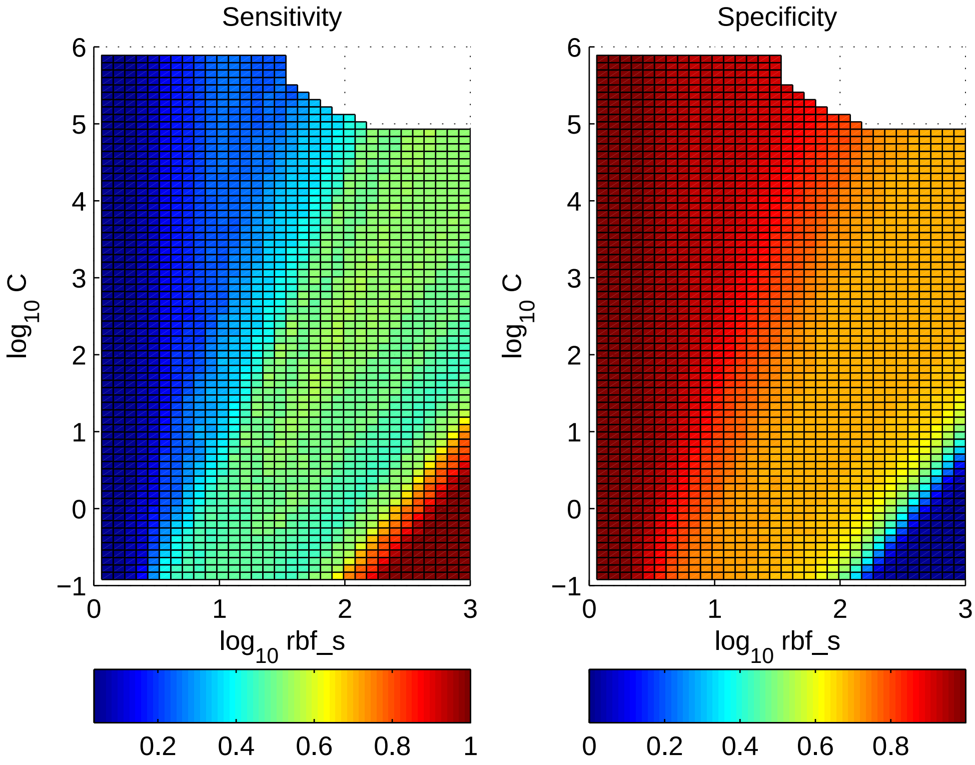


Figure 3.5 Grid search for FSVM with DEC

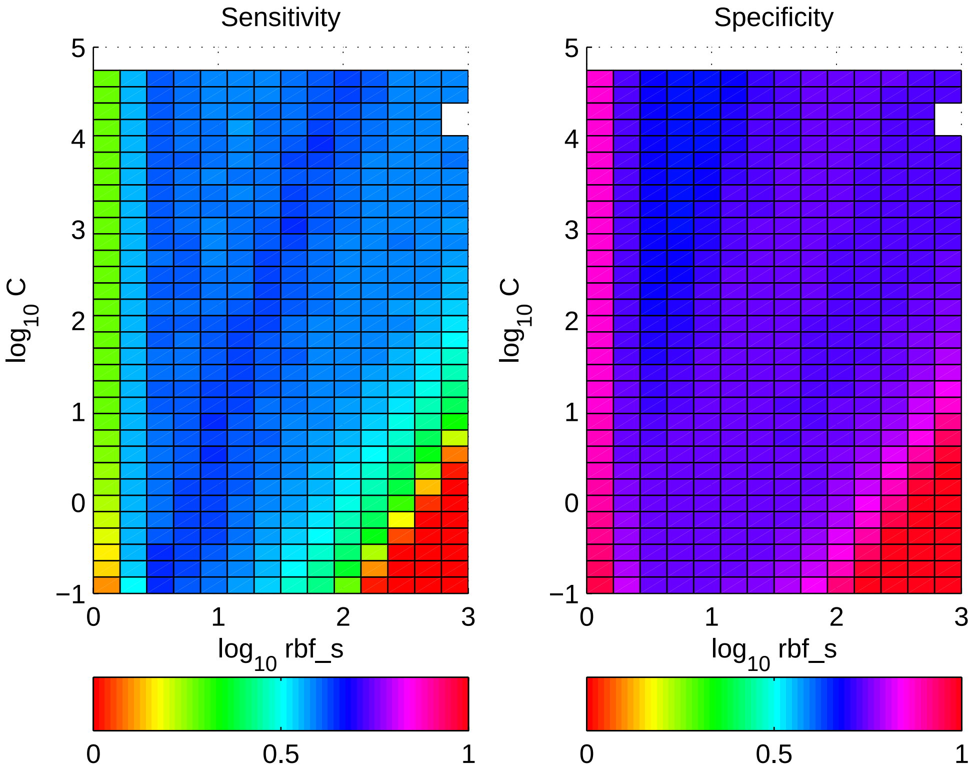


Figure 3.6 Grid search for SVM on balanced dataset

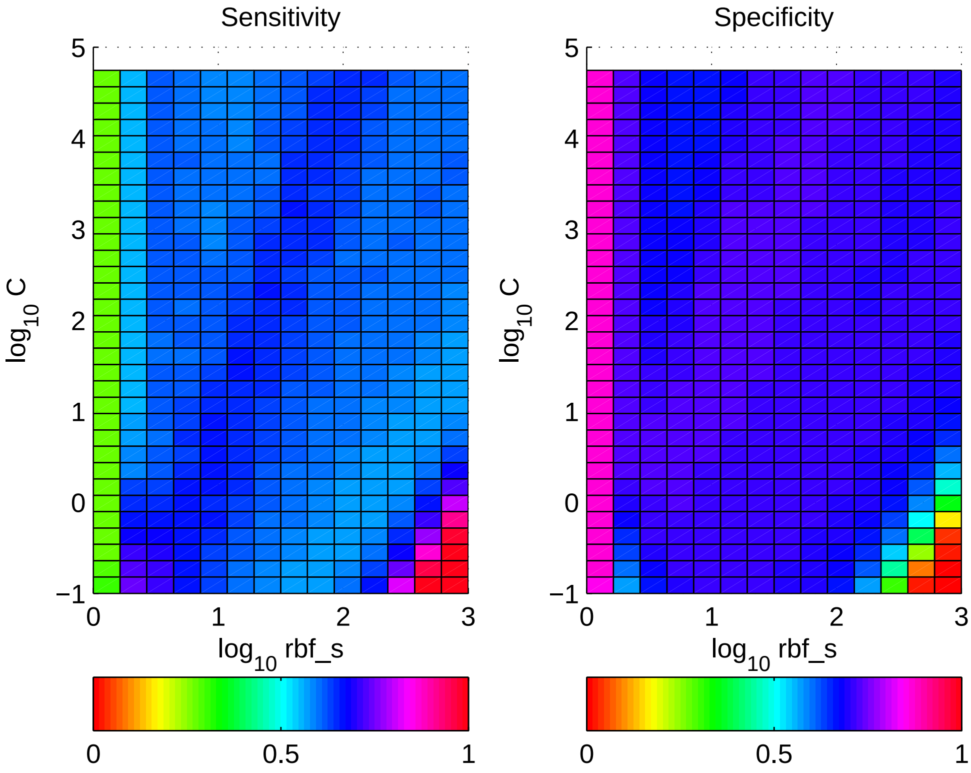


Figure 3.7 Grid search for FSVM on balanced dataset

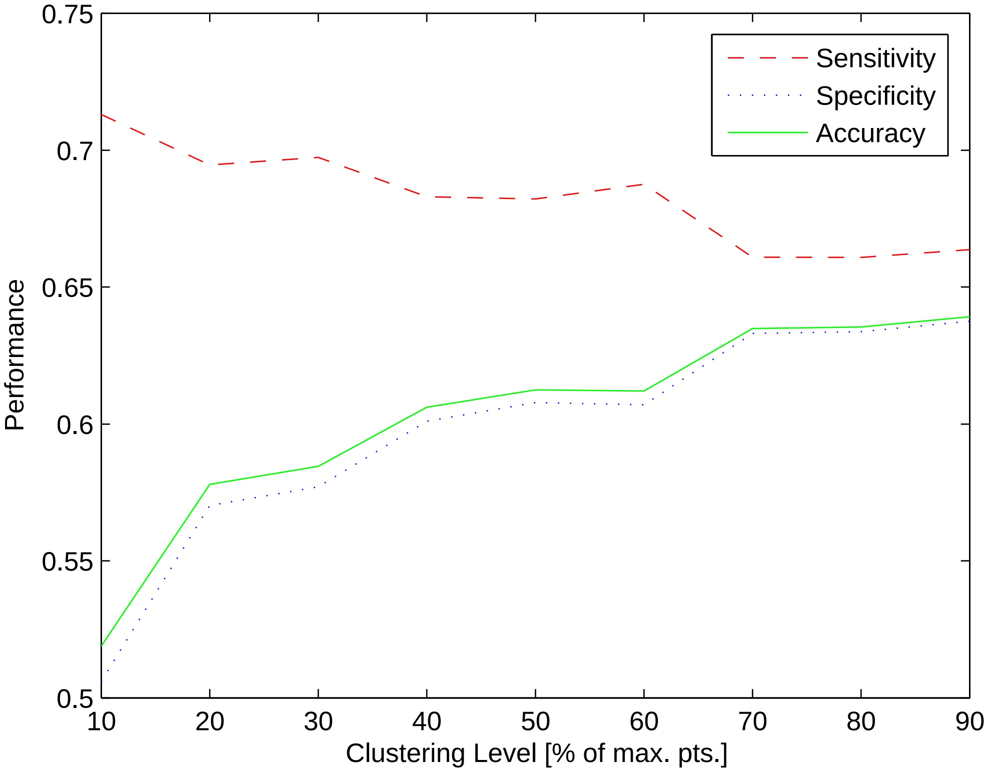


Figure 3.8 Performamce for different clustering levels

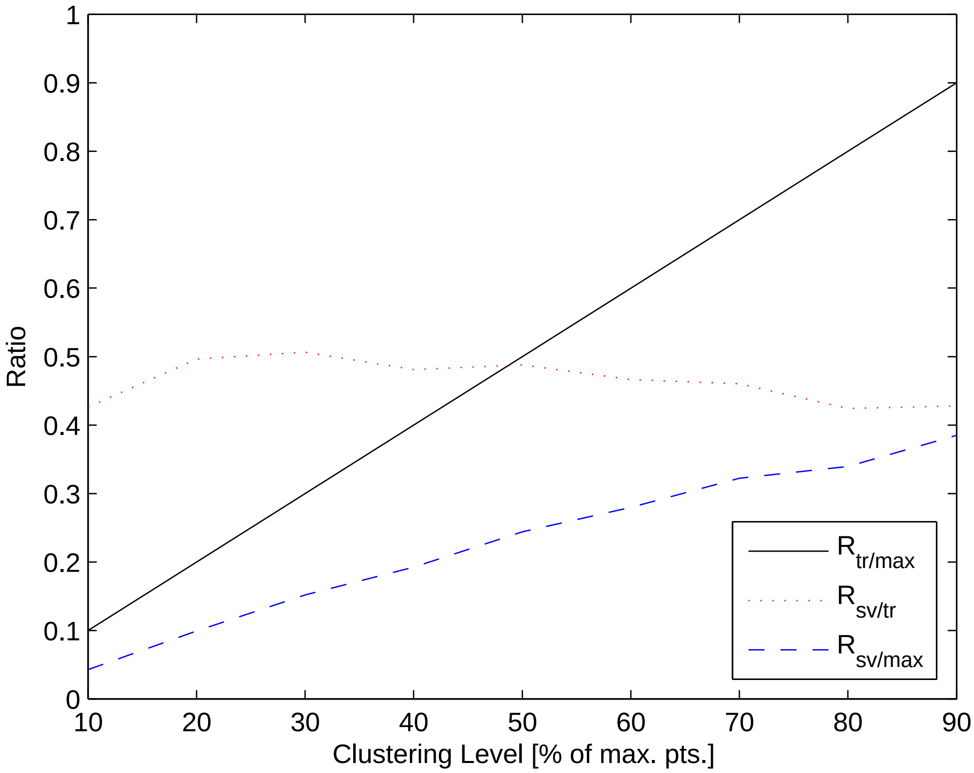


Figure 3.9 Support vector ratios for different clustering levels

Chapter 4

Quantum-Membership-Function-based Fuzzy Modelling with Application to Charpy Energy

4.1 Introduction

Classification and recognition are intrinsic human abilities which we use extensively in everyday life. The formal foundations of classification can be traced back to the ancient world of Plato and Aristotle [20]. One of the first examples of mathematical predictions can be accredited to Johann Carl Friedrich Gauss who in 1801 predicted the position of Ceres based on previous observations of the asteroid by the astronomer Giuseppe Piazzi [82]. In recent years, the use of computers helped to merge generally overlapping fields such as pattern classification and machine learning, and develop areas within them such as supervised, unsupervised and reinforcement learning into evolving research themes.

4.2 Artificial Neural Networks

One of the most popular machine learning algorithms are multilayer feed-forward neural networks. The foundations of neural networks were laid in the 20^{th} century, with noted works being the propositions of nervous activity [50], and the development of a learning procedure for the first hardware neural network [90]. The MLP neural network obtained its name from Rosenblatt's model [72]. In 1986, Rumelhart et al. [76] extended the learning rule mentioned previously to multiple layers, thus developing the error back-propagation algorithm, which still forms the basis to more advanced network learning techniques. The popularity of feed-forward neural networks increased as processing power became more available and also because, as explained in [6], various studies show that given suitable parameters they are universal function approximators. This means that neural networks can approximate complex systems using simple topologies. In fact, the universal approximation theorem [29, p. 167] states that, although maybe not in an optimum sense, a single hidden layer of neurons is sufficient to approximate a given function.

4.3 Fuzzy Logic

Fuzzy logic was introduced by L.A. Zadeh in 1965 [99]. Two perspectives that offer an insight to the advantages of fuzzy logic are: ‘to exploit the tolerance for imprecision’ and ‘the principle of incompatibility’. The former implies that there is no need for a model to be more complex and precise than actually needed. The latter implies that a simpler model leads to a more interpretable system, having more significance to the underlying process.

The principle of incompatibility states that:

> “As the complexity of a system increases, our ability to make precise and yet significant statements about its behaviour diminishes until a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive characteristics.” [100]

Mathematical models may be ineffective on their own for processes which are characterised by uncertainties, nonlinearities and unmodelled dynamics and therefore fuzzy logic provides a more interpretable approach to uncertainty. For complex systems, the interpretability of fuzzy logic also allows expert knowledge to be more easily incorporated in the system since the inference mechanism comprises understandable and intuitive IF-THEN rules.

As noted in [41], it was proven in multiple works that a fuzzy system representation can approximate functions to any degree of accuracy. Thus, fuzzy systems are universal approximators.

4.3.1 Fuzzy Sets

A crisp or binary set, A, identifies elements as either part of it or not, such that their membership is:

$$\mu_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases} \tag{4.1}$$

A fuzzy set generalises the idea of a crisp set such that elements, x, from the universe of discourse, X, are mapped to the interval $[0,1]$. Mathematically, a fuzzy set can be represented by a set of ordered pairs of elements and their degree of membership in the fuzzy set A:

$$A = \left\{ \left(x, \mu_A(x)\right) \mid \forall x \in X, 0 \leq \mu_A(x) \leq 1 \right\} \tag{4.2}$$

The membership value represents the degree of truth by which an element belongs to the set. The mapping between the universe of discourse and the membership values forms the membership function. For example, the fuzzy set 'Warm Weather' can be described by the membership function in Figure 4.1.

4.3.2 Type-1 Fuzzy Logic System

Figure 4.2 shows the general architecture of a Fuzzy Logic System (FLS) which consists of the following components [66]:

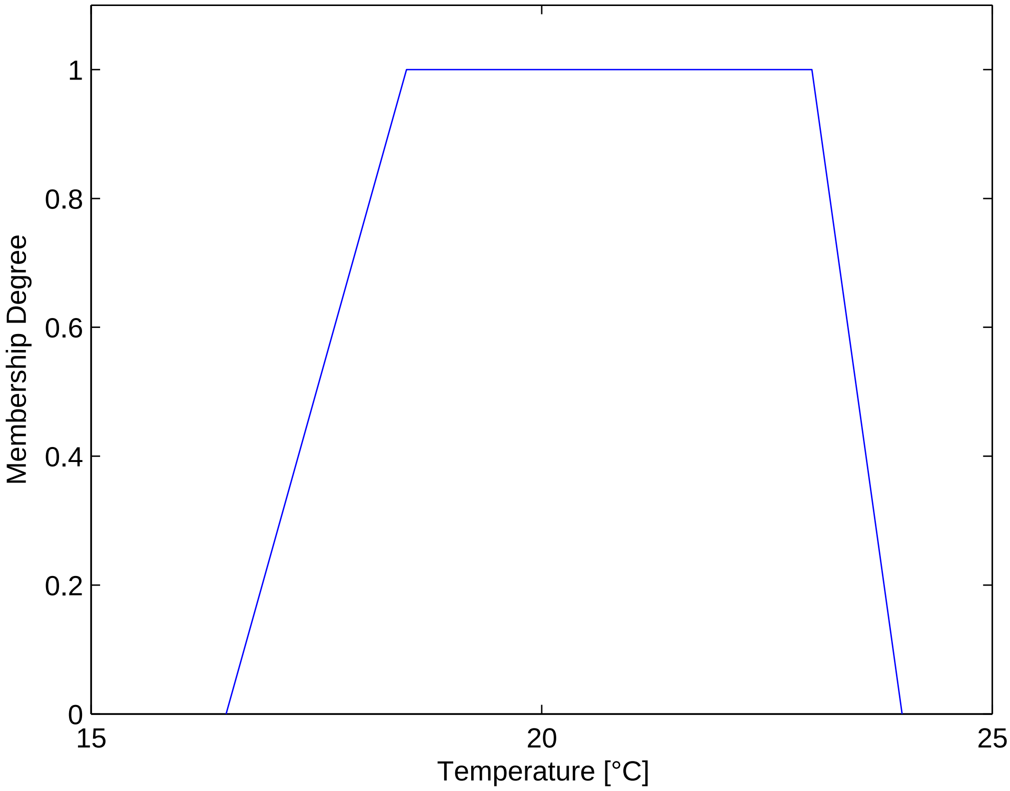


Figure 4.1 Fuzzy set describing 'Warm Weather'

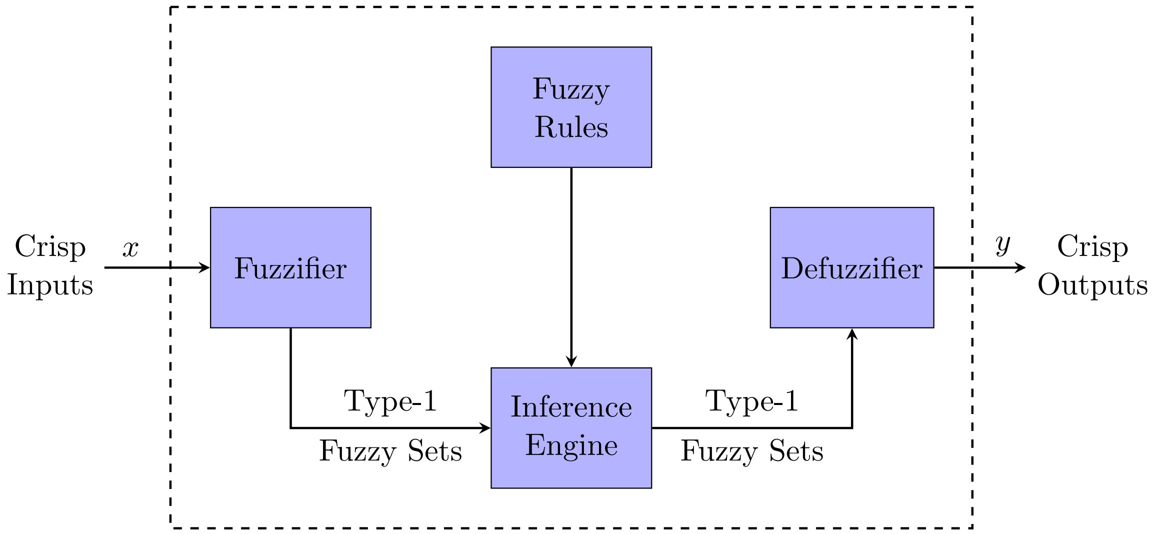


Figure 4.2 Type-1 Fuzzy Logic System

- Fuzzifier - Converts the crisp inputs into fuzzy sets.
- Fuzzy Rules - A rule base of fuzzy IF-THEN rules which provide a linguistic description of the output in terms of the input variables. The IF part (associated with inputs) is called the antecedent and the THEN part (associated with the output) is referred to as the consequent.
- Inference Engine - Combines the input fuzzy sets with the fuzzy rules using fuzzy reasoning to produce the output fuzzy sets.
- Defuzzifier - Converts the output fuzzy sets into a crisp output using one of various defuzzification methods.

The two most common inference mechanisms which were successfully applied to a wide range of applications are the Mamdani and Sugeno (or Takagi, Sugeno and Kang – TSK) methods [73]. The main difference between these two methods is that in the Mamdani method the consequent parts are fuzzy, while in the Sugeno type the consequents are mathematical functions.

4.4 Adaptive-Network-based Fuzzy Inference System

The ANFIS combines a fuzzy inference system with adaptive networks for nonlinear modelling [35].

The ANFIS architecture is shown in Figure 4.3 and its layers, which will be explained in Section 4.5.2, are as follows [36]:

Layer 1 - Membership Layer

Layer 2 - Intersection Layer

Layer 3 - Normalisation Layer

Layer 4 - Consequent Layer

Layer 5 - Output Layer

Regarding the notation used in the diagram, x and y are inputs, A_i and B_i are fuzzy sets, w_i are the multiplication layer outputs, $\overline{w}_i$ are the normalisation layer outputs, f_i are linear functions of the inputs and f is the model output.

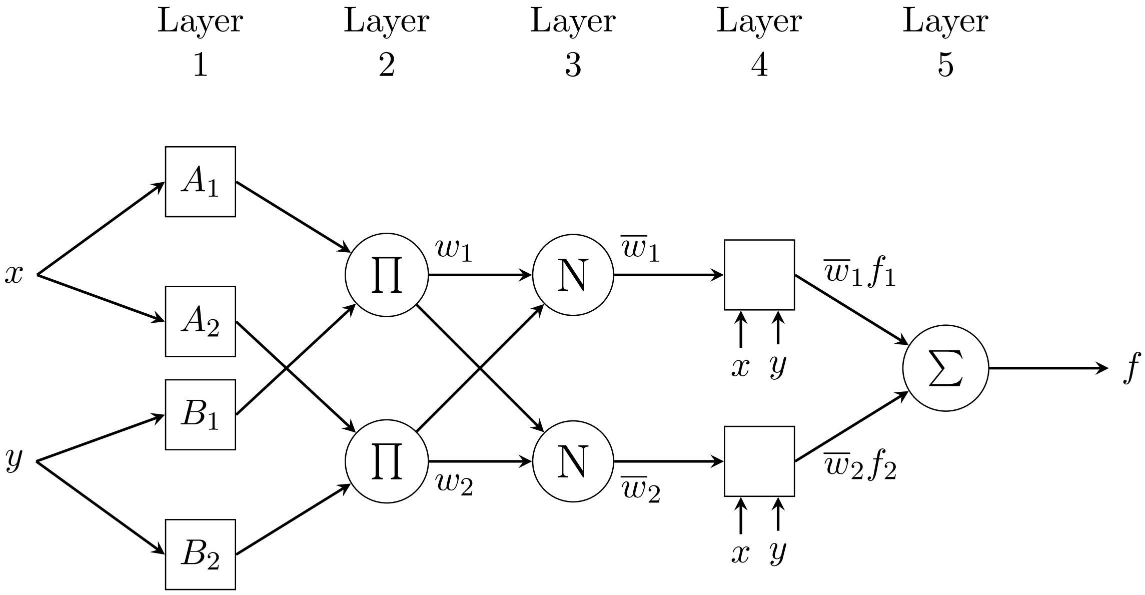


Figure 4.3 ANFIS architecture

The main advantage of this hybrid architecture is that the fuzzy inference mechanism, which allows human knowledge and reasoning to describe the system, is embedded into the structure of neural networks, providing the supervised computational learning abilities of adaptive networks.

4.5 Quantum Neuro-Fuzzy Inference System

This section presents the components of the proposed modelling technique with the model deriving its structure from the ANFIS architecture. The quantum membership function is proposed as a means of improving the uncertainty handling capabilities of the model.

4.5.1 Quantum Membership Function

Quantum membership functions have been employed in modelling problems, obtaining good classification accuracies [45, 46]. The quantum function was also considered as the activation function in neural networks [69, 42]. These studies indicate that quantum neural networks are able to model uncertainty by capturing the inherent structure of the data.

