The University Of Sheffield.

Data-Mining and Modelling With Complex Fuzzy Sets and Fuzzy Rough Sets: Algorithms and Applications

By:

Rafael Colas-Marquez

Supervisor:

Prof. Mahdi Mahfouf

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

The University of Sheffield
Faculty of Engineering
Department of Automatic Control & Systems Engineering

September 2019

Abstract

The increasing application of machine learning models in sensitive areas, such as medicine and manufacturing, without proper knowledge of the inference process occurring within these models raises serious ethical issues. It is more important than ever to focus in the development of interpretable and transparent models based on human intuition. Fuzzy logic represent knowledge utilizing human natural language, resulting in interpretable and transparent models.

This Thesis focuses on two expansions to the traditional fuzzy set, these are complex fuzzy sets and fuzzy rough sets. Complex fuzzy sets add context to linguistic variables, resulting in compact models capable of describing the interaction between features and outputs as interferences. The developed complex fuzzy inference systems are demonstrated to be transparent and interpretable with an increase of up to 10% in prediction accuracy in comparison with state-of-the-art known fuzzy modelling approaches and up to a 300% reduction in computational time for training. Further advances are presented for the development of a complex Gaussian membership function to model uncertainties. Expanding the model to the complex domain present further advantages, including the application of complex-valued statistics for the development of a feature selection algorithm. Fuzzy rough sets are implemented for identifying inconsistencies in datasets. The models and algorithms developed in this work are applied to four real-world datasets, demonstrating the applicability in different areas. The first two datasets are material testing datasets obtained from industrial applications; the third dataset contains the information of a survival analysis performed in patients suffering from bladder cancer; the fourth dataset describes the critical temperature of superconductors.

Publications

R. Colas-Marquez and M. Mahfouf, “Data Mining and Modelling of Charpy Impact Energy for Alloy Steels Using Fuzzy Rough Sets,” IFAC-Pap., vol. 50, no. 1, pp. 14970–14975, Jul. 2017.

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

ACNFIS	Adaptive Complex Neuro Fuzzy Inferential System
AI	Artificial Intelligence
ANCFIS	Adaptive Neuro Fuzzy Complex Inference System
ANN	Artificial Neural Networks
AUC	Area under the curve
CFIS	Complex Fuzzy inference System
CFL	Complex Fuzzy Logic
CFS	Complex Fuzzy Set
CNFS	Complex Neuro Fuzzy System
COG	Center Of Gravity
DBTT	Ductile to Brittle Transition Temperature
FCM	Fuzzy C-Means
FIS	Fuzzy Inference System
GA	Genetic Algorithm
GPU	Graphic Processing Unit
IT2-Squared	Interval Type-2 Takagi Sugeno Kang Fuzzy Inference System
KNN	K-Nearest Neighbour
LM	Levenberg-Marquardt
LoR	Logistic Regression

MOIT2FM	Multi- Objective Interval Type-2 Fuzzy Modelling
PC	Partial Correlation
Q	Quadrant
RBFN	Radial Basis Function Network
RMSE	Root Mean Squared Error
ROC	Receiver Operating Characteristic
SD	Standard Deviation
SIC	Single Input Connected
SICFIS	Single Input Complex Fuzzy Inference System
SIRM	Single Input Rule Module
s-norm	triangular conorm
t-conorm	triangular conorm
t-norm	triangular norm
TSK	Takagi-Sugeno-Kang
UTS	Ultimate Tensile Strength

Chapter 1

Motivation and Thesis Overview

1.1 Motivation and Introduction

The development and the application of machine learning and Artificial Intelligence (AI) models have increased significantly in the last decade. With the application of such algorithms to high impact areas such as medical diagnosis and manufacturing it is important to develop accurate but also interpretable models based on human intuition.

The increased availability of high computing power has made feasible to develop complex machine learning models capable of surpassing human performance in certain applications, such is the case with deep Artificial Neural Networks (ANN) [1]. Many of these algorithms are being deployed in sensitive areas such as medicine [2] and finance [3]. The problem with such complex machine learning algorithms remains the inability of interpreting the inference process of black-box models. In recent years The European Union's General Data Protection Regulation included a section known as the "right to explanation", these laws may have a serious impact in the accountability of companies and industries that use machine learning and AI algorithms, potentially leading to the development of laws requiring the utilization of interpretable machine learning models or the development of tools to interpret the inference process of black-box models [4].

Fuzzy logic was developed with the intention to model human reasoning [5]. Fuzzy Inference Systems (FIS) are AI models capable of describing a system utilizing a rule-base composed of linguistic variables [6]. Compared with black-box models, FISs are

known to be transparent and interpretable given the approximation with human natural language. The transparency of a FIS assures the applicability of the model within a range of operations, while the interpretability allows the model to be validated by experts and allows to extract valuable information from a dataset to derive conclusions and make decisions [7].