The quantum function is characterised by the sum of a number of sigmoid functions, depending on the number of quantum levels. The sigmoid functions are shifted along the universe of discourse by the quantum intervals, resulting in multileveled membership functions. A quantum membership function is defined as [45]:

$$\mu_A(x) = \frac{1}{n_\theta}\sum_{r=1}^{n_\theta}\left[\left(\frac{1}{1+\exp(-\beta(x-c+|\theta^r|))}\right)U(x;-\infty,c) + \left(\frac{\exp(-\beta(x-c-|\theta^r|))}{1+\exp(-\beta(x-c-|\theta^r|))}\right)U(x;c,\infty)\right] \tag{4.3}$$

where x is the input, $\mu_A(x)$ is the membership degree of x for fuzzy set A, β is the slope factor, θ^r is the quantum interval, c is the membership function centre, n_θ is the number of quantum levels and

$$U(x;a,b) = \begin{cases} 1 & \text{if } a \leq x < b, \\ 0 & \text{otherwise.} \end{cases} \tag{4.4}$$

Figure 4.4 illustrates the membership degree given by a three-level ($n_\theta = 3$) quantum membership function with $c = 0$, $\beta = 2$, and $\theta^r = [10, 20, 30]$. Figure 4.5 shows a quantum membership function extended to 2 input dimensions.

The advantages of employing the quantum membership function in highly uncertain modelling scenarios are:

- A quantum set offers better generalisation through a different definition of subjectivity which would normally require multiple sets.

- A quantum membership function captures and quantifies the structure of the input space.

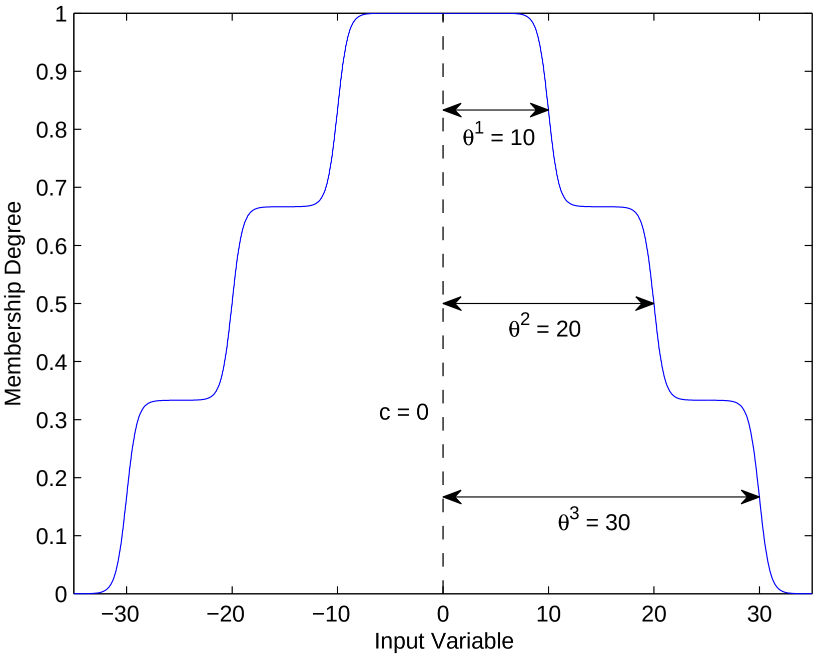


Figure 4.4 3-level Quantum membership function

- The underlying data distribution can be represented by ‘packets’ (quanta) of similar points by the same membership degree for the particular quantum interval (level).
- The nature of the membership function having layers with the same membership degree helps to deal with outlying data points more effectively.
- Uncertainty in the data is detected and modelled by the quantum intervals which also offer another degree of freedom that can be optimised along with the other parameters.

4.5.2 Modelling Architecture

The modelling structure is based on the ANFIS architecture and as shown in Figure 4.6, it is similar to the type-3 ANFIS [35] with a TSK method of fuzzy rule inference. The fuzzy IF-THEN rules are of the form:

$$R_j: \quad \text{IF } x_1 \text{ is } A_{1j} \textit{ and } x_2 \text{ is } A_{2j} \ \ldots \text{ and } x_n \text{ is } A_{nj} \text{ THEN } y_j \text{ is } b_j + \sum_{i=1}^{n} a_{ij}x_i \tag{4.5}$$

where x_i is the input variable, y is the output, A_{ij} is the linguistic quantum fuzzy set of the antecedent part with membership degree $\mu_{A_{ij}}$, b_j and a_{ij} are the consequent parameters, n is the input dimensionality, and R_j is the j^{th} fuzzy rule.

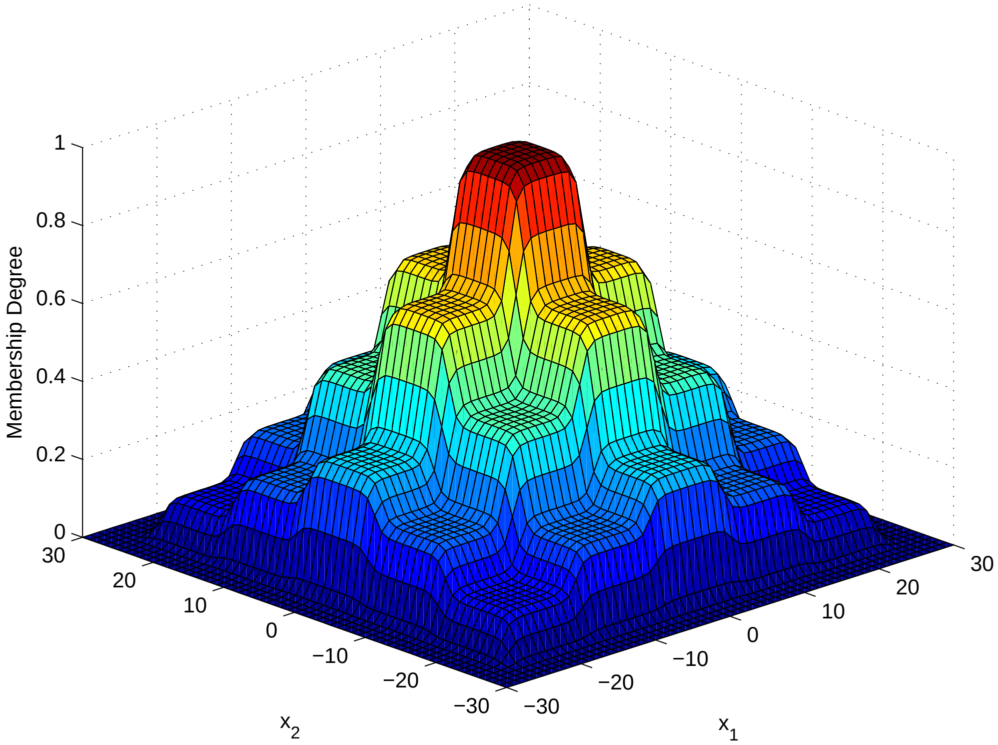


Figure 4.5 Quantum membership function with 2 input dimensions

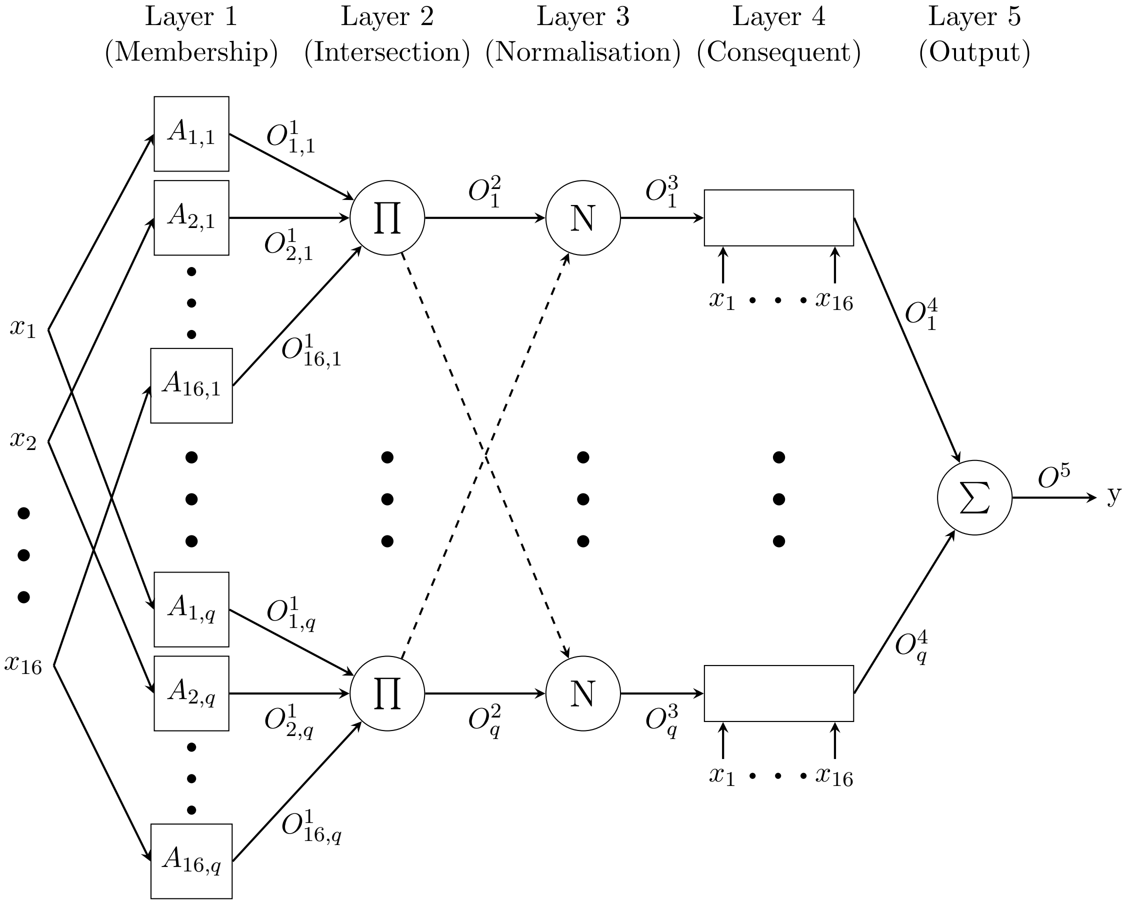


Figure 4.6 Quantum neuro-fuzzy inference system modelling architecture

Let q represent the number of fuzzy rules and O^l denote the output of a node in the l^{th} layer. The operations performed in each of the layers are:

Layer 1 (Membership) - The membership degree of quantum membership sets defining the linguistic variables. The number of linguistic variables for every input dimension is equal to the number of fuzzy rules which is also equal to the number of clusters (the clustering algorithm will be explained further on). The output of this layer is:

$$O^1_{ij} = \mu_{A_{ij}}(x_i) \tag{4.6}$$

Layer 2 (Intersection) - Expresses the ‘AND’ between premises (antecedents) which is performed through a multiplication. A firing strength for each rule is produced. An output from this layer is qiven by:

$$O^2_j = \prod_i O^1_{ij} \tag{4.7}$$

Layer 3 (Normalisation) - The ratio of the j^{th} rule firing strength to the sum of all rules' firing strengths:

$$O^3_j = \frac{O^2_j}{O^2_1 + O^2_2 + \ldots + O^2_q} \tag{4.8}$$

Layer 4 (Consequent) - The Sugeno processing rule:

$$O^4_j = O^3_j \left(b_j + \sum_{i=1}^{n} a_{ij} x_i \right) \tag{4.9}$$

Layer 5 (Output) - Rule aggregation which is performed by summing the output from all rules:

$$O^5 = \sum_{j=1}^{q} O^4_j \tag{4.10}$$

4.5.3 Clustering

Clustering is used to determine the number of quantum sets per input dimension and consequently the number of fuzzy rules. It also provides an initial estimate for the membership function centres and quantum intervals.

The algorithm employed performs one pass through the data and allocates the points to a cluster based on the distance between the point and the cluster centres [46]. If the smallest distance to a centre is larger than a certain threshold, a new cluster is formed. A flowchart of the clustering procedure is shown in Figure 4.7. The size of each cluster is monitored and later used to define the quantum intervals using the formula:

$$\theta_{ij}^{r} = \frac{r \cdot D_j}{\left(n_{\theta_{ij}} + 1\right)/2} \tag{4.11}$$

where $r = 1, \ldots, n_\theta$, $j = 1, \ldots, q$, and D_j is the diagonal distance of the cluster.

4.5.4 Parameter Optimisation

The parameters are updated by tuning the cost function along the negative gradient to achieve supervised learning based on the error back-propagation algorithm. This is used to update the consequent parameters, b_j and a_{ij}, the membership function centres, c_{ij}, and the quantum intervals, θ_{ij}^{r}. Let the cost function (for the case of a single output) be defined as:

$$E = \frac{1}{2} e^T \cdot e \tag{4.12}$$

where $e = y - y_t$, y is the predicted output and y_t is the target output value.
The error term to be back-propagated is described by:

$$\delta_e = -\frac{\partial E}{\partial y} = y_t - y = -e \tag{4.13}$$

The consequent parameter updates are:

$$\begin{aligned} \Delta b_j &= -\frac{\partial E}{\partial b_j} = \frac{\delta_e O_j^3}{\sum_{j=1}^{q} O_j^3} \\ \Delta a_{ij} &= -\frac{\partial E}{\partial a_{ij}} = \frac{\delta_e O_j^3 x_i}{\sum_{j=1}^{q} O_j^3} \end{aligned} \tag{4.14}$$

The consequent parameters are updated using:

$$\begin{aligned} b_j(k+1) &= b_j(k) + \eta_w \Delta b_j \\ a_{ij}(k+1) &= a_{ij}(k) + \eta_w \Delta a_{ij} \end{aligned} \tag{4.15}$$

where η_w is the network weight parameter learning rate and k is the time step.

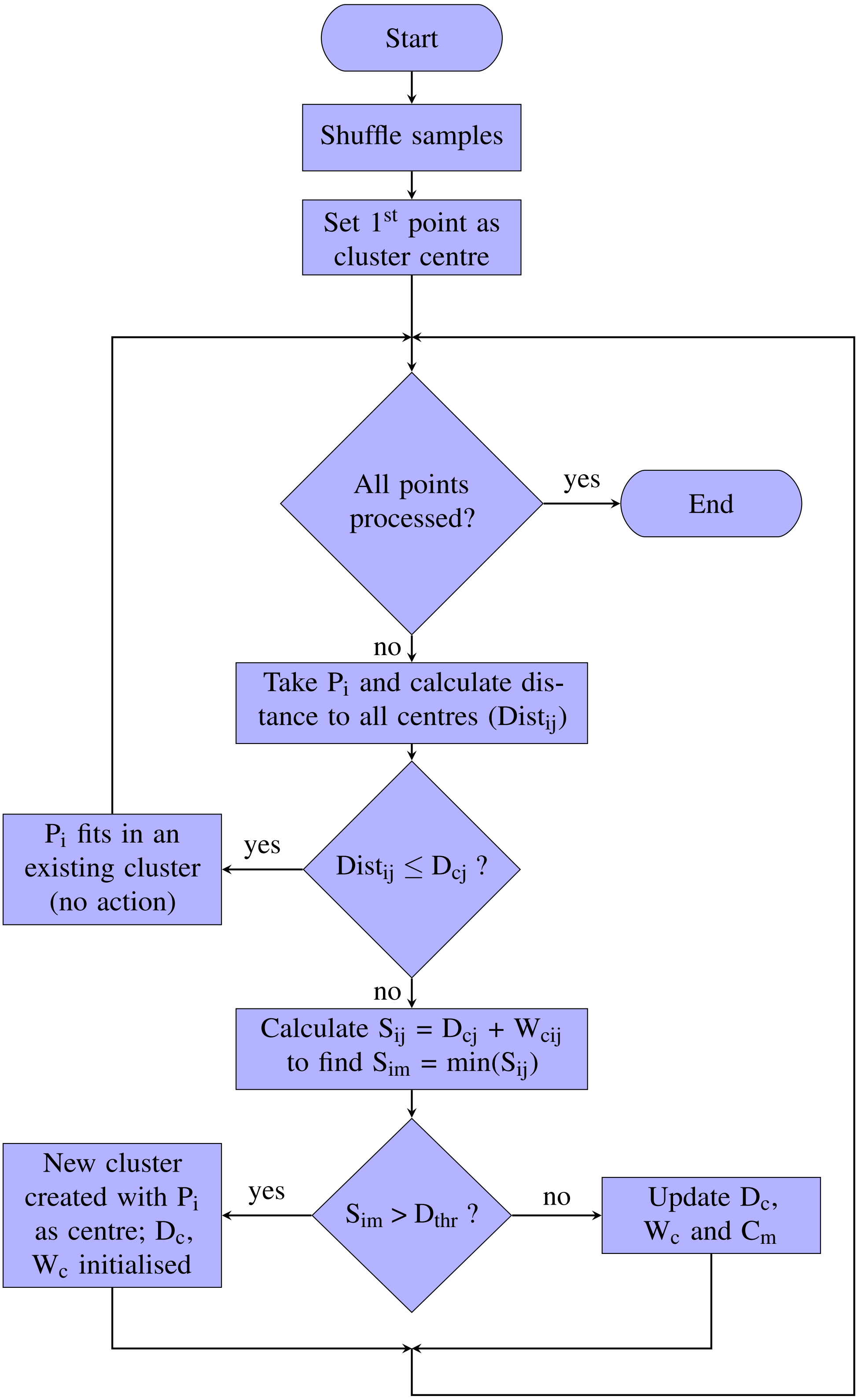

Start
Shuffle samples
Set 1st point as cluster centre
All points processed?
yes
End
no
Take P_i and calculate distance to all centres ($Dist_{ij}$)
$Dist_{ij} \leq D_{cj}$?
yes
P_i fits in an existing cluster (no action)
no
Calculate $S_{ij} = D_{cj} + W_{cij}$ to find $S_{im} = min(S_{ij})$
$S_{im} > D_{thr}$?
yes
New cluster created with P_i as centre; D_c, W_c initialised
no
Update D_c, W_c and C_m

Figure 4.7 Flowchart of the clustering algorithm

The centre and quantum interval updates are computed according to the chain rule, where the output error is back-propagated to the membership function layer. The centre updates are:

$$\begin{aligned}
\Delta c_{ij} &= -\frac{\partial E}{\partial c_{ij}} = -\frac{\partial E}{\partial O_j^5}\cdot\frac{\partial O_j^5}{\partial c_{ij}} \\
&= \delta_e \cdot \left[\frac{(b_j+\sum_{i=1}^{n} a_{ij}x_i)\cdot\sum_{j=1}^{q} O_j^3 - \sum_{j=1}^{q}\big(O_j^3(a_{oj}+\sum_{i=1}^{n} a_{ij}x_i)\big)}{\big(\sum_{j=1}^{q} O_j^3\big)^2}\right]\prod_{\substack{j=1\\ i\neq j}}^{q} A_{ij} \\
&\times \frac{1}{n_{\theta_{ij}}}\sum_{r=1}^{n_{\theta_{ij}}}\left[-\frac{\beta\cdot\exp\big(-\beta(x_i-c_{ij}+|\theta_{ij}^r|)\big)}{\Big(1+exp\big(-\beta(x_i-c_{ij}+|\theta_{ij}^r|)\big)\Big)^2}\cdot U(x_i;-\infty,c_{ij})\right. \\
&\qquad\left. +\frac{\beta\cdot\exp(-\beta(x_i-c_{ij}+|\theta^r|))}{\Big(1+\exp\big(-\beta(x_i-c_{ij}+|\theta_{ij}^r|)\big)\Big)^2}\cdot U(x;c,\infty)\right]
\end{aligned} \tag{4.16}$$

The quantum interval updates are:
If $\theta_{ij}^r \geq 0$, then

$$\begin{aligned}
\Delta\theta_{ij}^r &= -\frac{\partial E}{\partial \theta_{ij}^r} = -\frac{\partial E}{\partial O_j^5}\cdot\frac{\partial O_j^5}{\partial \theta_{ij}^r} \\
&= \delta_e \cdot \left[\frac{(b_j+\sum_{i=1}^{n} a_{ij}x_i)\cdot\sum_{j=1}^{q} O_j^3 - \sum_{j=1}^{q}\big(O_j^3\cdot(a_{oj}+\sum_{i=1}^{n} a_{ij}x_i)\big)}{\big(\sum_{j=1}^{q} O_j^3\big)^2}\right]\prod_{\substack{j=1\\ i\neq j}}^{q} A_{ij} \\
&\times \frac{1}{n_{\theta_{ij}}}\cdot\left[\frac{\beta\cdot\exp\big(-\beta(x_i-c_{ij}+\theta_{ij}^r)\big)}{\Big(1+exp\big(-\beta(x_i-c_{ij}+\theta_{ij}^r)\big)\Big)^2}\cdot U(x_i;-\infty,c_{ij})\right. \\
&\qquad\left. -\frac{\beta\cdot\exp(-\beta(x_i-c_{ij}+\theta^r))}{\Big(1+\exp\big(-\beta(x_i-c_{ij}+\theta_{ij}^r)\big)\Big)^2}\cdot U(x_i;c_{ij},\infty)\right]
\end{aligned} \tag{4.17}$$

else $\theta_{ij}^{r} < 0$

$$\begin{aligned}
\Delta\theta_{ij}^{r} &= -\frac{\partial E}{\partial \theta_{ij}^{r}} = -\frac{\partial E}{\partial O_j^5} \cdot \frac{\partial O_j^5}{\partial \theta_{ij}^{r}} \\
&= \delta_e \cdot \left[\frac{(b_j + \sum_{i=1}^{n} a_{ij} x_i) \cdot \sum_{j=1}^{q} O_j^3 - \sum_{j=1}^{q} \left(O_j^3 \cdot (a_{oj} + \sum_{i=1}^{n} a_{ij} x_i) \right)}{\left(\sum_{j=1}^{q} O_j^3 \right)^2} \right] \prod_{\substack{j=1 \\ i \neq j}}^{q} A_{ij} \\
&\qquad \times \frac{1}{n_{\theta_{ij}}} \cdot \Bigg[-\frac{\beta \cdot \exp\left(-\beta (x_i - c_{ij} - \theta_{ij}^{r})\right)}{\left(1 + exp\left(-\beta (x_i - c_{ij} - \theta_{ij}^{r})\right)\right)^2} \cdot U(x_i; -\infty, c_{ij}) \\
&\qquad\qquad + \frac{\beta \cdot \exp(-\beta (x_i - c_{ij} - \theta^{r}))}{\left(1 + \exp\left(-\beta (x_i - c_{ij} - \theta_{ij}^{r})\right)\right)^2} \cdot U(x_i; c_{ij}, \infty) \Bigg] \qquad (4.18)
\end{aligned}$$

These result in the membership function centres and quantum intervals being updated as follows:

$$\begin{aligned}
c_{ij}(k+1) &= c_{ij}(k) + \eta_c \Delta c_{ij} \\
\theta_{ij}^{r}(k+1) &= \theta_{ij}^{r}(k) + \eta_\theta \Delta \theta_{ij}^{r}
\end{aligned} \qquad (4.19)$$

where η_c, Δc_{ij} and η_θ, $\Delta\theta_{ij}^{r}$ are learning rate and update for the centres and quantum intervals respectively, and k is the time step.

4.5.5 Model Validation

The modelling architecture was successfully implemented for the three-class classification of the iris flower dataset. The data are multivariate with variables corresponding to sepal length, sepal width, petal length, petal width and the flower species. The dataset contains 50 instances of each of the three species, Iris Setosa, Iris Virginica and Iris Versicolor. It was first used in 1936 [21, 3] and is a common choice of dataset for classification using machine learning techniques.

The simulations resulted in very high classification accuracies, averaging 97.5% and comparable to those documented in [46], whose membership functions for the optimized model are shown in Figure 4.8.

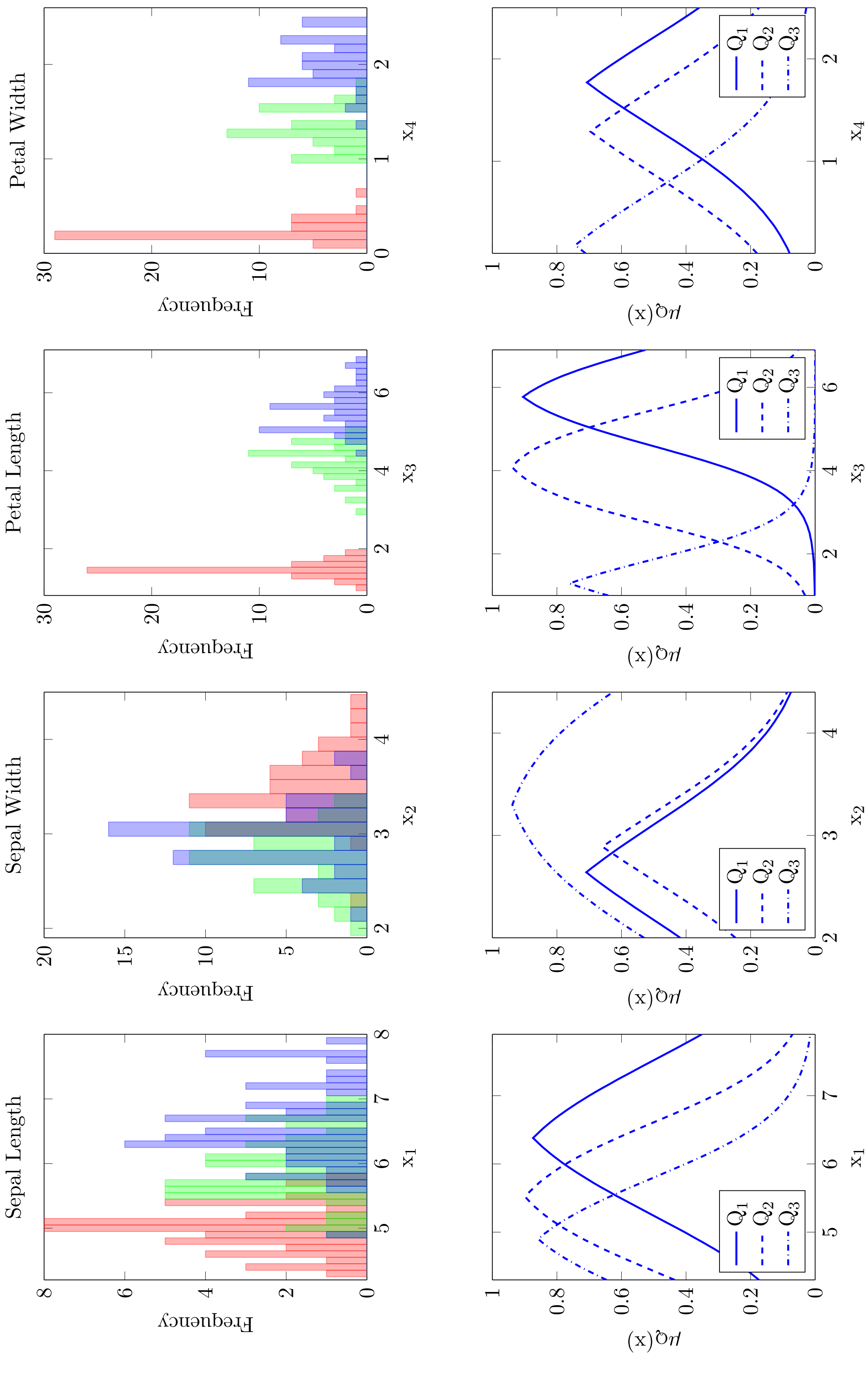


Figure 4.8 Quantum membership functions for classification of the iris dataset

4.6 Clustering Analysis and Simulations Setup

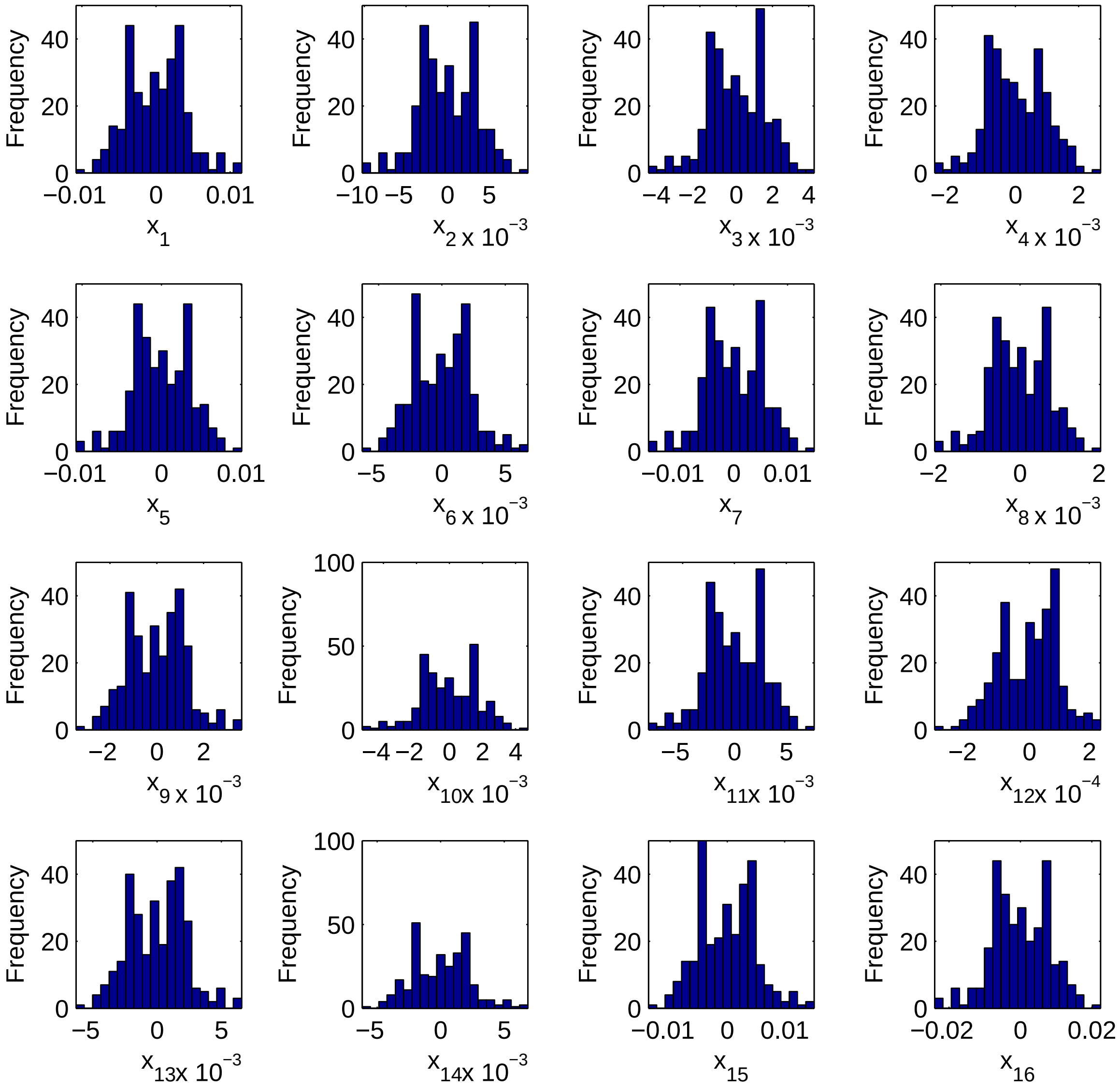
When fitting the model to the Charpy impact data, which was presented in Chapter 2, determining the appropriate distance threshold in the clustering algorithm proved difficult due to the different variable ranges combined with the need to try different normalisation methods. Moreover, the quantum intervals could not be optimised in the same way as in [46], using an entropy-based algorithm, as this relied on the data belonging to classes. FCM clustering was therefore used to provide an initial estimate for the centres of the quantum sets while the number of quantum levels was fixed to 3 per membership function. The data were partitioned into training, validation and testing sets using the ratios 0.55 : 0.15 : 0.30 respectively. The same partitioning ratios were adopted for fair comparison with previous work.

It was noticed that when clustering is done across all variables, the centres are random across the range of each variable (Figure 4.9a). However, clustering the variables individually results in specific points being chosen as centres (Figure 4.9b). These figures were obtained by plotting 100 iterations of FCM clustering with 3 centres each.

The effects of these clustering methods were analysed along with the different normalisation types, where data were either standardised using the mean and standard deviation of the training data, or normalised. For the following settings, 20 simulations of 25 epochs each (50 epochs in one case) were performed for each cluster setting between 3 and 10 clusters:

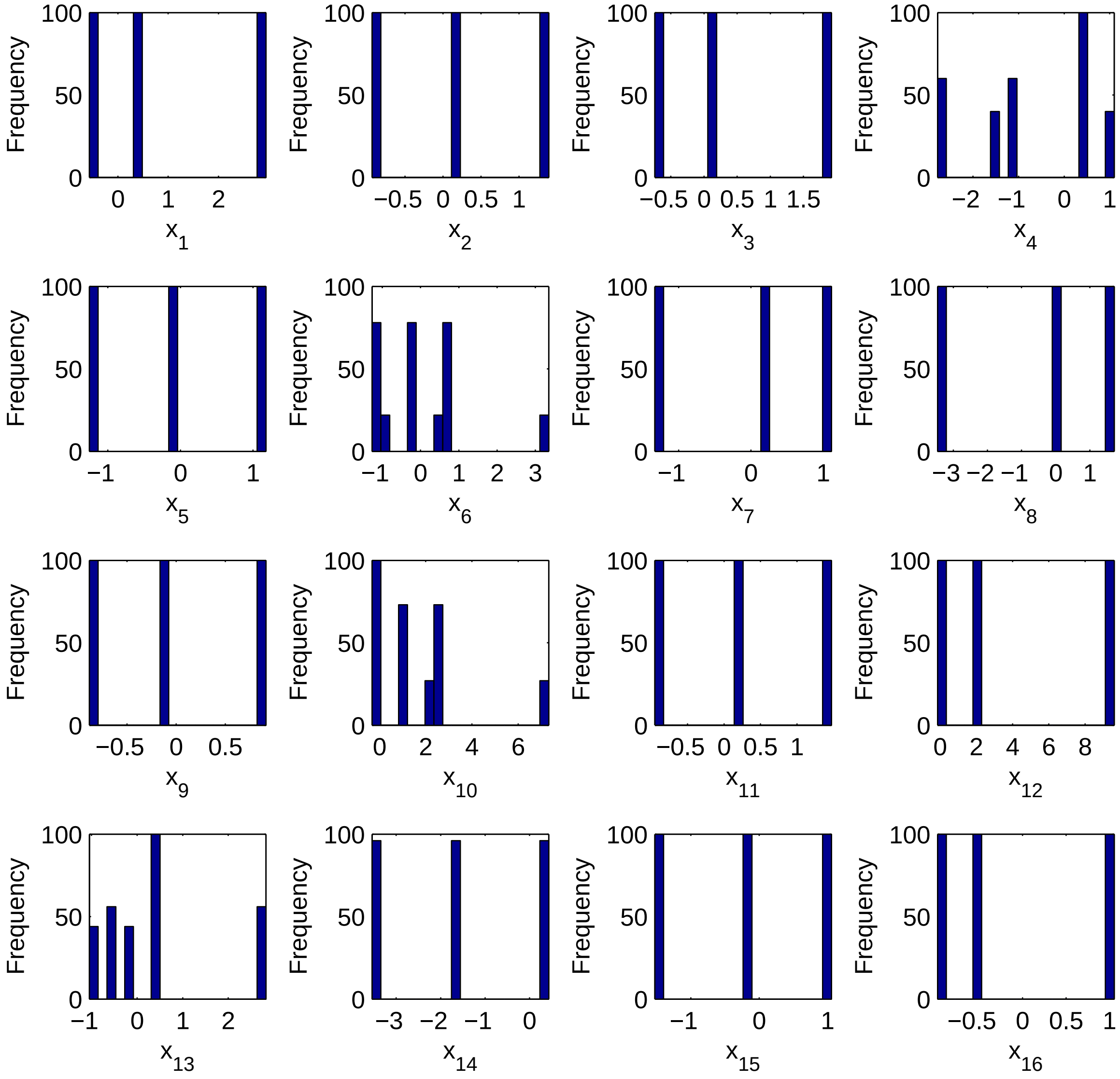
- Standardisation including the outputs (2 simulations with 25 epochs; 1 with 50 epochs)
- Standardisation excluding the outputs
- Normalisation $(-0.5 \ldots 0.5)$ including the outputs
- Normalisation $(-0.5 \ldots 0.5)$ excluding the outputs
- Normalisation $(0 \ldots 1)$ including the outputs
- Standardisation including outputs and clustering across all variables
- Normalisation including outputs and clustering across all variables

Tables 4.1 and 4.2 indicate the number of times that the model did not converge (from the 20 simulations) for every cluster value. Since for the simulation settings indicated in Table 4.2 the number of divergences was too high, only the settings shown in Table 4.1 were considered when choosing a suitable model.



(a) Clustering across all variables

Figure 4.9 Cluster centres obtained using FCM clustering



(b) Clustering on separate variables

Figure 4.9 Cluster centres obtained using FCM clustering (cont.)

Table 4.1 Number of times when model did not converge

Simulation	No convergence per cluster value							
	3	4	5	6	7	8	9	10
Standardisation including output (25 epochs)	1	2	2	5	4	4	8	8
Standardisation including output (25 epochs)	1	1	4	4	4	10	8	6
Standardisation including output (50 epochs)	1	0	6	6	7	6	9	7
Standardisation including output (clustering across all variables)	0	0	0	0	0	0	0	0
Normalisation (0...1) including output (clustering across all variables)	0	0	2	0	0	0	0	0

Table 4.2 Number of times when model did not converge (Results not analysed)

Simulation	No convergence per cluster value							
	3	4	5	6	7	8	9	10
Standardisation excluding output	10	12	13	13	13	14	13	13
Normalisation (-0.5...0.5) including output	10	14	16	17	15	18	14	13
Normalisation (-0.5...0.5) excluding output	18	17	16	17	12	17	19	18
Normalisation (0...1) including output	6	16	15	18	17	17	17	16

For the simulations that were analysed, the results were averaged across all simulations for each setting to allow comparison between the different architectures (Tables 4.3 to 4.6). The best testing results from each simulation for every cluster are shown in Tables 4.7 to 4.11. These are based on the testing data performance, with the corresponding training and validation results.

A higher number of epochs was also tested for some of the settings. However this resulted in overfitting the training data or no noticeable performance improvement. To summarise, results indicated that the following improve the modelling performance:

- Clustering done on the variables separately rather than across all variables.
- The data are standardised rather than normalised.
- The output is standardised or normalised as well.

Table 4.3 Standardisation including the outputs (25 epochs – average of the 2 simulations) (Average results)

Clusters	Average RMSE		
	Training	Validation	Testing
3	20.08	21.85	21.46
4	19.39	21.91	21.16
5	19.07	21.11	21.10
6	18.80	21.02	20.63
7	18.68	21.25	20.53
8	18.67	21.32	19.79
9	18.60	21.07	20.08
10	18.63	21.18	20.18

4.7 Results for the Chosen Model

Considering both the performance and the times when the model optimisation diverged (Tables 4.1 and 4.2), it was decided to use a model with 6 rules. Table 4.12 presents the results for a model with 6 clusters, with a resulting correlation coefficient of 82% between the real and predicted outputs for the testing data as shown in Figure 4.10. The low variation in Root-Mean-Square Error (RMSE) across the three data sets indicates that the model performs consistently on the data.

Table 4.4 Standardisation including the outputs (50 epochs) (Average results)

Clusters	Average RMSE		
	Training	Validation	Testing
3	19.82	21.70	21.17
4	18.83	20.95	20.76
5	18.46	20.90	20.98
6	18.25	20.76	20.65
7	18.14	20.87	20.64
8	17.94	20.64	20.75
9	17.82	20.31	20.77
10	17.99	20.58	20.47

Table 4.5 Standardisation including the outputs and clusters across all variables (Average results)

Clusters	Average RMSE		
	Training	Validation	Testing
3	20.55	21.94	22.28
4	19.84	21.98	21.35
5	19.93	22.26	21.39
6	20.03	22.35	21.21
7	20.07	22.25	21.09
8	20.27	22.32	21.47
9	20.47	22.30	21.77
10	20.61	22.27	21.90

Table 4.6 Normalisation (0...1) including the outputs and clusters across all variables (Average results)

Clusters	Average RMSE		
	Training	Validation	Testing
3	21.82	23.07	22.65
4	21.81	23.26	22.42
5	21.96	23.41	22.56
6	22.23	23.55	22.65
7	22.33	23.74	22.53
8	22.72	24.07	22.97
9	23.45	24.49	23.57
10	24.23	25.05	24.27

Table 4.7 Standardisation including the outputs (25 epochs) - 1st simulation (Best testing results)

Clusters	RMSE		
	Training	Validation	Testing
3	20.04	21.93	19.43
4	19.59	23.50	19.55
5	19.23	22.16	18.93
6	19.08	20.69	18.85
7	18.53	21.04	18.87
8	18.42	20.69	18.63
9	18.36	20.24	18.50
10	18.29	22.75	18.91

Table 4.8 Standardisation including the outputs (25 epochs) - 2nd simulation (Best testing results)

Clusters	RMSE		
	Training	Validation	Testing
3	20.86	22.09	20.23
4	19.71	20.63	19.85
5	19.36	21.02	19.45
6	19.10	21.11	19.09
7	19.04	21.07	18.79
8	18.39	20.61	18.02
9	18.62	22.16	18.40
10	18.27	22.62	18.53

Table 4.9 Standardisation including the outputs (50 epochs) (Best testing results)

Clusters	RMSE		
	Training	Validation	Testing
3	20.05	21.97	19.32
4	19.00	21.07	19.14
5	18.07	19.61	18.97
6	**17.75**	**18.84**	**18.17**
7	18.12	22.37	19.25
8	17.93	21.12	18.50
9	17.84	19.92	18.53
10	17.91	21.04	18.48

Table 4.10 Standardisation including the outputs and clusters across all variables (Best testing results)

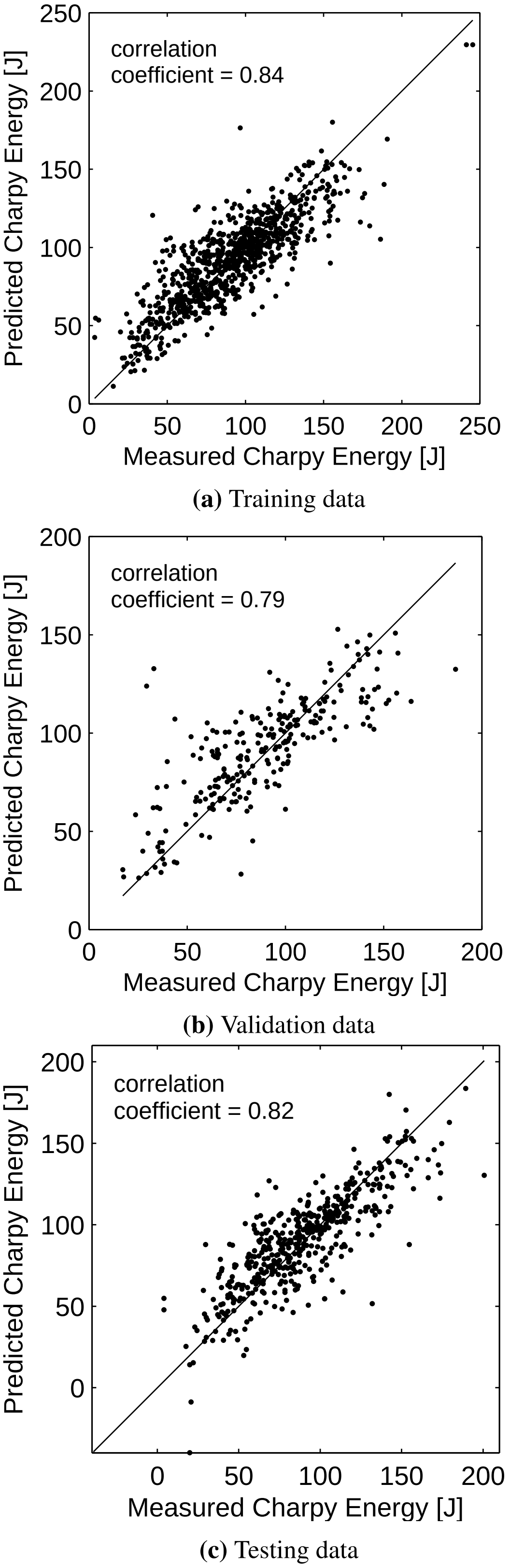
Clusters	RMSE		
	Training	Validation	Testing
3	19.94	22.05	20.86
4	19.68	22.26	20.59
5	19.72	22.02	20.73
6	19.95	22.25	20.78
7	19.87	22.28	20.83
8	20.06	22.29	21.06
9	20.22	22.27	21.50
10	20.54	22.24	21.74

Table 4.11 Normalisation (0...1) including the outputs and clusters across all variables (Best testing results)

Clusters	RMSE		
	Training	Validation	Testing
3	21.68	22.96	22.24
4	21.39	23.88	21.60
5	21.31	22.52	21.86
6	21.71	22.63	21.63
7	21.90	24.43	21.85
8	22.33	23.52	22.45
9	22.93	24.08	23.05
10	23.62	24.48	23.58

Table 4.12 Model performance

	Training data	Validation data	Testing data
Correlation coefficient	0.84	0.79	0.82
RMSE (Joules)	17.75	18.84	18.17



(a) Training data

(b) Validation data

(c) Testing data

Figure 4.10 Charpy energy prediction

These results are comparable with those obtained in previous publications using the same dataset [83, 48, 64, 96, 75] which are summarised in Table 4.13. Figure 4.11 shows a plot of the membership functions across the data variables.

Table 4.13 Past results of Charpy impact energy prediction

	RMSE Training data	RMSE Validation data	RMSE Testing data
Ensemble NN [83]	13.20	17.10	18.30
BNN [48]	17.31	20.77	19.49
GrC-NF [64]	14.66	21.24	20.42
GA-NN Ensemble [96]	13.12	17.25	18.13
IT2-GrC-NF [75]	16.27	18.20	19.87

4.8 Summary

This chapter showed that promising modelling results were obtained using a Quantum-membership-function-based fuzzy model to predict Charpy energy for data obtained from the Charpy impact test.

From Figure 4.11 it can be noticed that although the number of quantum levels was fixed to 3 per membership function, few of them exhibit evident quantum levels. This indicates that while the model was able to capture the uncertainty in the data, more research is required to understand the effects of quantum levels in these membership functions. This can be done by restricting the membership function widths and fixing some of the levels. Further changes that can be made to the model stem from whether it has a smooth or coarse decision surface with respect to the input variables. This is influenced by the shape of the membership functions and has an effect on the performance of the model.

Different optimisation procedures can also be implemented such as optimising and then fixing the parameters of the different sections of the model separately, and using an adaptive optimisation algorithm. To better understand the membership functions, a simpler model may also be used such as one based on a Mamdani-type fuzzy logic structure.

The next chapter attempts to improve the model by including Type-2 Fuzzy Sets into the modelling framework.

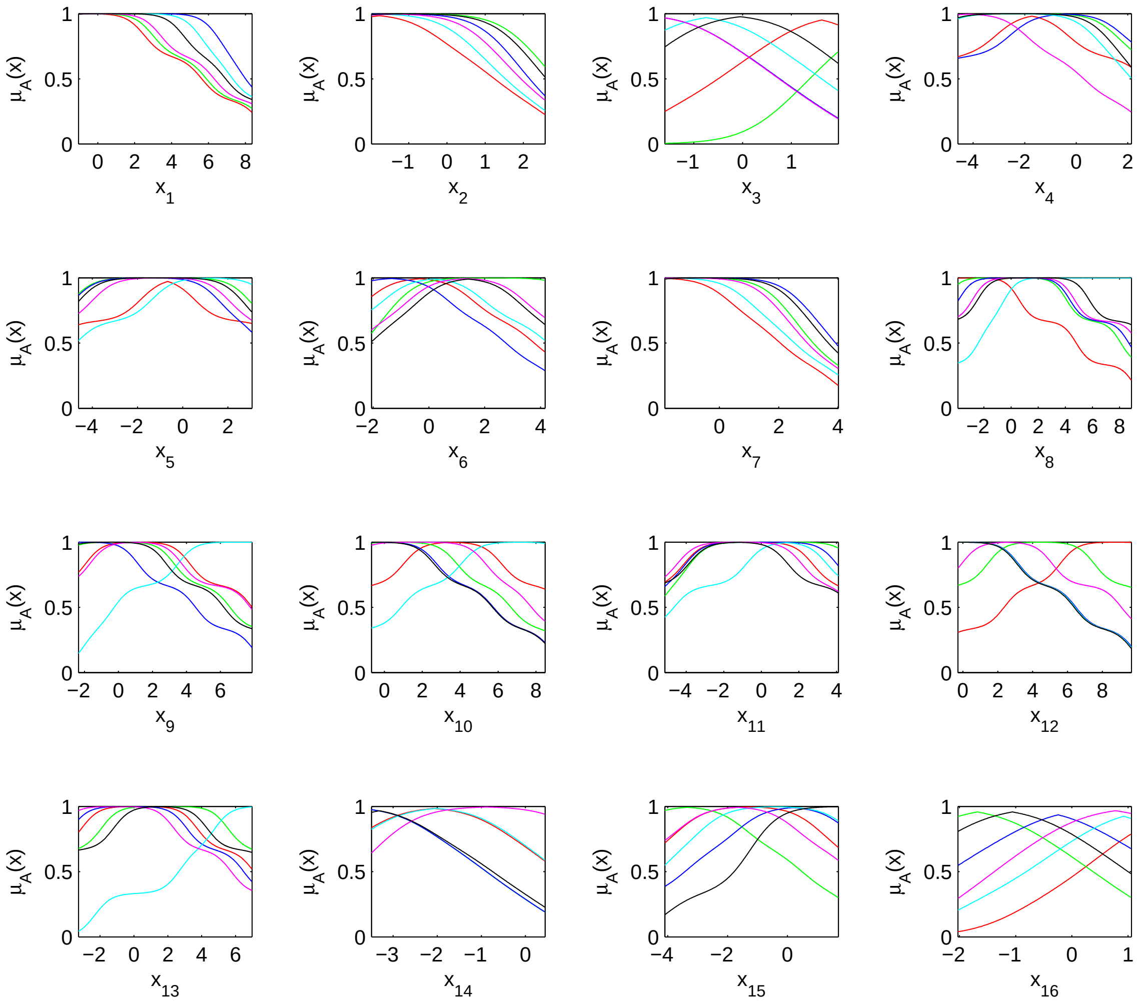


Figure 4.11 Membership functions for the Charpy impact data input variables

Chapter 5

Type-2 Quantum Membership Function-based Fuzzy Modelling

5.1 Introduction

It was shown previously in Chapter 4 how fuzzy logic can prove useful when dealing with uncertainty. However, although it seems paradoxical, Type-1 Fuzzy Sets have limited capability in handling uncertainty, where handle means ‘to model and minimise the effect of’[53]. This is because the membership degree of a Type-1 Fuzzy Set is a crisp value, ignoring any uncertainty that might surround it. In his work on fuzzy sets, Zadeh had already proposed the idea of higher-order forms of fuzzy logic [101]. This chapter will therefore investigate the inclusion of Type-2 Fuzzy Sets to improve the uncertainty handling capability of the modelling framework.