The objective of this Thesis is to develop transparent, interpretable and accurate fuzzy logic models. For this work, two expansions to the fuzzy set are studied, these being, Complex Fuzzy Sets (CFS) [8] and fuzzy rough sets [9]. CFSs expand the traditional fuzzy set into the complex domain, allowing to embed concepts such as context and time. Fuzzy rough sets allow the representation of information within two approximations to model uncertainty, vagueness and inconsistencies in the data.

The models and tools developed are implemented using four different real-world datasets. The first two are industrial datasets, containing information of two common material testing, Charpy impact test, and Ultimate Tensile Strength (UTS) one. The third dataset is a medical dataset obtained from a survival study of patients suffering from bladder cancer. The fourth dataset describes the critical temperature of superconductors.

Each one of the datasets studied in this work present different challenges. Applying the tools developed on such different datasets demonstrates its generalization properties and the possibilities to expand the application of such tools to other areas.

1.2 Thesis Overview

Chapter 2 contains the literature review surveyed in this work. A brief overview of fuzzy logic and fuzzy sets is provided, followed by a review of the different types of FISs, including neuro-FIS. New advances in the expansion of the fuzzy sets are later

introduced, including the rough sets and CFS, the focus of this work. The chapter is finalized by presenting an overview of interpretability.

Chapter 3 include detailed information regarding the four datasets studied in this Thesis. The first two are material testing datasets obtained from a Charpy impact and a UTS testing. The third dataset is a survival study performed on patients suffering from bladder cancer. The fourth dataset contains information related to the critical temperature of superconductors.

Chapter 4 introduces the Single Input Complex Fuzzy Inference System (SICFIS). The SICFIS is a single feature partition per rule FIS. The concept of interference is exploited to represent the complex interaction between features and outputs. The SICIFIS model is proved to be transparent and interpretable, with a performance superior to state-of-the-art fuzzy models.

Chapter 5 improves the known Adaptive Neuro Fuzzy Inference System (ANFIS) model by substituting the linear regression consequents with SICFISs models. The ANFIS-SICFIS therefore becomes a global model composed of local interpretable SICFISs, results obtained are comparable with ensemble-ANN and evolutionary ANN models. The interpretability of the model is assessed by using a local-global performance index.

Chapter 6 introduces a complex Gaussian membership function for the development of a Mamdani-SICFIS model. The Mamdani-SICFIS is a linguistic interpretable complex FIS capable of developing models with uncertainties present in the datasets.

Chapter 7 presents the development of a filter method for feature selection based on the SICFIS model developed in Chapter 4. The results obtained are comparable with known feature selection algorithms with a considerable reduced computing time.

In Chapter 8 fuzzy rough sets are utilized for data-mining applications in the Charpy impact test dataset and the Bladder Cancer dataset. Fuzzy rough sets offers a novel tool to obtain deeper insight in the datasets and extract valuable information for developing prediction models.

Chapter 9 presents the conclusions and the future work in the field of complex FIS.

Chapter 2

State of the Art

2.1 Fuzzy Sets and Fuzzy Logic

Fuzzy sets and fuzzy logic were developed by Zadeh in [5] to model and approximate human reasoning. Fuzzy sets have a continuum grade of membership between 0 and 1, which allows the representation of vagueness and uncertainty in human natural language and in real world objects. While traditional sets classify objects with an absolute membership value of either belonging or not belonging to a class (truth or false; 1 or 0) statements such as “the oven is hot” are not intuitively represented as either completely truth or false. For example, an oven at a temperature of 160° degrees can be considered as “hot”, or even “very hot”, another oven with a temperature of 175° may considered to be between “hot” and “very hot”. Traditional logic is not capable of representing such statement as intuitively as fuzzy logic. Because of the continuum degrees of membership, it is possible to define “soft” boundaries between classes, allowing for an intuitive transition between class membership and the changes in a feature. In contrast with traditional logic which can be considered as having “hard” boundaries, small changes in a feature could mean complete change in a class membership, for example, an oven whose temperature changes from 174° to 176° would change from class membership “hot” to “very hot” instantly.

A graphical representation of the oven example from the previous paragraph is shown in Figure 2.1. The class membership assigned to each one of the values is performed by a mathematical function defined as *fuzzy membership function*, defined below:

Definition 1: Fuzzy membership function [10]:

If *X* is a collection of objects denoted generically by *x*, then a *fuzzy set A* in *U* is defined as a set of ordered pairs:

$$A = \{(x, \mu_A(x)) | x \in U\} \tag{2.1}$$

where $\mu_A(x)$ is called the membership function for the fuzzy set *A*. The membership function maps each element of *U* to a membership grade (or membership value) between 0 and 1. The set *U* is usually referred to as the universe of discourse.