5.2 Fuzzy Logic and Type-2 Fuzzy Sets

As briefly indicated, a Type-1 fuzzy system does not capture uncertainty well enough because the membership function of a Type-1 fuzzy set is certain (i.e. the membership degree being a crisp value). As ‘words mean different things to different people’ [53], various persons can form different fuzzy sets to represent a particular concept and thus the need for Type-2 Fuzzy Sets arises when more levels of uncertainty need to be represented.

5.2.1 Type-2 Fuzzy Sets

A type-2 fuzzy set can be defined as follows:

$$\tilde{A} = \left\{ \left((x,u), \mu_{\tilde{A}}(x,u) \right) \mid \forall x \in X, \forall u \in J_x \subseteq [0,1], 0 \leq \mu_{\tilde{A}}(x,u) \leq 1 \right\} \tag{5.1}$$

X is the primary domain of $\tilde{A}$ and u is called the primary membership. J_x is called the secondary domain and $\mu_{\tilde{A}}$ is called the secondary membership.

Referring to the example presented in Section 4.3.1, it was shown how ‘Warm Weather’ may be described by a fuzzy set. However, another person can form a different ‘Warm Weather’ fuzzy set as indicated in Figure 5.1. This gives rise to interpersonal uncertainty as a group of people have different views about the representative meaning of a word [93].

More fuzzy sets can be added and eventually a particular temperature can be mapped to a range of membership degrees, having a minimum and a maximum. A Type-2 Fuzzy Set has another membership function, the secondary membership function, on top of this range with its own membership degrees. The membership degrees of the secondary membership

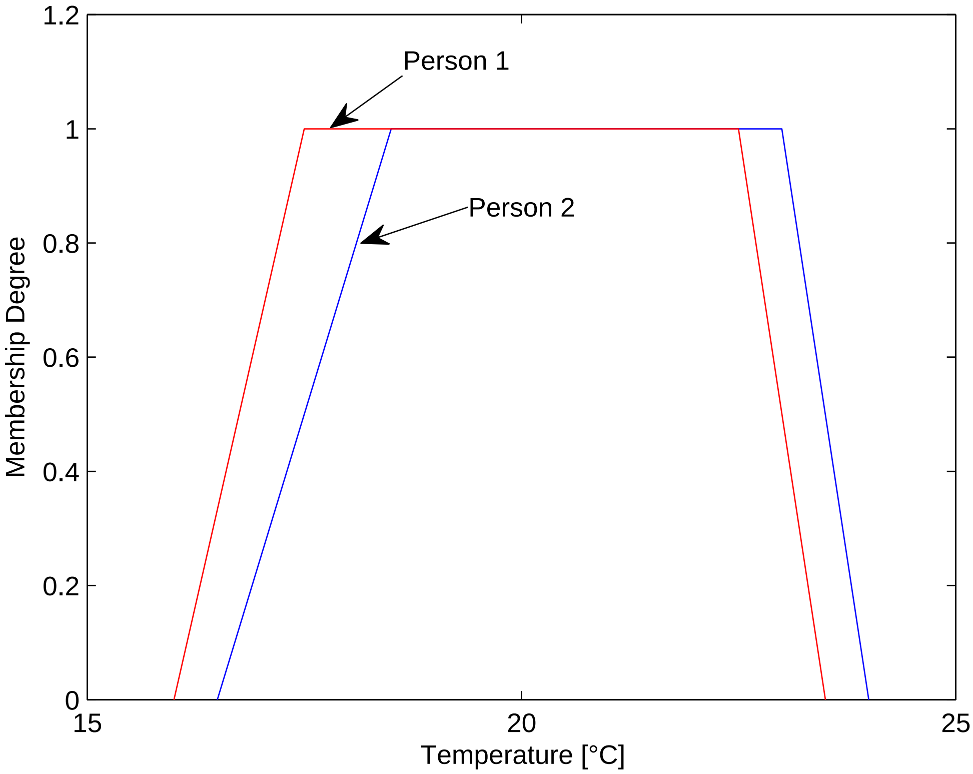


Figure 5.1 2 ‘Warm Weather’ Type-1 membership functions

function can be thought of as a normalised histogram made from the collection of different primary memberships at each particular point as indicated in Figure 5.2. Because of this a Type-2 Fuzzy Set can also be called a ‘fuzzy-fuzzy set’.

Although a Type-1 Fuzzy Set can be seen to represent intrapersonal uncertainty as it shows the varying degree of membership a person assigns to a particular word, its membership function is completely certain. For this reason, a Type-2 Fuzzy Set is more suited to represent a linguistic variable and can model both interpersonal and intrapersonal uncertainties better than a Type-1 Fuzzy Set [93].

There are two main representations for a Type-2 Fuzzy Set: vertical-slice representation and wavy-slice representation. The vertical-slice representation (Figure 5.3b) is very useful for computational purposes while the wavy-slice representation (Figure 5.3c), also known as the Mendel-John Representation Theorem [53], is more utilised in theoretical derivations.

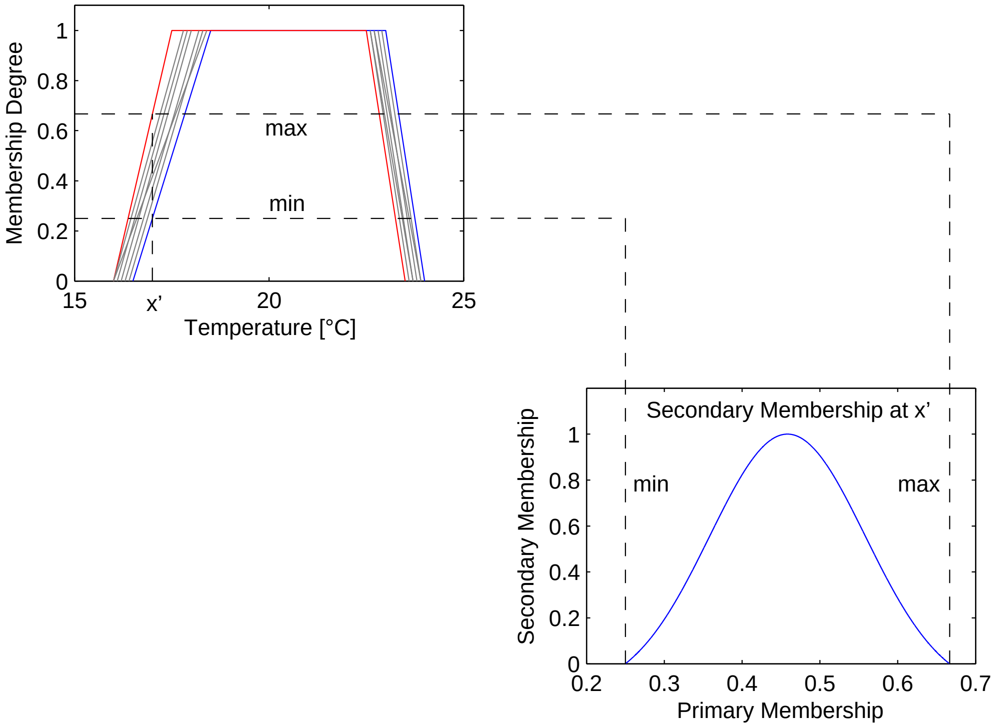
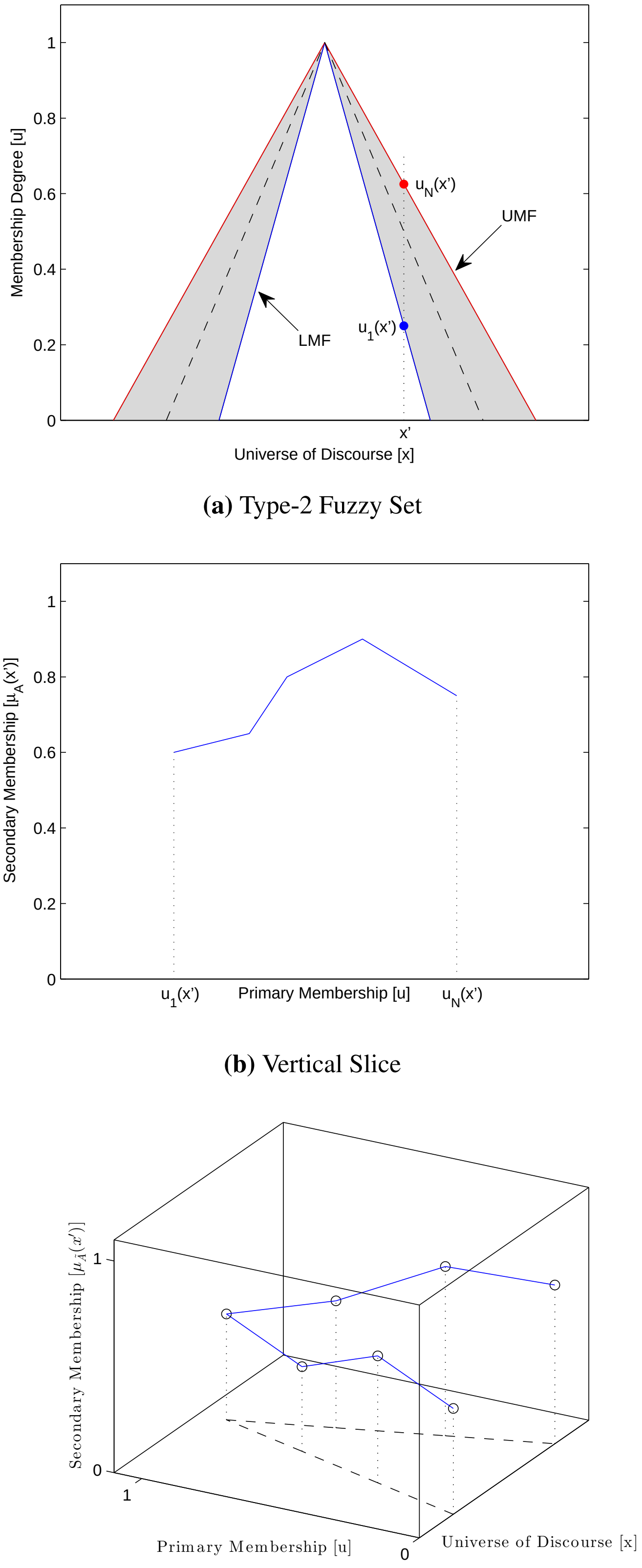


Figure 5.2 Type-2 membership function with secondary membership at x’



(a) Type-2 Fuzzy Set

(b) Vertical Slice

(c) Wavy Slice

Figure 5.3 Type-2 Fuzzy Set representation

5.2.2 Type-2 Fuzzy Logic System

As Type-2 Fuzzy Sets were proposed for more uncertain situations, a more elaborate account of uncertainty is deemed to be necessary at this stage. Some questions that one might ask are: What is uncertainty? Are there different types of uncertainty? What are the causes or sources of uncertainty?

In [52], Mendel quotes Klir and Wierman [40], stating that the occurrence of uncertainty is an unavoidable reality: at an empirical level in measurement, at the cognitive level in language, and even at the social level. As to the nature of uncertainty they state that three types can be identified: fuzziness (or vagueness), nonspecificity (or imprecision), and strife (or discord), along with giving a number of synonyms for each type.

Although on its own 'uncertainty' may seem abstract, there are four practical sources of uncertainty which can be identified with reference to the FLS of Figure 4.2 [52]:

i. Words used for the antecedent and consequent clauses may be different for different people;

ii. Experts may not necessarily agree on the consequent and hence the logic of the rules;

iii. Noisy data are used to train the system;

iv. Noisy measurements activate the system.

Since Type-2 Fuzzy Sets are better suited to handle uncertainty than their Type-1 counterparts, a Type-2 FLS can be used to deal with some of these uncertainties. The general form of a Type-2 FLS is shown in Figure 5.4. The inference engine in such a system processes Type-2 Fuzzy Sets and for this reason the output processing block is made up of a type-reducer and defuzzifier. The type-reducer is necessary to transform a Type-2 Fuzzy Set into a Type-1 Fuzzy Set for defuzzification.

John and Coupland [37] reference Turksen [86] when noting that,

> "the expressive power of Type-2 fuzzy reasoning lies in the ability to retain the uncertainty throughout the inferencing process."

5.2.3 Interval Type-2 Fuzzy Sets

Type reduction is a computationally intensive procedure while working with General Type-2 Fuzzy Sets requires complex computations. In fact, most of the research on the application of Type-2 Fuzzy Sets has focused on Interval Type-2 Fuzzy Sets [55]. This was facilitated by

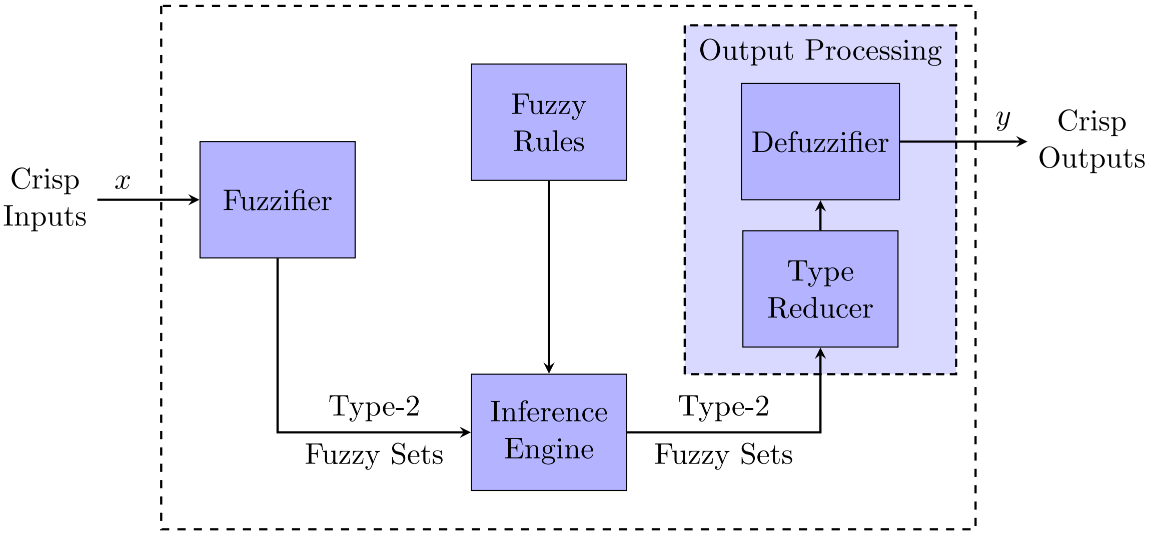


Figure 5.4 Type-2 Fuzzy Logic System

the development of efficient type reduction techniques such as the Karnik-Mendel algorithms [52, 39].

With reference to Equation 5.1, an Interval Type-2 Fuzzy Set is a Type-2 Fuzzy Set whose secondary membership grades are all equal to one,

$$\tilde{A} = \left\{ \big((x,u),1\big) \mid \forall x \in X, \forall u \in J_x \subseteq [0,1] \right\} \tag{5.2}$$

The Representation Theorem [54] makes it possible to express Interval Type-2 Fuzzy Set operations using Type-1 Fuzzy Set mathematics [55]. The fuzzy set is represented by the union of its embedded sets (or wavy slices):

$$\tilde{A} = \sum_{j=1}^{n_A} \tilde{A}_e^j \qquad \text{where } j = 1, \ldots, n_A \tag{5.3}$$

$$\tilde{A}_e^j = \sum_{i=1}^{N} [1/u_i^j]/x_i \qquad \text{where } u_i^j \in J_{x_i} \subseteq [0,1] \tag{5.4}$$

$$n_A = \prod_{i=1}^{N} M_i \tag{5.5}$$

where M_i is the number of discretisation levels of the primary membership u_i^j for each of the N x_i.

Since the secondary memberships are all equivalent to unity, an Interval Type-2 Fuzzy Set is completely described by the Footprint of Uncertainty (FOU) which is the union of all primary memberships.

$$\text{FOU}(\tilde{A}) = \bigcup_{x \in X} J_x \tag{5.6}$$

The FOU is bounded by the Upper Membership Function (UMF) and the Lower Membership Function (LMF), which are embedded Type-1 Fuzzy Sets (Figure 5.5).

$$\begin{aligned} \overline{\mu}_{\tilde{A}}(x) &\equiv \overline{\text{FOU}(\tilde{A})} \quad \forall x \in X \\ \underline{\mu}_{\tilde{A}}(x) &\equiv \underline{\text{FOU}(\tilde{A})} \quad \forall x \in X \end{aligned} \tag{5.7}$$

As with Type-1 Fuzzy Sets, various membership function shapes can be used when modelling using Interval Type-2 Fuzzy Sets. However, a common practice when initialising membership functions is to vary either the centre or spread of the function [43]. This is depicted in Figure 5.6 for a Gaussian membership function where one of the two moments (i.e. mean or standard deviation) is fixed while the other is uncertain.

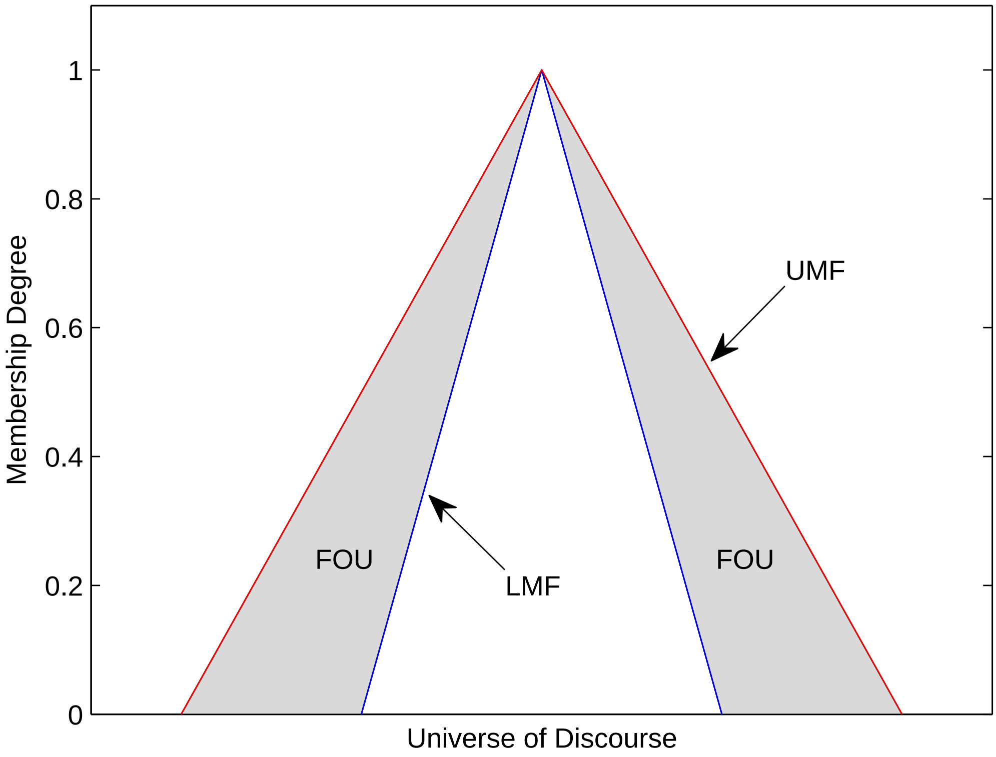
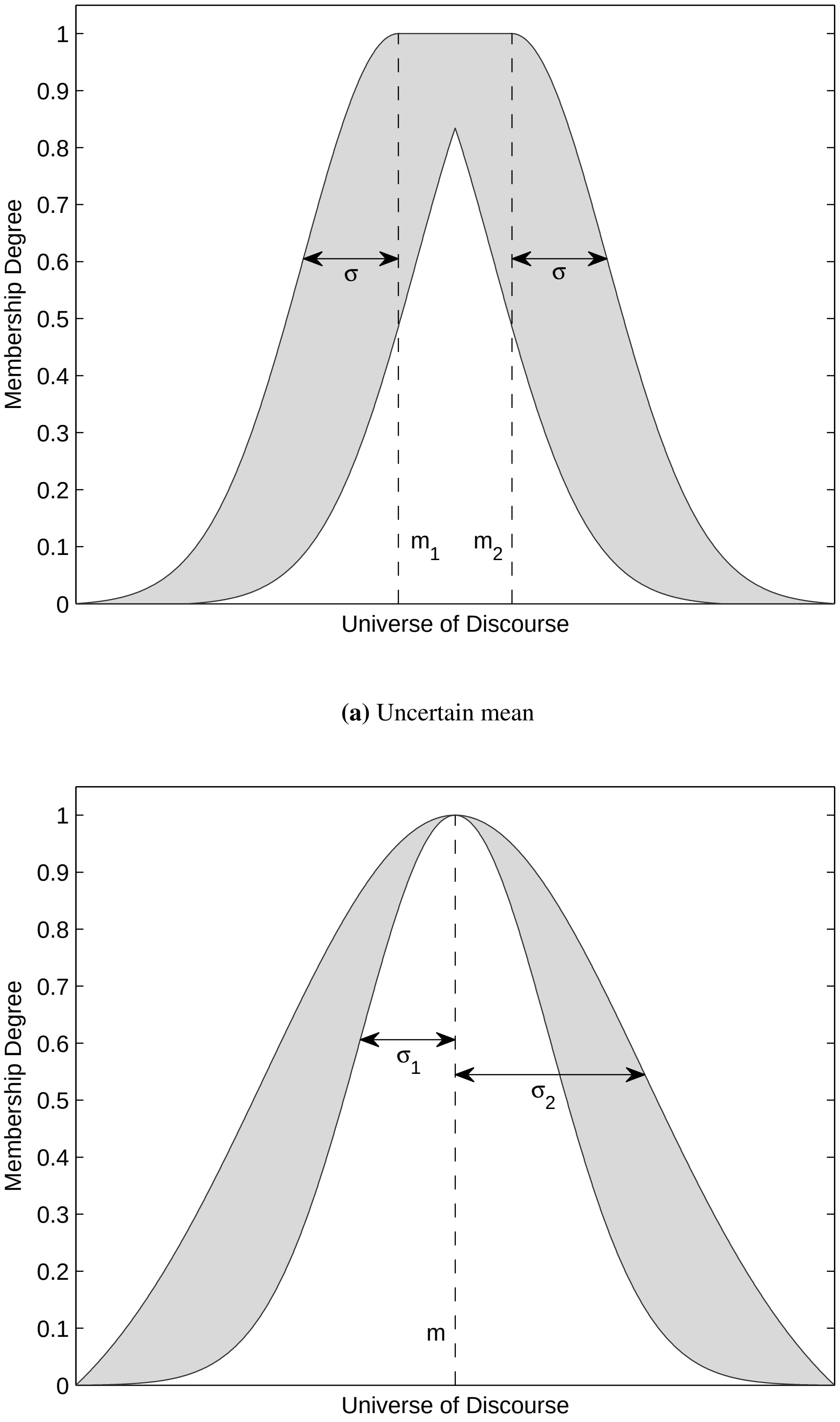


Figure 5.5 Interval Type-2 Fuzzy Set



(a) Uncertain mean

(b) Uncertain standard deviation

Figure 5.6 Interval Type-2 membership function shapes

5.3 Methodology for Eliciting the Type-2 FLS from the Data

5.3.1 Clustering

As with Type-1 systems, one of the methods for initialising a Type-2 system is through clustering [13]. A Type-2 fuzzy clustering algorithm, as described in [33] and employed in [61], was implemented to obtain the initial structure of the Interval Type-2 fuzzy model. The algorithm generalises FCM clustering as presented in Section 3.3.6. Two fuzzifier values, m_1 and m_2, are used to obtain a fuzzy region between clusters. The objective functions to be minimised are now as follows:

$$\begin{aligned} J_{m_1}(X,U,V) &= \sum_{i=1}^{N}\sum_{j=1}^{C} \mu_j(\boldsymbol{x}_i)^{m_1} d_{ji}^2 \\ J_{m_2}(X,U,V) &= \sum_{i=1}^{N}\sum_{j=1}^{C} \mu_j(\boldsymbol{x}_i)^{m_2} d_{ji}^2 \end{aligned} \tag{5.8}$$

where $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N]$ represents the data samples, $U = [\mu_j(\boldsymbol{x}_i)]$ is the fuzzy partition matrix, $d_{ji}^2 = \|\boldsymbol{x}_i - \boldsymbol{v}_j\|^2$, $V = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_C]$ is the vector of cluster centres, m_1 and m_2 are the weighting exponents.

Since Interval Type-2 Fuzzy Sets are used for computational practicality, the primary memberships can be expressed as an FOU and the lower and upper membership degrees of every point to each cluster are:

$$\begin{aligned} \underline{\mu}_j(x_i) &= \begin{cases} \frac{1}{\sum_{k=1}^{C}(d_{ij}/d_{ik})^{2/(m_1-1)}} & \text{if } \frac{1}{\sum_{k=1}^{C}(d_{ij}/d_{ik})^{2/(m_1-1)}} \leq \frac{1}{\sum_{k=1}^{C}(d_{ij}/d_{ik})^{2/(m_2-1)}}, \\ \frac{1}{\sum_{k=1}^{C}(d_{ij}/d_{ik})^{2/(m_2-1)}} & \text{otherwise.} \end{cases} \\ \overline{\mu}_j(x_i) &= \begin{cases} \frac{1}{\sum_{k=1}^{C}(d_{ij}/d_{ik})^{2/(m_1-1)}} & \text{if } \frac{1}{\sum_{k=1}^{C}(d_{ij}/d_{ik})^{2/(m_1-1)}} > \frac{1}{\sum_{k=1}^{C}(d_{ij}/d_{ik})^{2/(m_2-1)}}, \\ \frac{1}{\sum_{k=1}^{C}(d_{ij}/d_{ik})^{2/(m_2-1)}} & \text{otherwise.} \end{cases} \end{aligned} \tag{5.9}$$

The cluster centres are also Interval Type-2 Fuzzy Sets and therefore type reduction is necessary for the centres to be computed and used in the update equations. Generalised Centroid (GC) type reduction for a discrete Type-2 FS, $\tilde{A}$, is given by:

$$\boldsymbol{v}_{\tilde{A}} = \sum\nolimits_{u(x_1)\in J_{x_1}} \cdots \sum\nolimits_{u(x_N)\in J_{x_N}} [f(u(x_1)) * \cdots * f(u(x_N))] \Big/ \frac{\sum_{i=1}^{N} \boldsymbol{x}_i u(\boldsymbol{x}_i)}{\sum_{i=1}^{N} u(\boldsymbol{x}_i)} \tag{5.10}$$

When considering Interval Type-2 Fuzzy Sets and including the weighting exponent m, the clustering centres can be represented as:

$$\boldsymbol{v}_{\tilde{A}} = [v_L, v_R] = \sum\nolimits_{u(x_1)\in J_{x_1}} \cdots \sum\nolimits_{u(x_N)\in J_{x_N}} 1 \Bigg/ \frac{\sum_{i=1}^{N} \boldsymbol{x}_i u(\boldsymbol{x}_i)^m}{\sum_{i=1}^{N} u(\boldsymbol{x}_i)^m} \tag{5.11}$$

The interval $[v_L, v_R]$ is computed using the iterative Karnik-Mendel (KM) algorithms [55]:

$$\begin{aligned} v_j^L = v_j(L) &= \frac{\sum_{i=1}^{L} x_i \overline{u}_j(x_i) + \sum_{i=L+1}^{N} x_i \underline{u}_j(x_i)}{\sum_{i=1}^{L} \overline{u}_j(x_i) + \sum_{i=L+1}^{N} \underline{u}_j(x_i)} \\ v_j^R = v_j(R) &= \frac{\sum_{i=1}^{R} x_i \underline{u}_j(x_i) + \sum_{i=R+1}^{N} x_i \overline{u}_j(x_i)}{\sum_{i=1}^{R} \underline{u}_j(x_i) + \sum_{i=R+1}^{N} \underline{u}_j(x_i)} \end{aligned} \tag{5.12}$$

The centres are then defuzzified as follows:

$$v_j = \frac{v_j^L + v_j^R}{2} \tag{5.13}$$

Further details about the type reduction and hard-partitioning of the membership degrees can be found in [33]. These are sometimes required to specifically assign points to clusters such as in a classification context. A flowchart of the Interval Type-2 clustering algorithm is shown in Figure 5.7.

5.3.2 Interval Type-2 Quantum Membership Function

Through the clustering algorithm, an Interval Type-2 fuzzy partition matrix was obtained, consisting of the upper and lower membership degrees of every training point to each cluster. Upper and lower fuzzy covariance matrices were created for every dimension of each cluster using the fuzzy covariance defined as follows [26]:

$$F_i = \frac{\sum_{k=1}^{N} (\mu_{ik})^m (z_k - v_i)(z_k - v_i)^T}{\sum_{k=1}^{N} (\mu_{ik})^m} \tag{5.14}$$

These were then used to initialise Interval Type-2 Quantum membership functions with cluster centres providing fixed centres and variable widths obtained by scaling the lower and upper standard deviations for the respective dimension.

$$\theta_{ij}^r = \sqrt{F_i} \times r \qquad \text{for } r = 1, 2, \ldots, n_\theta \tag{5.15}$$

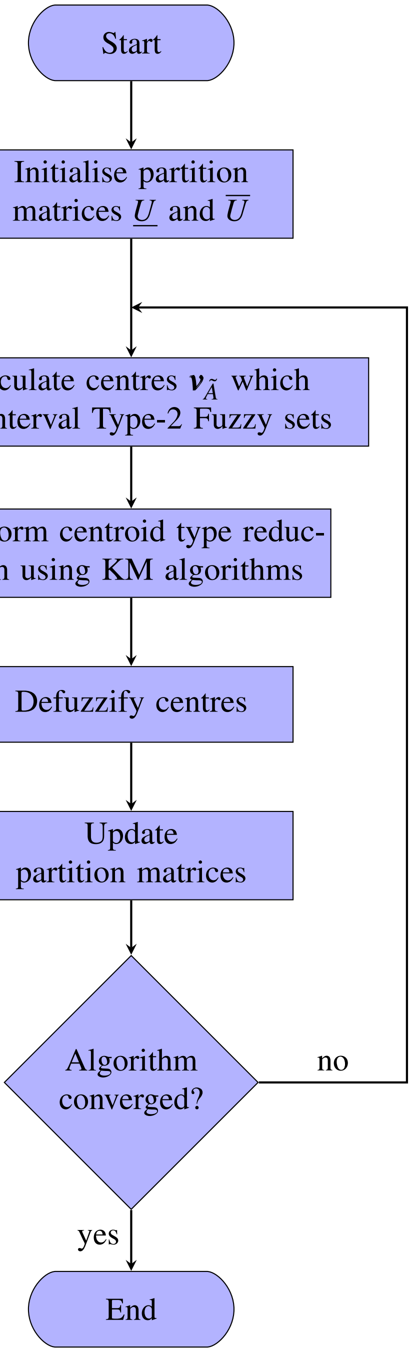

Start
Initialise partition matrices $\underline{U}$ and $\overline{U}$
Calculate centres $\boldsymbol{v}_{\tilde{A}}$ which are Interval Type-2 Fuzzy sets
Perform centroid type reduction using KM algorithms
Defuzzify centres
Update partition matrices
Algorithm converged?
no
yes
End

Figure 5.7 Flowchart of the Interval Type-2 clustering algorithm

An example of such a membership function is shown in Figure 5.8 while Figure 5.9 shows the Interval Type-2 membership functions with 3 clusters for all dimensions of the Charpy impact data.

5.3.3 Interval Type-2 TSK FLS

The interval type-2 sets were used as antecedents in a first-order TSK fuzzy system with rules:

$$R_j: \quad \text{IF } x_1 \text{ is } \tilde{A}_{1j} \textit{ and } x_2 \text{ is } \tilde{A}_{2j} \ \dots \text{ and } x_n \text{ is } \tilde{A}_{nj} \ \text{ THEN } \ y_j \text{ is } b_j + \sum_{i=1}^{n} a_{ij} x_i \tag{5.16}$$

The firing set for rule j can be represented by the interval set $F^j(x) = [\underline{f}^j(\boldsymbol{x}), \overline{f}^j(\boldsymbol{x})]$, where $\underline{f}^j(\boldsymbol{x}) = \underline{\mu}_{\tilde{A}^j_1}(x_1) * \cdots * \underline{\mu}_{\tilde{A}^j_{16}}(x_{16})$ and $\overline{f}^j(\boldsymbol{x}) = \overline{\mu}_{\tilde{A}^j_1}(x_1) * \cdots * \overline{\mu}_{\tilde{A}^j_{16}}(x_{16})$. The output of the

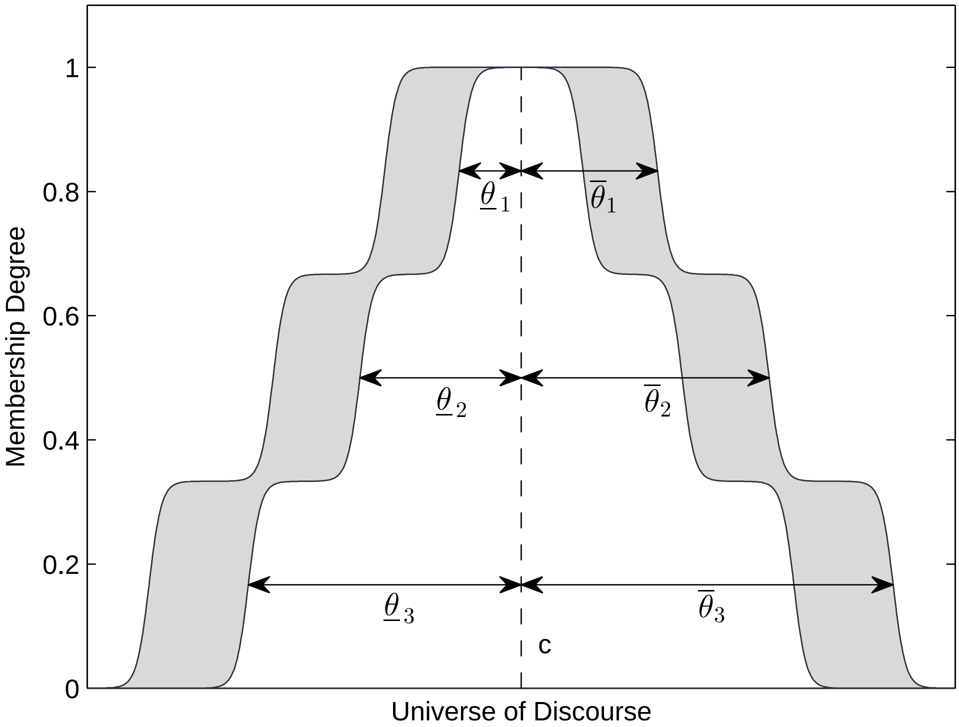


Figure 5.8 Interval Type-2 Quantum membership function

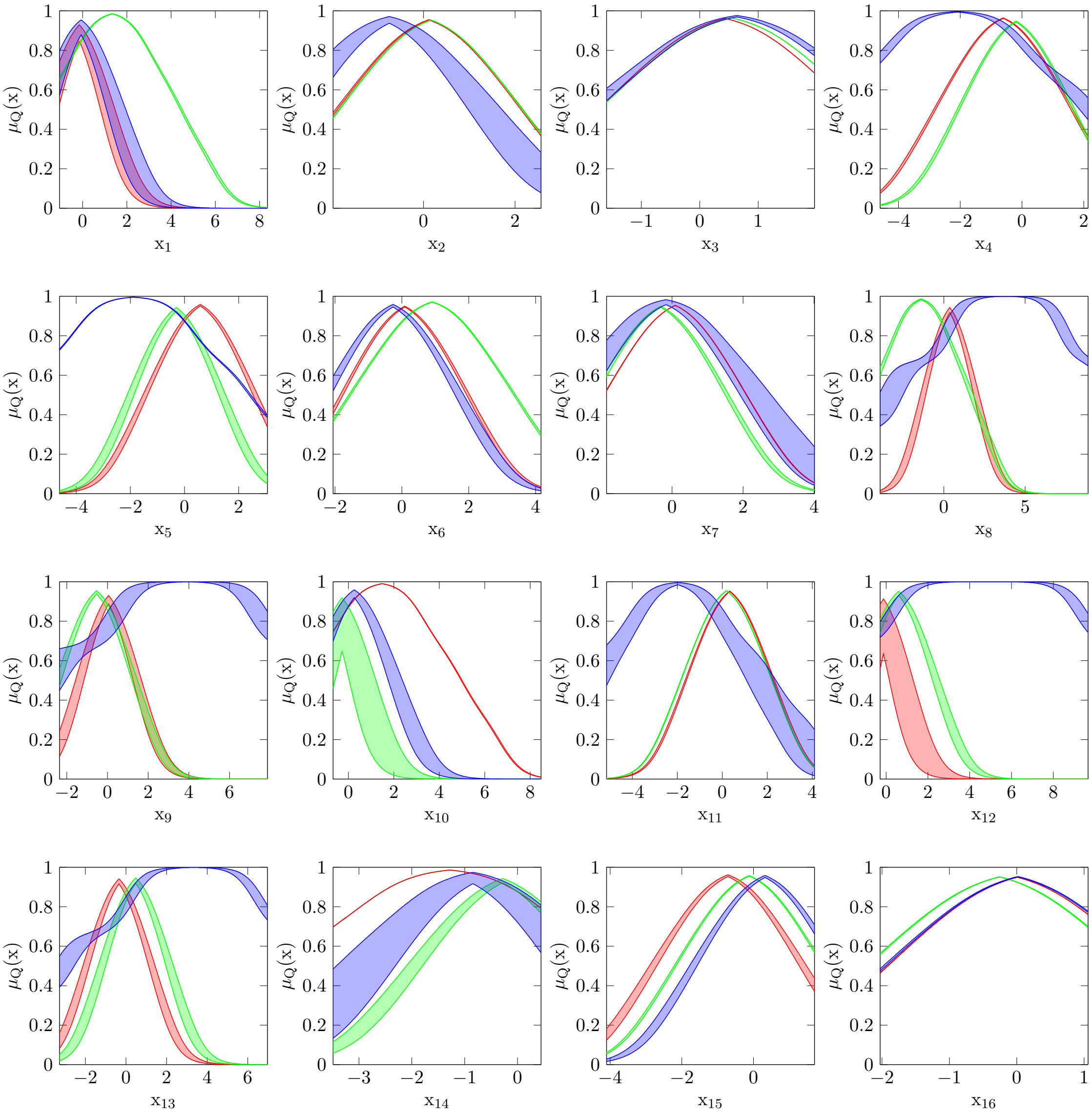


Figure 5.9 Interval Type-2 Quantum membership functions for the Charpy impact data input variables

system is an Interval Type-2 Fuzzy Set:

$$Y(\boldsymbol{x}) = [y_l, y_r] = \int_{f^1 \in \left[\underline{f}^1, \overline{f}^1\right]} \cdots \int_{f^M \in \left[\underline{f}^M, \overline{f}^M\right]} 1 \Big/ \frac{\sum_{i=1}^{M} f^i y^i}{\sum_{i=1}^{M} f^i} \tag{5.17}$$

This was type-reduced using the efficient Nie-Tan type reduction [60]:

$$y = \frac{\sum_{n=1}^{N} y^n \left(\underline{f}^n + \overline{f}^n\right)}{\sum_{n=1}^{N} \left(\underline{f}^n + \overline{f}^n\right)} \tag{5.18}$$

5.3.4 Parameter Optimisation

The parameters were updated using a genetic algorithm. Genetic algorithms are part of the broader fields of evolutionary algorithms and evolutionary computation. They are modelled after natural evolutionary mechanisms and use stochastic search strategies to optimise nonlinear systems. They were developed during the 1970s by Holland [32] and further improved by Goldberg [25].

Figure 5.10 shows a flowchart of the genetic algorithm. The algorithm starts by encoding the optimisation parameters as a chromosome. For the current scenario, the chromosome consisted of the following parameters:

- TSK biases;
- TSK weights;
- Membership function centres;
- LMF Quantum intervals;
- UMF Quantum intervals.

While Holland [32] first suggested the use of a binary encoded representation for the chromosome, other studies have shown that alternative representations such as the direct use of integers or real numbers are also possible, provided that genetic operators supporting the representation are available [22].

The algorithm iterates through a number of steps, modifying a population of chromosomes. The population was set to 1000 chromosomes to allow for the necessary genetic diversity, given the high number of parameters to be optimised.

The next step is to evaluate the fitness of each chromosome in the population. The fitness function returns a measure of the suitability of a solution to the problem. The algorithm also makes use an objective function, whose outputs are scaled through the fitness function. For

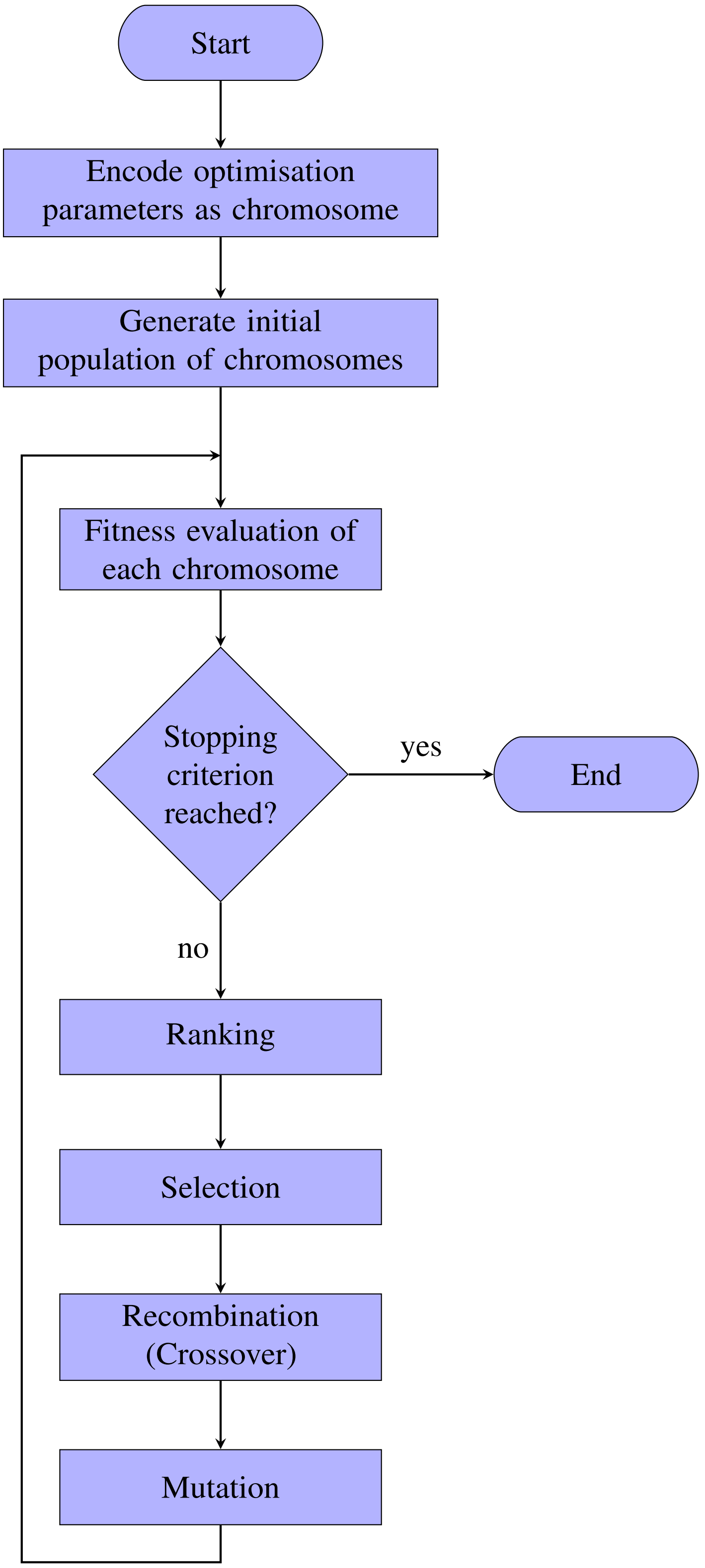

Start
Encode optimisation parameters as chromosome
Generate initial population of chromosomes
Fitness evaluation of each chromosome
Stopping criterion reached?
yes
End
no
Ranking
Selection
Recombination (Crossover)
Mutation

Figure 5.10 Flowchart of the Genetic Algorithm

the current problem the objective function returns the RMSE of the model on the training data using the model parameters of a particular chromosome.

After checking the stopping criteria, the algorithm selects the parents based on their fitness to create the next generation. The default selection algorithm in the MATLAB genetic algorithm is stochastic uniform selection. This ranks parents on a line with the length of their segment corresponding to their scaled fitness. Parents are then chosen at evenly spaced intervals, with lower fitness individuals having a chance to be chosen as well.

Apart from the 'elite', a proportion of the population corresponding to the crossover fraction is conceived by crossover between pairs of parents. The rest of the population are children formed from parent chromosomes by mutation. Elitism is where the fittest members of the population are directly propagated to the next generation. These genetic operators were set as follows:

- Elite: 0.05 × population size;
- Crossover: 0.8 × (population size – elite);
- Mutation: 0.2 × (population size – elite).

The algorithm then iterates through the outlined steps for each generation. The termination condition can be made of a number of options, including [77]:

- No significant improvement in successive generations;
- The objective function evaluation becomes equal to a certain value;
- An upper limit of the number of generations is reached.

The MATLAB parallel computing toolbox was used to run the simulations on 4 parallel workers on the local machine. This allows the fitness function to be executed in parallel, thus reducing the optimisation time.

5.4 Simulation Details

Simulations were run with a Type-2 TSK fuzzy system consisting of 3 or 4 rules and optimised using a genetic algorithm with a generation limit between 500 and 2500.
Simulations 1 to 7 had the following common parameters:

- 3 fuzzy rules;
- 387 optimisation parameters;

- population of 1000 chromosomes.

Simulations 8 to 16 had the following common parameters:

- 4 fuzzy rules;
- 516 optimisation parameters;
- population of 1000 chromosomes.

These are details for the simulations regarding the initialisation of the membership functions and the objective function used for the optimisation:

- Simulation 1:
  - ∗ 3 quantum levels per membership function initialised using the covariance from Type-2 fuzzy clustering: $w_{min}(j,i) \times [1;2;3]$ and $w_{max}(j,i) \times [1;2;3]$.
- Simulation 2:
  - ∗ Same as Simulation 1 but using the final population from the last generation of Simulation 1 as the initial population.
- Simulation 3:
  - ∗ 3 quantum levels per membership function initialised using the covariance from Type-2 fuzzy clustering: $sqrt(w_{min}(j,i)) \times [1;2;3]$ and $sqrt(w_{max}(j,i)) \times [1;2;3]$.
- Simulation 4:
  - ∗ Same as Simulation 3 but using validation dataset as well with cost function: $J = (J_1 \times J_2)/(J_1 + J_2) + J_1^2 + J_2^2$ where J_1 = RMSE training data and J_2 = RMSE validation data.
  - ∗ RMSE testing data = 21.23 after removal of erratic point (point 296 in testing dataset).
- Simulation 5:
  - ∗ Same as Simulation 4 but using the final population of Simulation 4 as the initial population.
  - ∗ RMSE testing data = 21.20 after removal of erratic point (point 296 in testing dataset).

- Simulation 6:
  - $*$ Same as Simulation 4 but using the final population of Simulation 4 as the initial population (1000 generations more not 500).
  - $*$ RMSE testing data = 21.27 after removal of erratic point (point 296 in testing dataset).
- Simulation 7:
  - $*$ Same as Simulation 3.
- Simulation 8:
  - $*$ 3 quantum levels per membership function initialised using the covariance from Type-2 fuzzy clustering: $sqrt(w_{min}(j,i)) \times [1;2;3]$ and $sqrt(w_{max}(j,i)) \times [1;2;3]$.
- Simulation 9:
  - $*$ Same as Simulation 8.
- Simulation 10:
  - $*$ Same as Simulation 8.
- Simulation 11:
  - $*$ Same as Simulation 8.
- Simulation 12:
  - $*$ Same as Simulation 8.
- Simulation 13:
  - $*$ Same as Simulation 8.
- Simulation 14:
  - $*$ Same as Simulation 8 but using validation set as well with cost function: $J = (J_1 \times J_2)/(J_1 + J_2) + J_1^2 + J_2^2$ where J_1 = RMSE training data and J_2 = RMSE validation data.
- Simulation 15:
  - $*$ Same as Simulation 14.

- Simulation 16:
  - ∗ Same as Simulation 8.

5.5 Modelling Results

The results of these simulations are shown in Tables 5.1 and 5.2 respectively. Figures 5.11 to 5.26 show the mean and best fitness plots for the optimisation procedures.

The chosen model has 4 rules with the results presented in Table 5.3. The results in this table indicate that the model shows good generalisation performance across the datasets. Apart from the RMSE, Table 5.3 also gives the Pearson correlation coefficients between the measured and predicted outputs which are also shown in Figure 5.27.

As indicated in Table 5.4, the Interval Type-2 model provides a performance improvement over the Type-1 model, where, in terms of the RMSE for the testing data, the Interval Type-2 model predicts the output 0.94% better than the Type-1 model.

5.6 Impact Transition Curve

To evaluate the models from a metallurgical perspective it was decided to plot the output of the models while varying the impact test temperature to obtain the impact transition curve according to the Type-1 and Type-2 fuzzy models.

For comparison, the graph obtained by Tenner [83] using a neural network ensemble model is shown in Figure 5.28. Tenner used the 'median' steel for the fixed inputs as tabulated in Table 5.5. These are the values for the most common type of steel in the dataset which is 1%CrMo steel.

Another set of model outputs were obtained by fixing the inputs to the mean value of the variables which are indicated in Table 5.6. The test temperature was varied across its range, this being -59°C to 23°C. The impact energy was obtained as the output of the models and plotted against the impact test temperature.

Figure 5.29 shows the plots obtained from the Type-1 and Type-2 fuzzy models. It was concluded that the plot in the dashed red line is the closest when compared to the plot obtained by Tenner. The graph shows that although the curve is monotonically increasing, its shape is not similar to that of an impact transition curve for steel which is displayed in Figure 5.30 (for the corresponding Carbon content). The impact transition curve for steel has a point of inflexion which represents the ductile-to-brittle transition temperature.

Table 5.1 Simulation results using 3 rules

Simulation	Generations	Optimisation Time	RMSE		
			Training	Validation	Testing
1	1000	~24 hrs	19.58	22.50	20.99
2	500	~12 hrs	18.67	22.41	23.99
3	1000	~23 hrs	18.16	28.98	20.63
4	1000	~28 hrs	21.20	19.41	469.30
5	500	~14 hrs	20.95	19.34	415.81
6	1000	~29 hrs	20.97	19.08	521.04
7	2000	~42 hrs	18.91	22.58	19.64

Table 5.2 Simulation results using 4 rules

Simulation	Generations	Optimisation Time	RMSE		
			Training	Validation	Testing
8	1000	~29 hrs	20.08	22.80	20.94
9	1500	~41 hrs	18.65	20.70	18.90
10	**2000**	**~53 hrs**	**16.71**	**19.13**	**18.00**
11	2000	~54 hrs	16.71	19.13	18.00
12	2000	~53 hrs	16.71	19.13	18.00
13	2500	~67 hrs	18.64	20.97	19.92
14	2000	~72 hrs	17.39	15.45	18.80
15	2000	~72 hrs	17.39	15.45	18.80
16	2500	~71 hrs	18.64	20.97	19.92

Table 5.3 Interval Type-2 fuzzy model performance

	Training data	Validation data	Testing data
Correlation coefficient	0.861	0.813	0.842
RMSE (Joules)	16.71	19.13	18.00

Table 5.4 Type-1 and Interval Type-2 fuzzy model performance comparison

	Type-1 Model	Type-2 Model	Percentage Difference
Training Data RMSE	17.75	16.71	-5.86
Validation Data RMSE	18.84	19.13	+1.54
Testing Data RMSE	18.17	18.00	-0.94

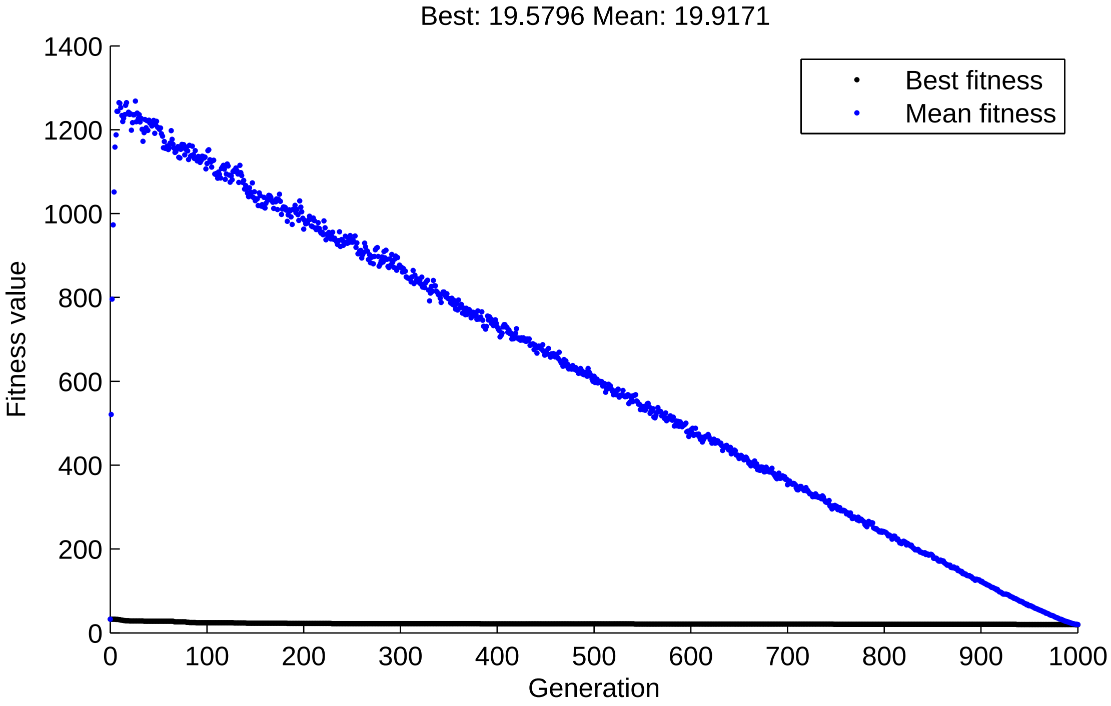


Figure 5.11 Simulation 1

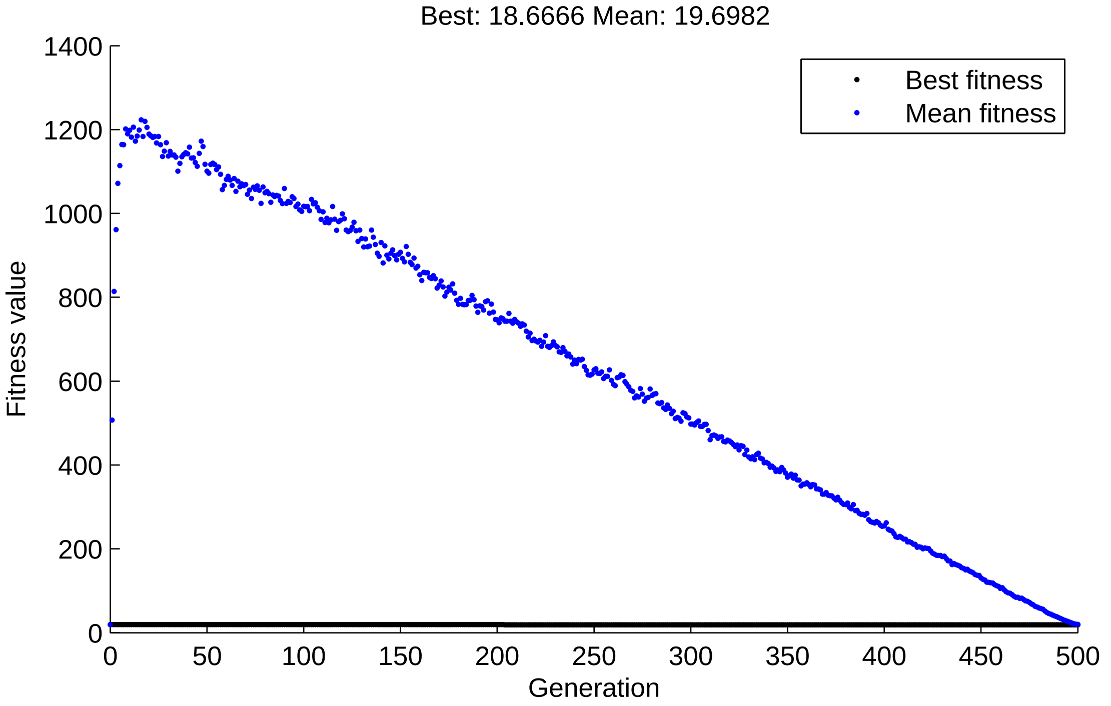


Figure 5.12 Simulation 2

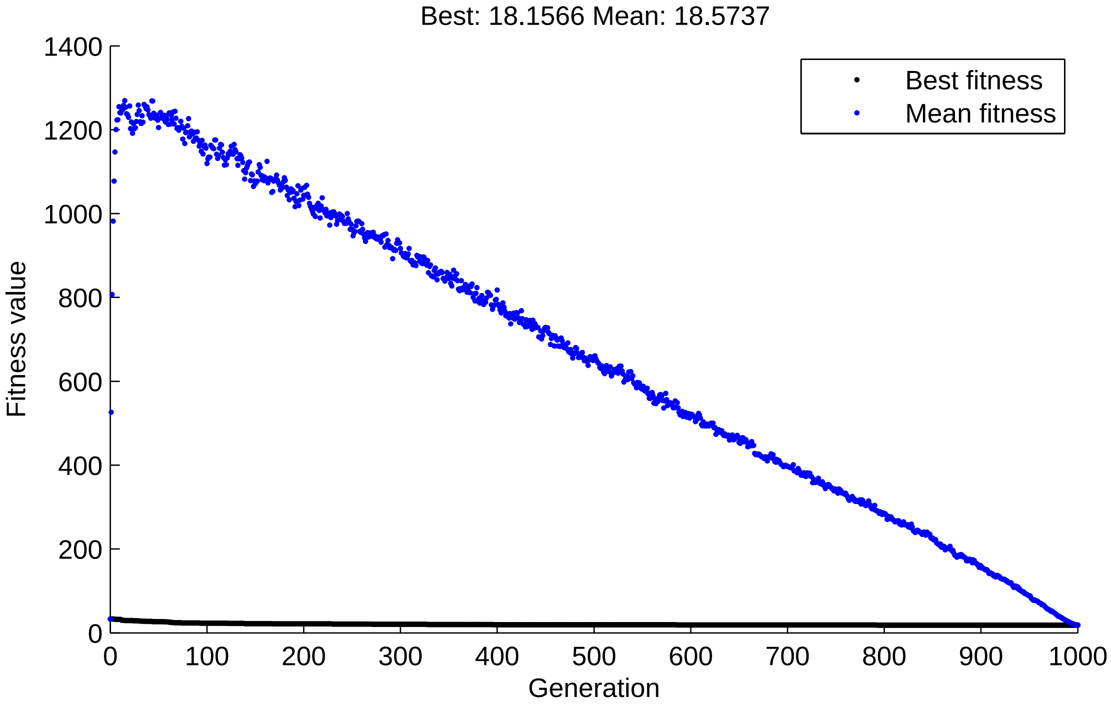


Figure 5.13 Simulation 3

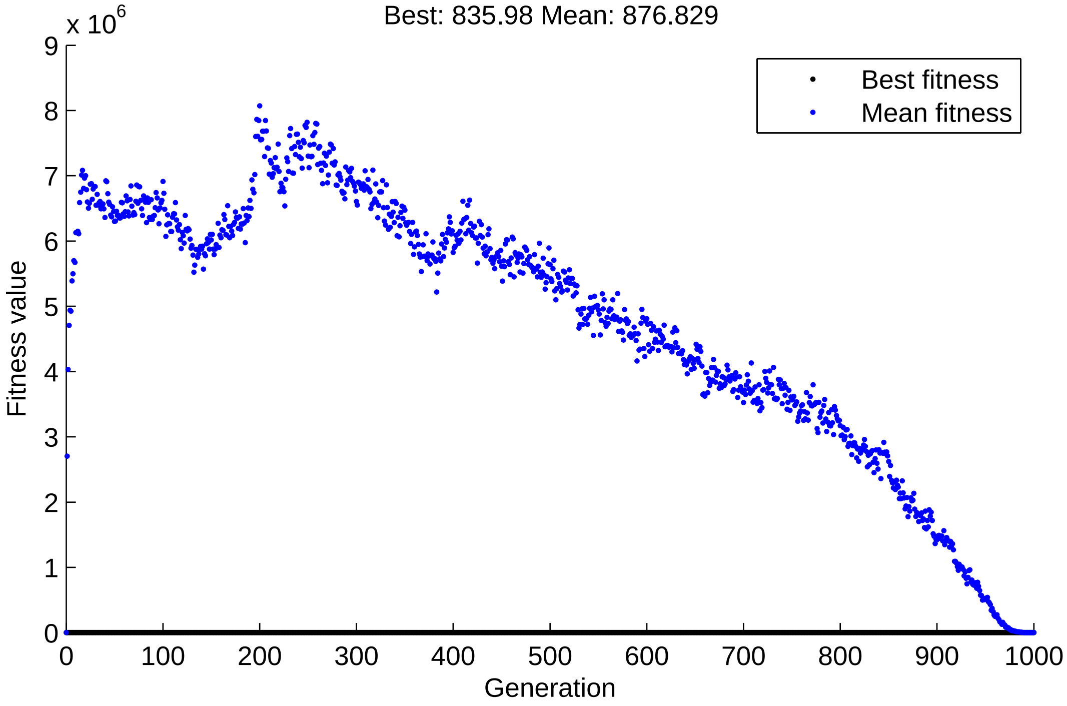


Figure 5.14 Simulation 4

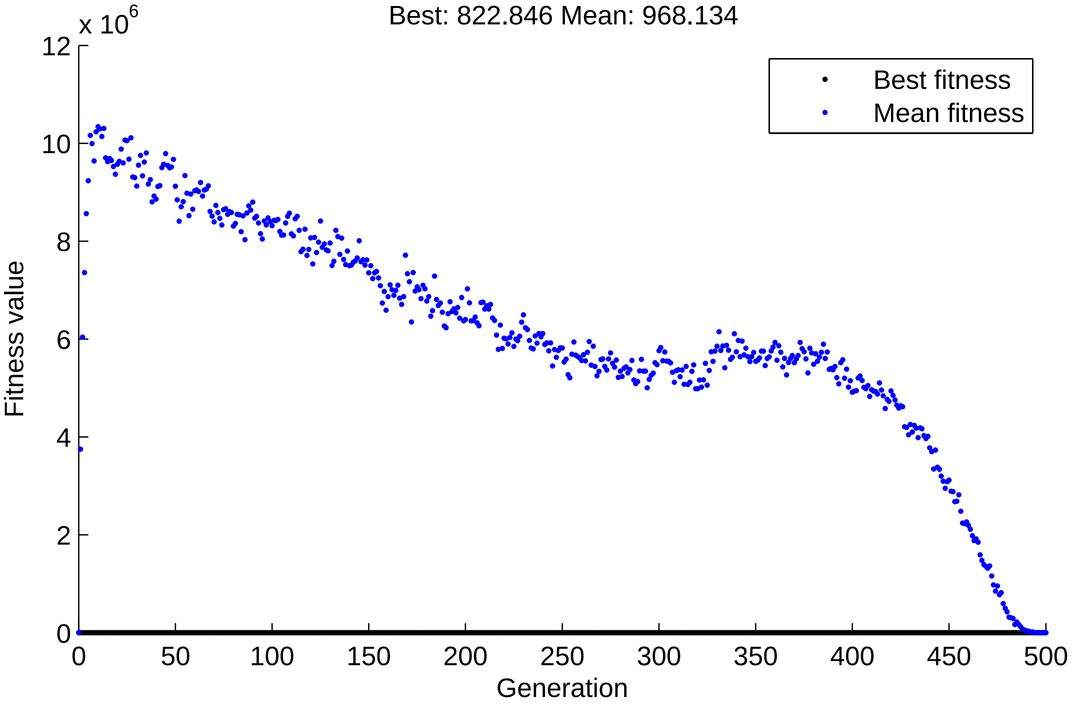


Figure 5.15 Simulation 5

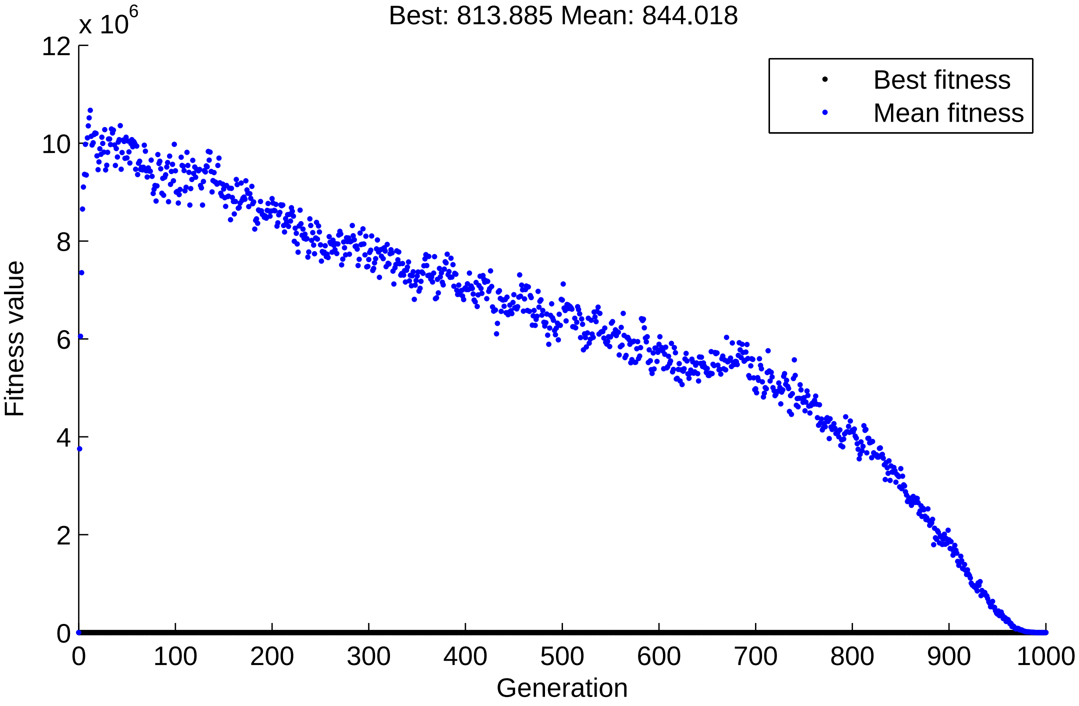


Figure 5.16 Simulation 6

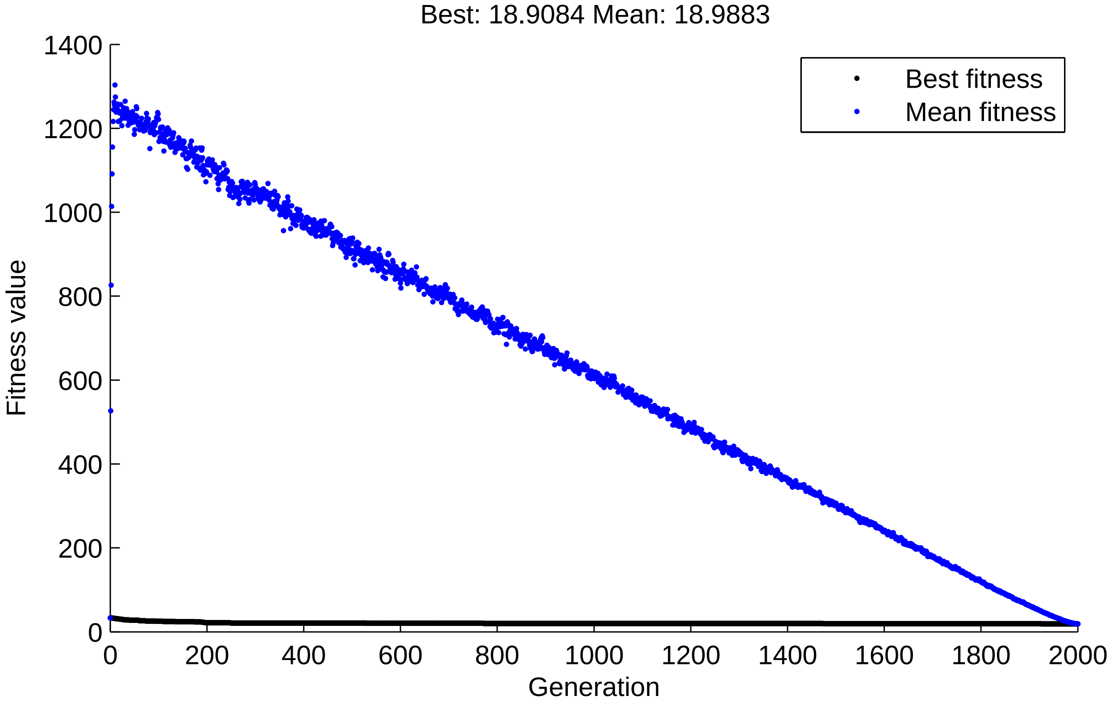


Figure 5.17 Simulation 7

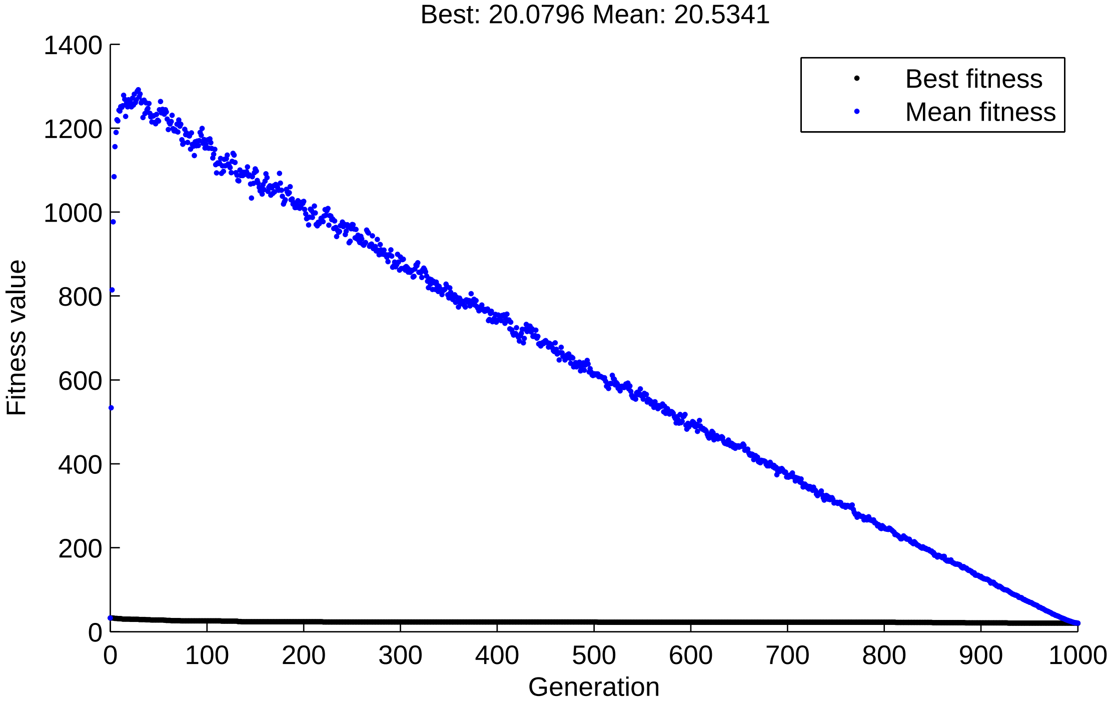


Figure 5.18 Simulation 8

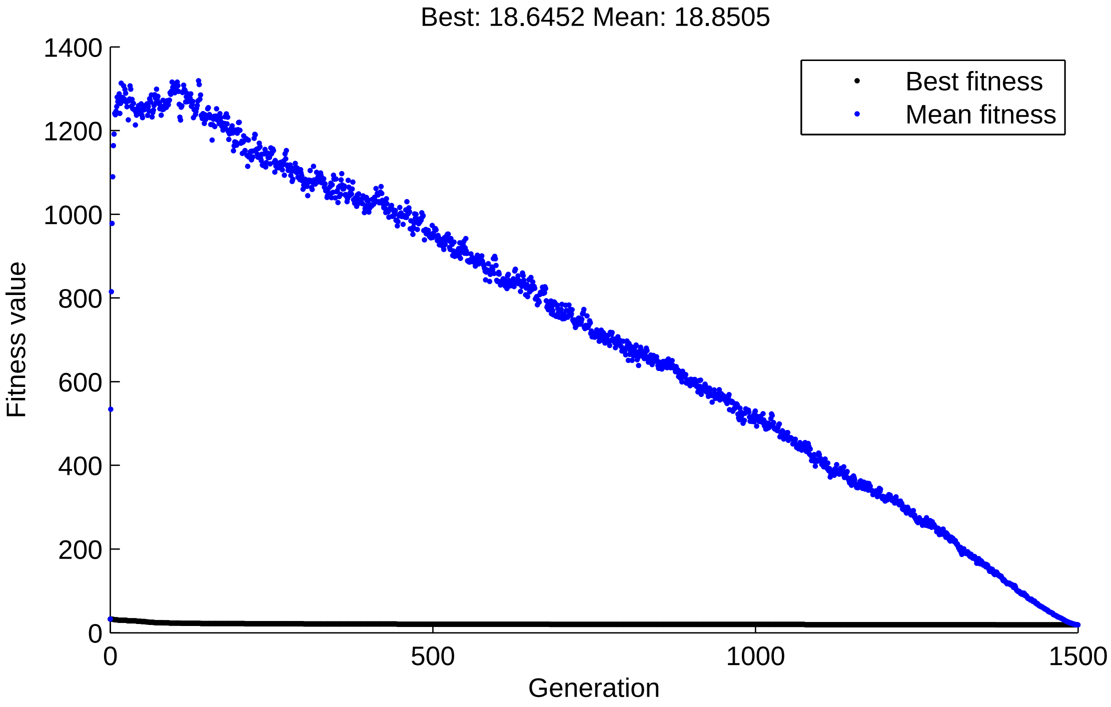


Figure 5.19 Simulation 9

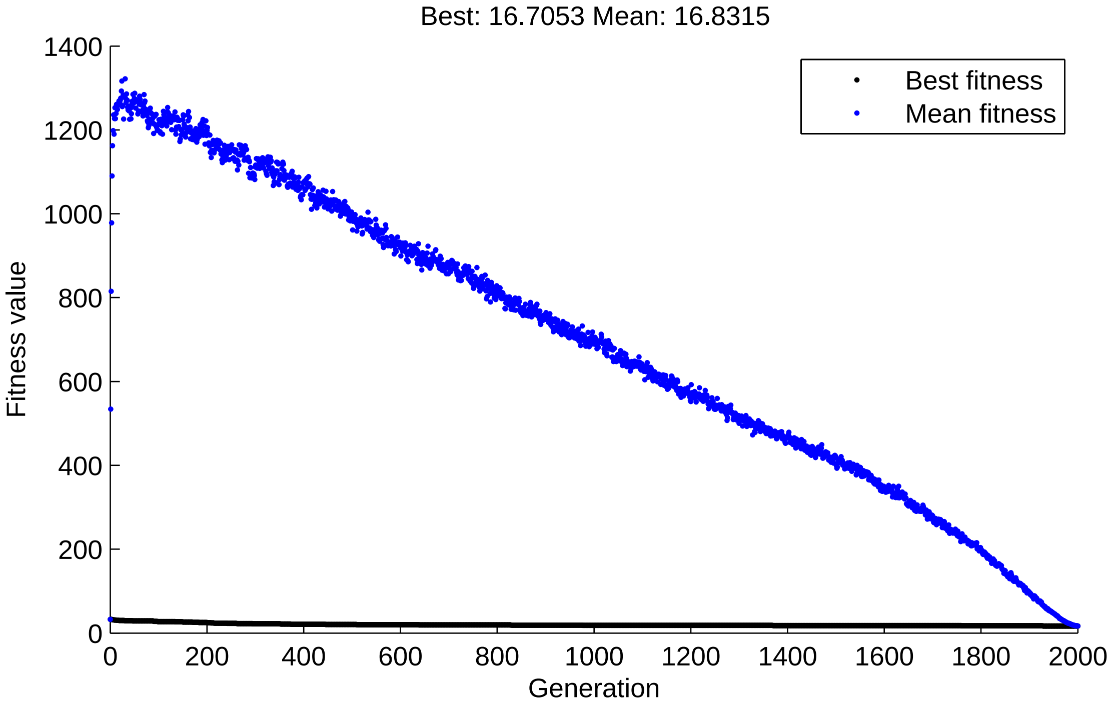


Figure 5.20 Simulation 10

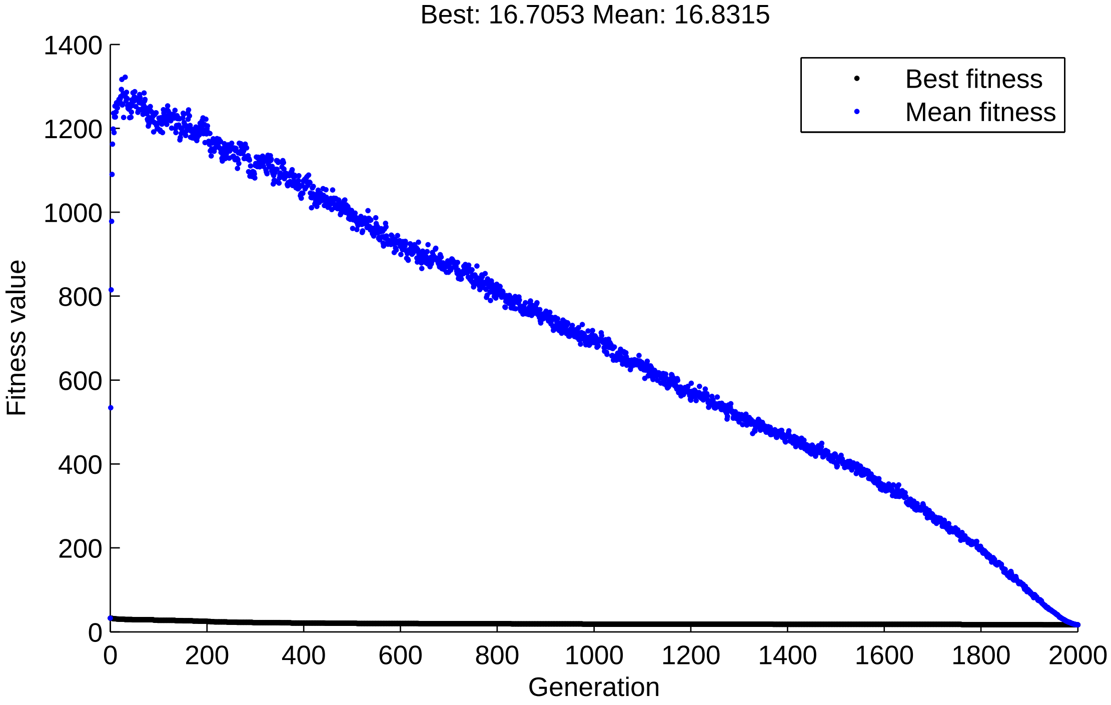


Figure 5.21 Simulation 11

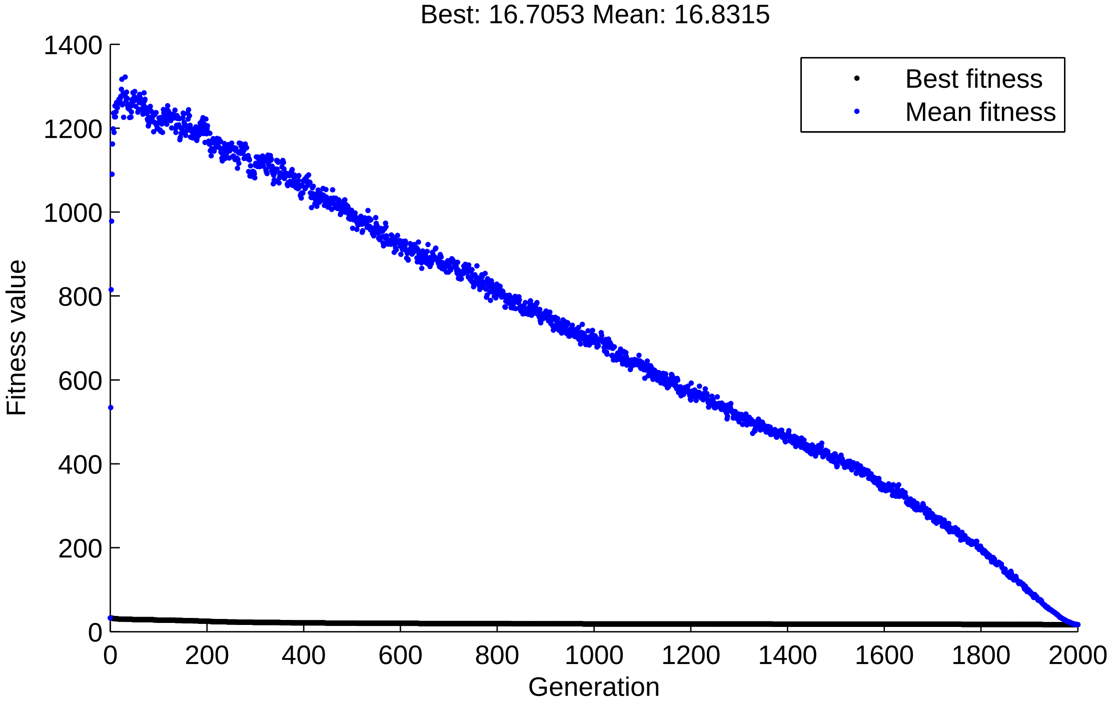


Figure 5.22 Simulation 12

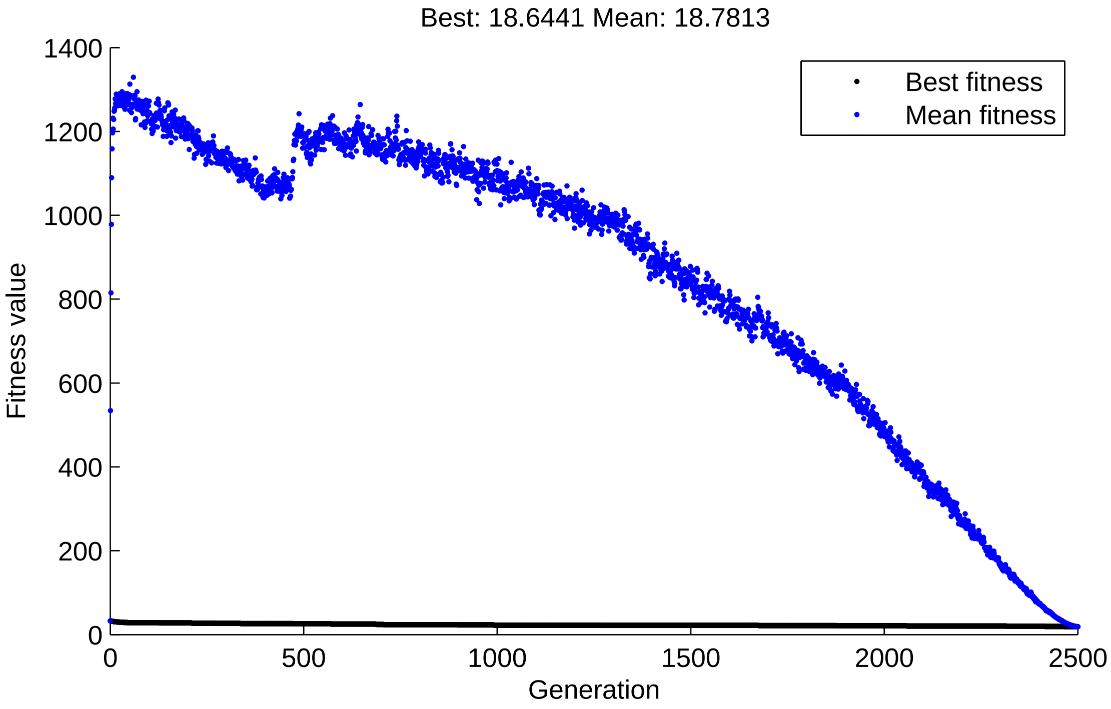
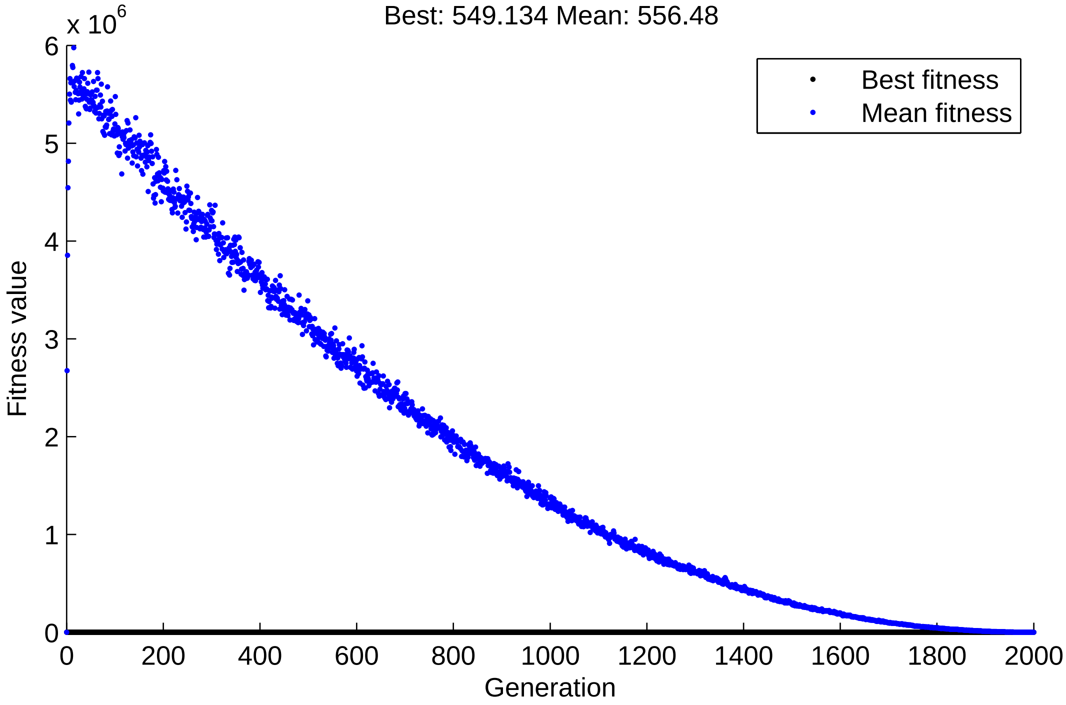


Figure 5.23 Simulation 13



]

Figure 5.24 Simulation 14

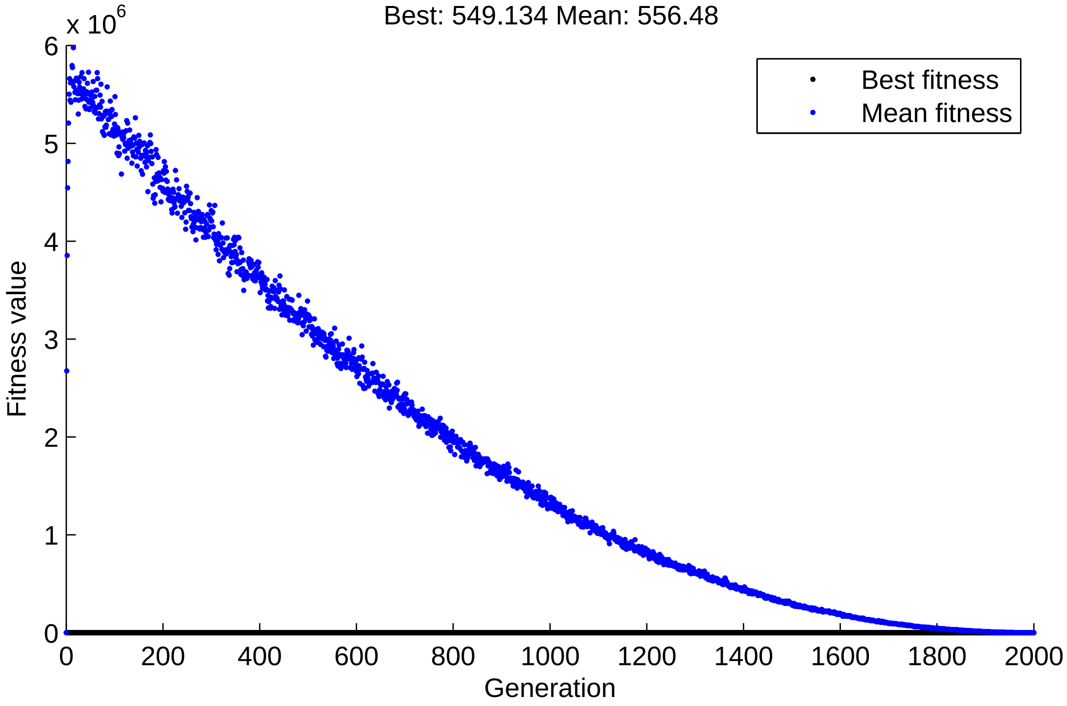


Figure 5.25 Simulation 15

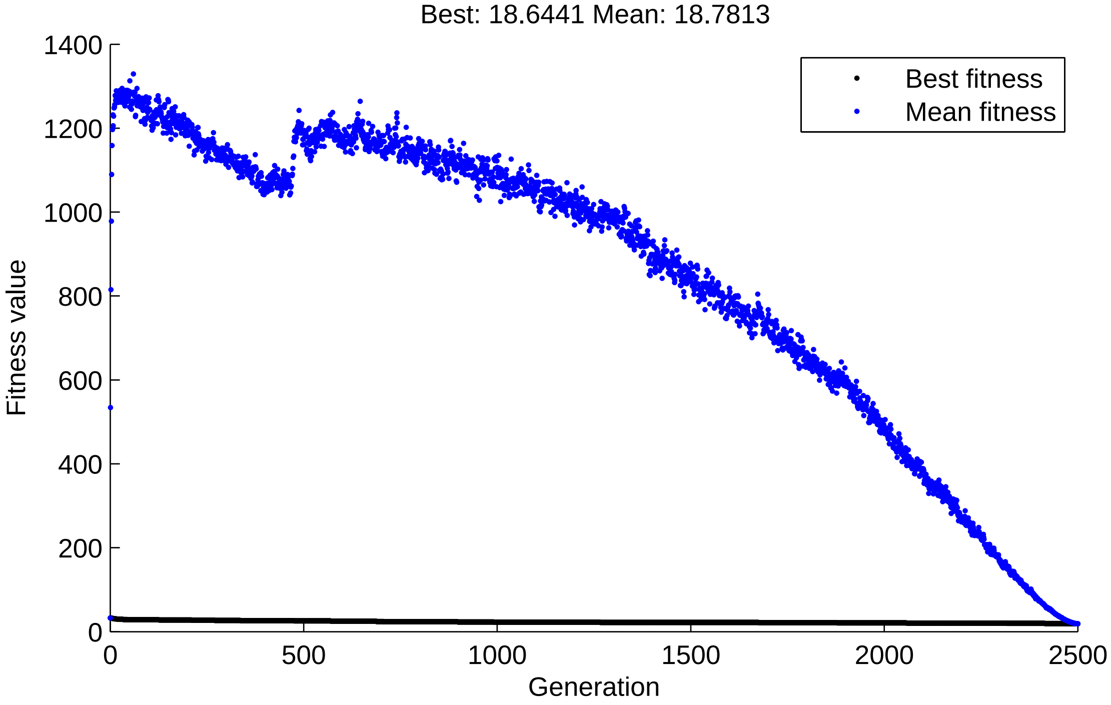
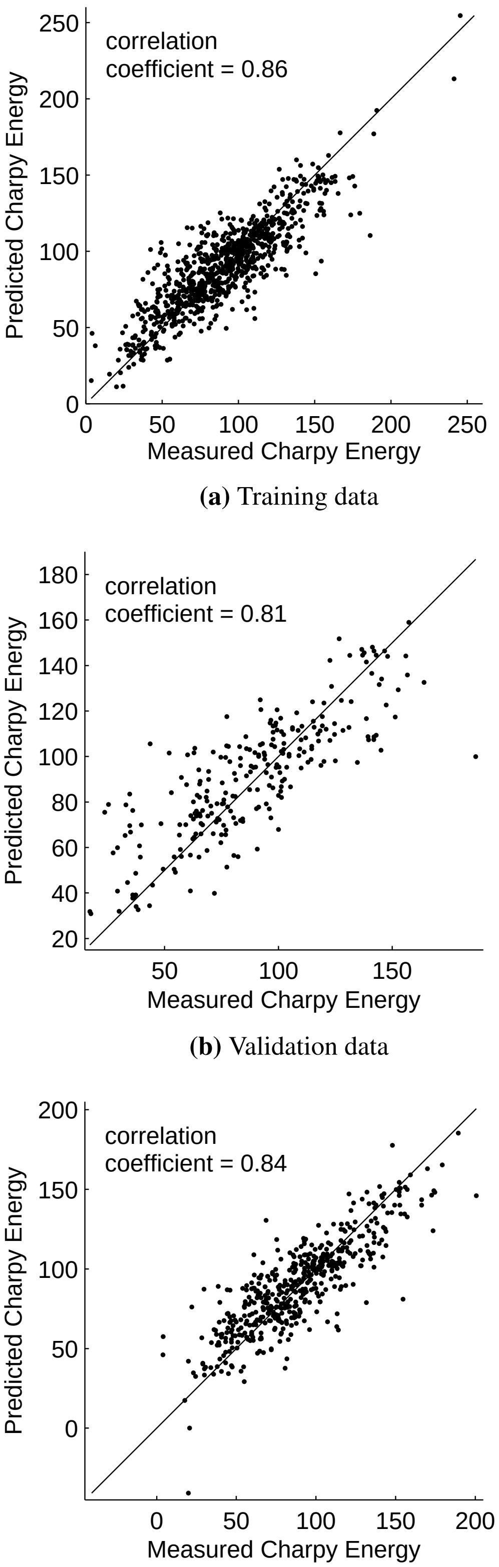


Figure 5.26 Simulation 16



(a) Training data

(b) Validation data

(c) Testing data

Figure 5.27 Charpy energy prediction using the Type-2 fuzzy model

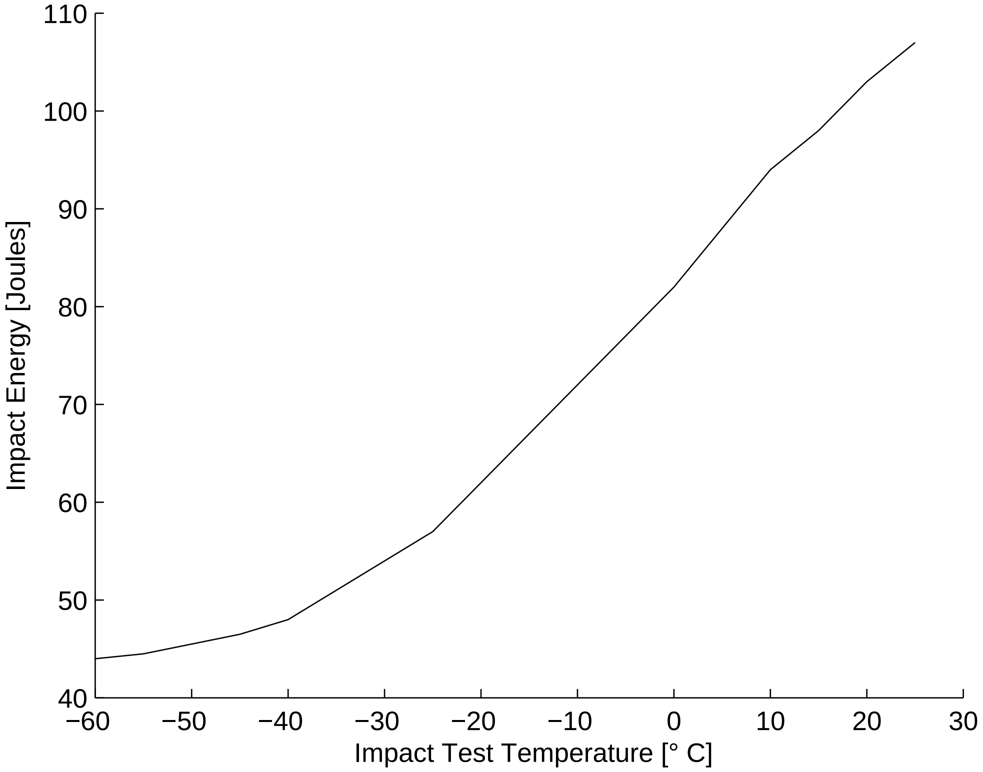


Figure 5.28 Impact transition curve obtained by Tenner [83, p. 237]

One possible reason is that the models average over the different steel types present in the dataset. Another reason is that the shape of the impact transition curve may vary from the ideal, depending on the various compositions of steel [68] and therefore the various values set for the fixed inputs. This is because the shape also depends on the value of other alloys in the steel [38].

5.7 Summary

This chapter has analysed the introduction of Type-2 Fuzzy Sets for the fuzzy model. This formed Interval Type-2 Quantum membership functions which provided more degrees of freedom for the optimisation to work with. The model was optimised using a genetic algorithm and the results show an improvement over the Type-1 fuzzy model. The Type-2 fuzzy model is more generalising as the results were achieved using less rules (4 instead of 6) compared to the Type-1 fuzzy model. Finally, the models derived from the data have been used to plot the effect of varying the impact test temperature on the impact energy.

The following chapter will provide general conclusions about the project together with some recommendations for work that may expand the research undertaken here.

Table 5.5 Input variables fixed at ‘median’ 1%CrMo steel values with varying test temperature

Variable	Value
Test Depth [mm]	12.7
Size [mm]	180
Coded Site	3
C [%]	0.41
Si [%]	0.27
Mn [%]	0.78
S [%]	0.023
Cr [%]	1.08
Mo [%]	0.22
Ni [%]	0.19
Al [%]	0.027
V [%]	0.005
Hardening Temperature [°C]	860
Cooling Medium	-
Tempering Temperature [°C]	630
Test Temperature [°C]	(-59) – 23

Table 5.6 Mean input variables with varying impact test temperature

Variable	Value
Test Depth [mm]	20.80
Size [mm]	172.49
Coded Site	3.80
C [%]	0.39
Si [%]	0.25
Mn [%]	0.84
S [%]	0.02
Cr [%]	1.08
Mo [%]	0.24
Ni [%]	0.37
Al [%]	0.03
V [%]	7.7e-3
Hardening Temperature [°C]	864.02
Cooling Medium	2.77
Tempering Temperature [°C]	647.19
Test Temperature [°C]	(-59) – 23

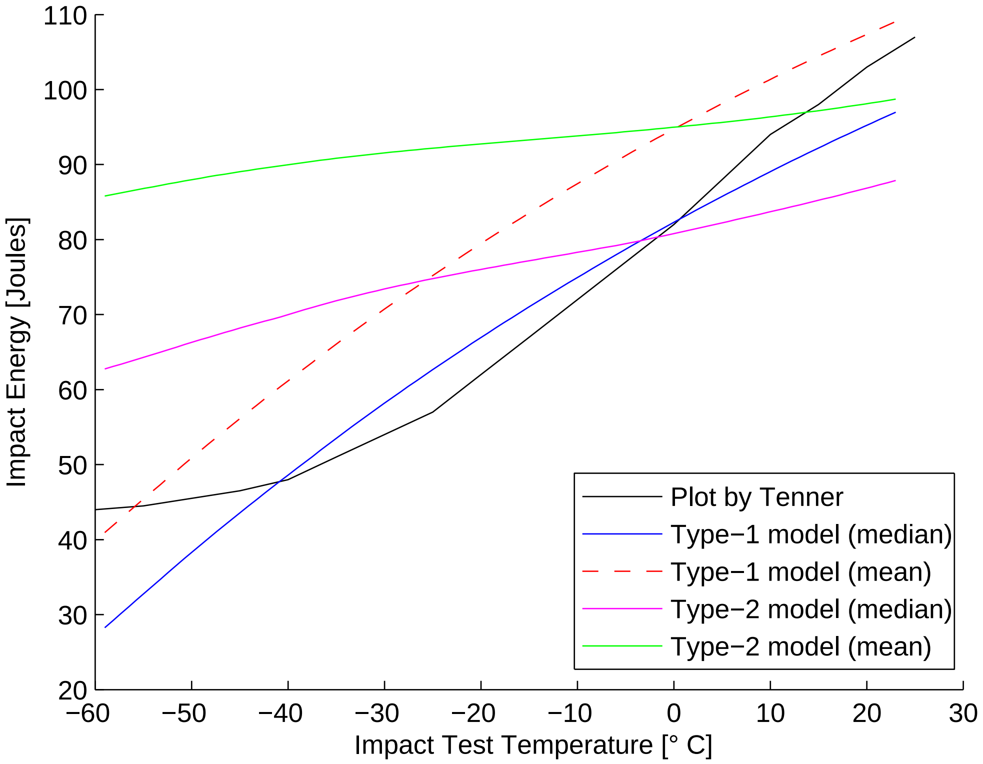


Figure 5.29 Impact energy against test temperature curves from the models

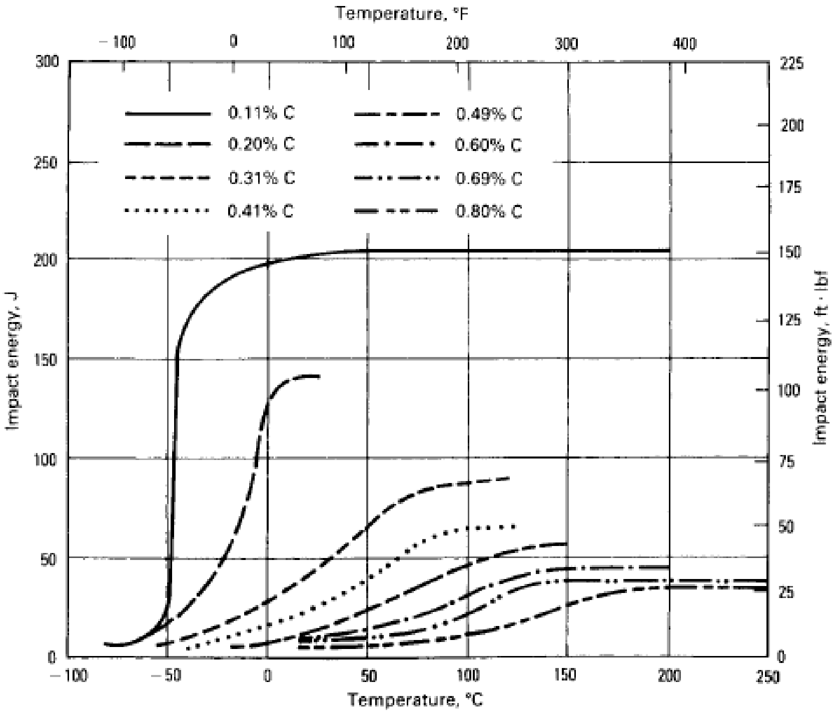


Figure 5.30 Impact energy against test temperature for steel for varying Carbon content [18]

Chapter 6

Conclusions and Future Work

6.1 Conclusions

The main aim of this research has been to apply machine learning techniques to materials property prediction as outlined in Chapter 1.

Chapter 2 gave a deeper insight into the context of the research together with related literature.

Chapter 3 evaluated techniques derived from SVMs for the classification of a dataset from the testing of rails. Although the SVM does not suffer from the local minima problem since its optimisation is formulated as a convex optimisation problem, its performance is severely affected by class imbalance. For this reason, an internal (FSVM with DEC) and external (data under-sampling) class imbalance learning methods were applied to the data. The best performance, with a sensitivity of 67.55%, was obtained when the FSVM was applied to the under-sampled dataset (for the majority class), thus effectively dealing with the imbalance in the data. As quadratic optimisation scales poorly with increasing sample numbers, the techniques were further developed by applying FCM clustering to reduce training time and model complexity by lowering the number of support vectors in the model.

In Chapter 4, the Quantum membership function was proposed as part of a fuzzy modelling framework based on the ANFIS. This was used to predict the Charpy impact energy of steel samples, with promising results compared to those found in the literature. This chapter demonstrated the advantages of a synergistic combination of modelling techniques from various fields in terms of capturing uncertainty, learning ability and interpretability.

The uncertainty handling and generalising capabilities of the model were enhanced in Chapter 5, where Interval Type-2 Fuzzy Sets were integrated with the Quantum membership functions to improve the results obtained in Chapter 4. In fact, an RMSE of 18.0 Joules was obtained when predicting the testing data. Comparing Table 5.3 with Table 4.12 shows that this is 0.72% better than the best result found in the literature when modelling the same dataset.

It is important to note the relevance of fuzzy logic when implementing these modelling techniques:

- FCM clustering (Type-1 and Type-2 fuzzy clustering) was used to identify the structure of the models to efficiently model the data.
- The effect that the inclusion of the fuzzy membership vector has on the SVM techniques is that the model becomes less sensitive to parameters C and σ. This can be observed from the grid search plots in Section 3.4 when the corresponding plots of SVM and FSVM are compared. This shows that the fuzzy membership captures some of the noise in the data and therefore makes the model more robust and generalising by being less sensitive to changes in model parameters.
- In the case of the Quantum membership function model, the Quantum function was suitable as a fuzzy membership function not only because of the extra degrees of freedom but also because it inherently deals with outlying data points. For example a Quantum membership function may use the quantum levels to keep the membership low for a number of outlying data points. This property is especially relevant when modelling a complex dataset such as the Charpy impact dataset analysed in this research.
- Type-2 fuzzy sets offer more parameters for optimisation and therefore more degrees of freedom. The Type-2 fuzzy model uses 4 rules compared to 6 rules for the Type-1 model. This not only makes the model more compact, but also improves the generalisation ability and transparency of the model.
- The combination of Quantum membership functions and Type-2 Fuzzy Sets worked well, in the sense that Quantum functions were able to form feasible Interval Type-2 Fuzzy Sets.

6.2 Recommendations for Future Work

With a critical view of the work presented in this thesis, further research can be suggested:

- Extend the SVM external imbalance learning method by including data over-sampling of the minority class. This could be done by combining both over-sampling, such as using the Synthetic Minority Over-sampling Technique (SMOTE), and under-sampling, as suggested by Chawla et al. [11]. In this manner, loss of information due to undersampling is minimised, while also helping the classifier to learn the correct decision boundary due to training on a balanced dataset.

- Modify the SVM techniques, developed in Chapter 3 for classification, to be applied to the regression problem of Charpy impact energy prediction. Support Vector Regression (SVR) follows the same basic principles as SVMs but the definition of the problem takes into account that the error for each data point is within a certain threshold. One of the methods to work with continuous values (instead of classification into classes) is by using an ε-insensitive loss function when defining the problem [79].

- Include confidence bounds on the predictions either by characterisation of the modelling error or by using the intervals from the non-reduced fuzzy sets of the Type-2 fuzzy model. This would assist the model in being more accurate in its predictions as it would indicate the level of confidence with which predictions in certain areas are being made.

- As opposed to Interval Type-2 Fuzzy Sets, General Type-2 Fuzzy Sets are better at capturing uncertainty and therefore their use would be beneficial in dealing with high problem uncertainty. However, research is still ongoing on how to reduce the computational complexity necessary to perform calculations when working with these Fuzzy Sets, i.e. type reduction and defuzzification.

- As noted in Section 5.6, more work is necessary to ensure that the model does not overfit the data and to assess its metallurgical significance. Since the focus is on Charpy energy prediction, this is especially relevant with regards to the impact transition curve.

References

[1] Agrawal, B. K. (1988). *Introduction to engineering materials*. Tata McGraw-Hill Education.

[2] Akbani, R., Kwek, S., and Japkowicz, N. (2004). Applying support vector machines to imbalanced datasets. *Machine learning: ECML 2004*, pages 39–50.

[3] Anderson, E. (1936). The species problem in iris. *Annals of the Missouri Botanical Garden*, 23(3):457–509.

[4] Batuwita, R. and Palade, V. (2013). Class imbalance learning methods for support vector machines. In He, H. and Ma, Y., editors, *Imbalanced Learning: Foundations, Algorithms, and Applications*, pages 83–99. John Wiley & Sons, Inc., Hoboken, NJ.

[5] Bezdek, J. C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA.

[6] Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer, New York.

[7] Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM.

[8] Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167.

[9] Callister, W. D. and Rethwisch, D. G. (2013). *Materials Science and Engineering: An Introduction*. John Wiley & Sons, 9th edition.

[10] Cervantes, J., Li, X., and Yu, W. (2006). Support vector machine classification based on fuzzy clustering for large data sets. In *MICAI*, volume 6, pages 572–582. Springer.

[11] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of artificial intelligence research*, 16:321–357.

[12] Chen, M.-Y. and Linkens, D. A. (2006). Impact toughness prediction for TMCP steels using knowledge-based neural-fuzzy modelling. *ISIJ international*, 46(4):586–590.

[13] Choi, B.-I. and Rhee, F. C.-H. (2009). Interval type-2 fuzzy membership function generation methods for pattern recognition. *Information Sciences*, 179(13):2102–2122.

[14] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

[15] Cox, E. (1999). *The fuzzy systems handbook: A practitioner's guide to building, using, and maintaining fuzzy systems*. AP Professional, San Diego, CA, USA, 2nd edition.

[16] Cristianini, N. and Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press.

[17] Davies, L., Denial, M. J., Locke, A. W., and Reay, M. E. (1981). *Investigating Chemistry*. Heinemann Educational Books, 2nd edition.

[18] Davis, J. R. (2001). *Alloying: understanding the basics*. ASM international.

[19] Deo, R. C. (2015). Machine learning in medicine. *Circulation*, 132(20):1920–1930.

[20] Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern Classification*. Wiley-Interscience publication. John Wiley & Sons, New York, 2nd edition.

[21] Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188.

[22] Fleming, P. and Purshouse, R. (2002). Evolutionary algorithms in control systems engineering: a survey. *Control Engineering Practice*, 10(11):1223–1241.

[23] Fletcher, T. (2009). Support vector machines explained. *University College London, London*.

[24] Ghosh, A. and Chatterjee, A. (2008). *Iron making and steelmaking: theory and practice*. PHI Learning Pvt. Ltd.

[25] Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[26] Gustafson, D. E. and Kessel, W. C. (1978). Fuzzy clustering with a fuzzy covariance matrix. In *Decision and Control including the 17th Symposium on Adaptive Processes, 1978 IEEE Conference on*, pages 761–766. IEEE.

[27] Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182.

[28] Guyon, I., Matic, N., and Vapnik, V. (1996). Discovering informative patterns and data cleaning. In Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., editors, *Advances in Knowledge Discovery and Data Mining*, pages 181–203. American Association for Artificial Intelligence, Menlo Park, CA, USA.

[29] Haykin, S. (2009). *Neural networks and learning machines*. Pearson, Upper Saddle River, NJ, USA, 3rd edition.

[30] Hertzberg, R. W., Vinci, R. P., and Hertzberg, J. L. (2012). *Deformation and Fracture Mechanics of Engineering Materials*. John Wiley & Sons, Inc., 5th edition.

[31] Hill, G. C. and Holman, J. S. (1983). *Chemistry in context*. Thomas Nelson & Sons Ltd., 2nd edition.

[32] Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. University of Michigan Press, Ann Arbor, MI, USA.

[33] Hwang, C. and Rhee, F. C.-H. (2007). Uncertain fuzzy clustering: Interval type-2 fuzzy approach to C-means. *IEEE Transactions on Fuzzy Systems*, 15(1):107–120.

[34] Hwang, T. (2018). Computational power and the social impact of artificial intelligence. *SSRN Electronic Journal*.

[35] Jang, J.-S. R. (1993). ANFIS: Adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, 23(3):665–685.

[36] Jang, J.-S. R., Sun, C.-T., and Mizutani, E. (1997). *Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence*. Prentice Hall, Upper Saddle River, NJ.

[37] John, R. and Coupland, S. (2007). Type-2 fuzzy logic: A historical view. *IEEE computational intelligence magazine*, 2(1):57–62.

[38] Kah, P., Layus, P., Martikainen, J., et al. (2015). Influence of alloying elements on the low-temperature properties of steel. In *The Twenty-fifth International Ocean and Polar Engineering Conference*. International Society of Offshore and Polar Engineers.

[39] Karnik, N. N. and Mendel, J. M. (2001). Centroid of a type-2 fuzzy set. *Information Sciences*, 132(1):195–220.

[40] Klir, G. and Wierman, M. J. (1998). *Uncertainty-based Information: Elements of Generalized Information Theory*. Physica-Verlag, Heidelberg.

[41] Kreinovich, V., Nguyen, H. T., and Yam, Y. (1999). Fuzzy systems are universal approximators for a smooth function and its derivatives. *International Journal of Intelligent Systems*.

[42] Kretzschmar, R., Bueler, R., Karayiannis, N. B., and Eggimann, F. (2000). Quantum neural networks versus conventional feedforward neural networks: an experimental study. In *Neural Networks for Signal Processing X, 2000. Proceedings of the 2000 IEEE Signal Processing Society Workshop*, volume 1, pages 328–337. IEEE.

[43] Liang, Q. and Mendel, J. M. (2000). Interval type-2 fuzzy logic systems: theory and design. *IEEE Transactions on Fuzzy systems*, 8(5):535–550.

[44] Lin, C.-F. and Wang, S.-D. (2002). Fuzzy support vector machines. *IEEE Transactions on neural networks*, 13(2):464–471.

[45] Lin, C.-J., Chen, C.-H., and Lee, C.-Y. (2004). A self-adaptive quantum radial basis function network for classification applications. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 4, pages 3263–3268. IEEE.

[46] Lin, C.-J., Chung, I.-F., and Chen, C.-H. (2007). An entropy-based quantum neuro-fuzzy inference system for classification applications. *Neurocomputing*, 70(13):2502–2516.

[47] Lont, M. (2000). The determination of uncertainties in charpy impact testing. *UNCERT COP*, 6.

[48] Mahfouf, M., Yang, Y. Y., and Zhang, Q. (2009). Characterisation of model error for charpy impact energy of heat treated steels using probabilistic reasoning and a gaussian mixture model. *IFAC Proceedings Volumes*, 42(23):225–230.

[49] Mazumdar, D. and Evans, J. W. (2009). *Modeling of steelmaking processes*. CRC Press.

[50] McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

[51] McNeil, I. (2002). *An encyclopedia of the history of technology*. Routledge.

[52] Mendel, J. M. (2001). *Uncertain rule-based fuzzy logic systems: Introduction and new directions*. Prentice Hall, Upper Saddle River, NJ.

[53] Mendel, J. M. (2007). Type-2 fuzzy sets and systems: An overview. *IEEE Computational Intelligence Magazine*, 2(2):20–29.

[54] Mendel, J. M. and John, R. B. (2002). Type-2 fuzzy sets made simple. *IEEE Transactions on fuzzy systems*, 10(2):117–127.

[55] Mendel, J. M., John, R. I., and Liu, F. (2006). Interval type-2 fuzzy logic systems made simple. *IEEE transactions on fuzzy systems*, 14(6):808–821.

[56] Meyers, M. and Chawla, K. (2009). *Mechanical Behavior of Materials*. Cambridge University Press.

[57] Moore, C. and Marshall, R. I. (1980). Modern steelmaking methods. *The Institution of Metallurgists*.

[58] Moore, C. and Marshall, R. I. (1991). *Steelmaking*. Institute of Materials.

[59] Moosbrugger, C., editor (2002). *Atlas of Stress-strain Curves*. ASM International, OH, USA, 2$^{\text{nd}}$ edition.

[60] Nie, M. and Tan, W. W. (2008). Towards an efficient type-reduction method for interval type-2 fuzzy logic systems. In *2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence)*, pages 1425–1432.

[61] Obajemu, O., Mahfouf, M., and Torres-Salomao, L. (2014). A new interval type-2 fuzzy clustering algorithm for interval type-2 fuzzy modelling with application to heat treatment of steel. *IFAC Proceedings Volumes*, 47(3):10658–10663.

[62] Panoutsos, G. and Mahfouf, M. (2005). Discovering knowledge and modelling systems using granular computing and neurofuzzy structures. In *NiSIS'05, European Symposium on Nature-Inspired Smart Information Systems*.

[63] Panoutsos, G. and Mahfouf, M. (2008). Modelling imprecise and scattered multidimensional data using granular data compression and multiple granularity modelling. In *Granular Computing, 2008. GrC 2008. IEEE International Conference on*, pages 512–517. IEEE.

[64] Panoutsos, G. and Mahfouf, M. (2010). A neural-fuzzy modelling framework based on granular computing: Concepts and applications. *Fuzzy Sets and Systems*, 161(21):2808–2830.

[65] Parker, R. and United States Ship Structure Committee (1957). *Brittle behavior of engineering structures*. Wiley.

[66] Passino, K. M., Yurkovich, S., and Reinfrank, M. (1998). *Fuzzy control*. Addison-Wesley, Menlo Park, CA.

[67] Peacey, J. G. and Davenport, W. G. (1979). *The iron blast furnace: theory and practice*. Pergamon Press.

[68] Pisarski, H. G., Hayes, B., Olbricht, J., Lichter, P., and Wiesner, C. S. (2002). Validation of idealised charpy impact energy transition curve shape. In François, D. and Pineau, A., editors, *From Charpy To Present Impact Testing*, volume 30 of *European Structural Integrity Society*, pages 333 – 340. Elsevier.

[69] Purushothaman, G. and Karayiannis, N. B. (1997). Quantum neural networks (QNNs): inherently fuzzy feedforward neural networks. *IEEE Transactions on neural networks*, 8(3):679–693.

[70] Qiu, J., Wu, Q., Ding, G., Xu, Y., and Feng, S. (2016). A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing*, 2016(1):67.

[71] Reardon, A. C. (2011). *Metallurgy for the Non-metallurgist*. ASM International, 2$^{\text{nd}}$ edition.

[72] Rosenblatt, F. (1962). *Principles of neurodynamics: Perceptrons and the theory of brain mechanisms*. Spartan Books.

[73] Ross, T. J. (2004). *Fuzzy logic with engineering applications*. Wiley, 2$^{\text{nd}}$ edition.

[74] Rubio-Solis, A. and Panoutsos, G. (2014). Fuzzy uncertainty assessment in rbf neural networks using neutrosophic sets for multiclass classification. In *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*, pages 1591–1598. IEEE.

[75] Rubio-Solis, A. and Panoutsos, G. (2015). Interval type-2 radial basis function neural network: a modeling framework. *IEEE Transactions on Fuzzy Systems*, 23(2):457–473.

[76] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533–538.

[77] Safe, M., Carballido, J., Ponzoni, I., and Brignole, N. (2004). On stopping criteria for genetic algorithms. In Bazzan, A. L. C. and Labidi, S., editors, *Advances in Artificial Intelligence – SBIA 2004*, pages 405–413, Berlin, Heidelberg. Springer Berlin Heidelberg.

[78] Schaffer, J. P., Saxena, A., Antolovich, S. D., Sanders, T. H., and Warner, S. B. (1999). *The science and design of engineering materials*. McGraw-Hill, New York, 2nd edition.

[79] Smola, A. J. and Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222.

[80] Solis, A. R. and Panoutsos, G. (2013). Granular computing neural-fuzzy modelling: A neutrosophic approach. *Applied Soft Computing*, 13(9):4010–4021.

[81] Splett, J. D., McCowan, C. N., Iyer, H. K., and Wang, C.-M. (2007). Nist recommended practice guide: Computing uncertainty for charpy impact machine test results. *NIST Special Publication*, 960:18.

[82] Tennenbaum, J. and Director, B. (1998). How gauss determined the orbit of ceres. *Fidelio Magazine*, 7(2).

[83] Tenner, J. (1999). *Optimisation of the heat treatment of steel using neural networks*. PhD thesis, University of Sheffield.

[84] The Royal Society (2017). Machine learning: the power and promise of computers that learn by example. Technical report, The Royal Society (United Kingdom).

[85] Tóth, L., Rossmanith, H.-P., and Siewert, T. (2002). Historical background and development of the charpy test. In François, D. and Pineau, A., editors, *From Charpy To Present Impact Testing*, volume 30 of *European Structural Integrity Society*, pages 3–19. Elsevier.

[86] Turksen, I. B. (1995). Knowledge representation and approximate reasoning with type II fuzzy sets. In *Proceedings of 1995 IEEE International Conference on Fuzzy Systems*, volume 4, pages 1911–1917.

[87] Vapnik, V. (1979). *Estimation of Dependences Based on Empirical Data* [in Russian]. Nauka, Moscow. (English translation: V. Vapnik, Springer Verlag, New York, 1982).

[88] Veropoulos, K., Campbell, C., and Cristianini, N. (1999). Controlling the sensitivity of support vector machines. In *Proceedings of the international joint conference on AI*, pages 55–60.

[89] Weiss, G. M. (2004). Mining with rarity: a unifying framework. *ACM Sigkdd Explorations Newsletter*, 6(1):7–19.

[90] Widrow, B. and Hoff, M. E. (1960). Adaptive switching circuits. Technical report, Stanford University, CA, Stanford Electronic Labs.

[91] World Steel (2013). Overview of the steelmaking process. [Online; Accessed August 2017] https://www.worldsteel.org/publications/bookshop/product-details~Overview-of-the-steelmaking-process~PRODUCT~Poster2011~.html.

[92] World Steel (2017). World steel in figures 2017. [Online; Accessed August 2017] https://www.worldsteel.org/en/dam/jcr:0474d208-9108-4927-ace8-4ac5445c5df8/World+Steel+in+Figures+2017.pdf.

[93] Wu, D. (2013). Two differences between interval type-2 and type-1 fuzzy logic controllers: Adaptiveness and novelty. In *Advances in Type-2 Fuzzy Sets and Systems*, pages 33–48. Springer.

[94] Xiong, S.-W., Liu, H.-B., and Niu, X.-X. (2005). Fuzzy support vector machines based on FCM clustering. In *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, volume 5, pages 2608–2613. IEEE.

[95] Yang, Y., Mahfouf, M., and Panoutsos, G. (2011a). Confidence interval assessment for charpy impact energy predictions – a gaussian mixture model (gmm)-based approach. *IFAC Proceedings Volumes*, 44(1):11738–11743.

[96] Yang, Y. Y., Mahfouf, M., and Panoutsos, G. (2011b). Development of a parsimonious GA-NN ensemble model with a case study for charpy impact energy prediction. *Advances in Engineering Software*, 42(7):435–443.

[97] Yang, Y. Y., Mahfouf, M., and Panoutsos, G. (2012). Probabilistic characterisation of model error using gaussian mixture model—with application to charpy impact energy prediction for alloy steel. *Control Engineering Practice*, 20(1):82–92.

[98] Yang, Y. Y., Mahfouf, M., Panoutsos, G., Zhang, Q., and Thornton, S. (2011c). Adaptive neural-fuzzy inference system for classification of rail quality data with bootstrapping-based over-sampling. In *Fuzzy Systems (FUZZ), 2011 IEEE International Conference on*, pages 2205–2212. IEEE.

[99] Zadeh, L. A. (1965). Fuzzy sets. *Information and control*, 8(3):338–353.

[100] 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-3:28–44.

[101] Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning – I. *Information sciences*, 8(3):199–249.

[102] Zhang, Q. and Mahfouf, M. (2011). A hierarchical mamdani-type fuzzy modelling approach with new training data selection and multi-objective optimisation mechanisms: A special application for the prediction of mechanical properties of alloy steels. *Applied soft computing*, 11(2):2419–2443.

[103] Zughrat, A., Mahfouf, M., Yang, Y., and Thornton, S. (2014). Support vector machines for class imbalance rail data classification with bootstrapping-based over-sampling and under-sampling. *IFAC Proceedings Volumes*, 47(3):8756–8761.