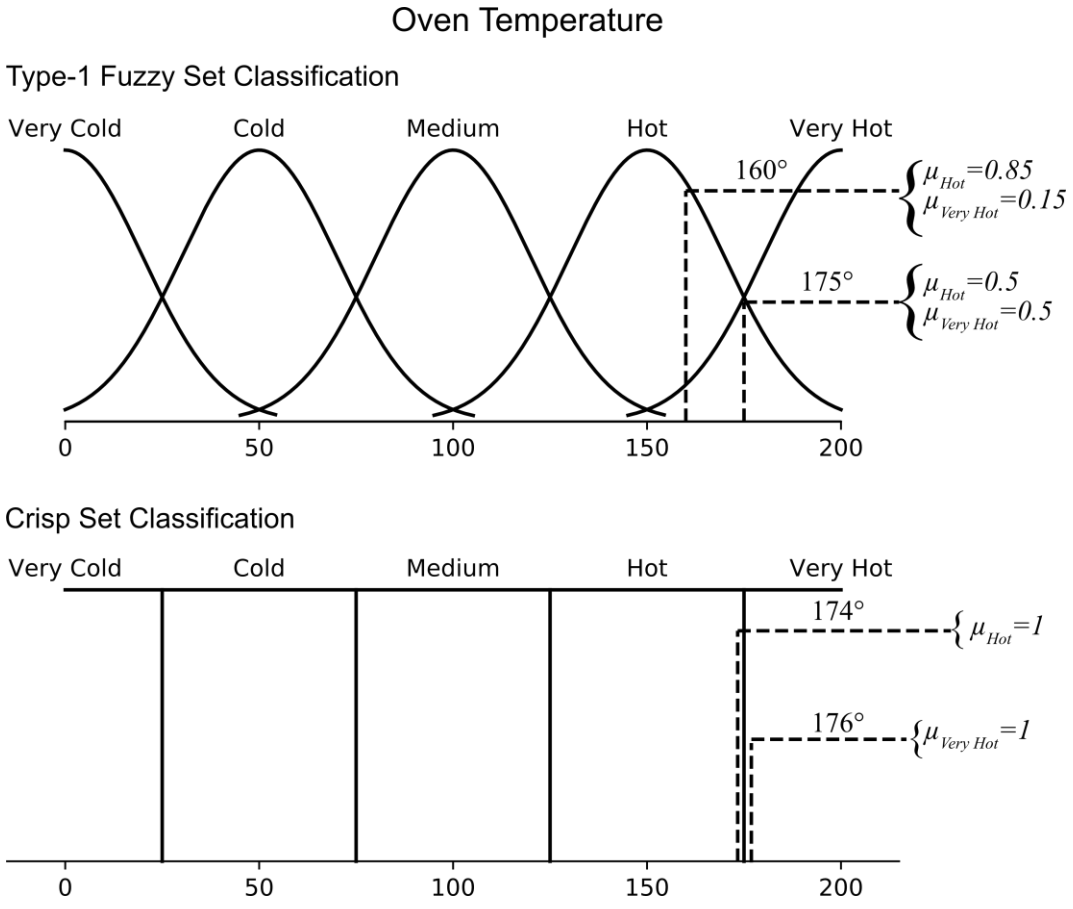


Figure 2.1: Oven temperature example to compare fuzzy sets and crisp sets.

2.1.1 Fuzzy Membership Functions

In the example shown in Figure 2.1 a Gaussian membership function (2.2) is utilized. Other examples of membership functions are the singleton membership function (2.4), and the triangular membership function (2.3) among others. A graphical representation of the Gaussian, triangular and singleton membership functions is shown in Figure 2.2.

$$\text{Gaussian membership function}: \mu_A(x,c,\sigma) = \exp\left(-\frac{1}{2}\left(\frac{\|x-c\|}{\sigma}\right)^2\right) \quad (2.2)$$

$$\text{Triangular membership function}: \mu_A(x,a,b,c) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \quad (2.3)$$

$$\text{Singleton membership function}: \mu_A(x,b) = \begin{cases} 1 \; if \; x = b \\ 0 \; if \; x \neq b \end{cases} \quad (2.4)$$

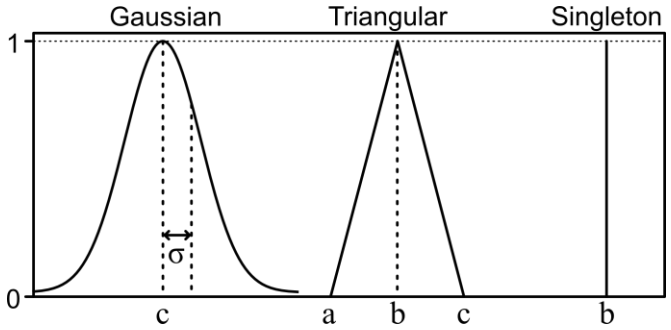


Figure 2.2: Gaussian, triangular and singleton membership functions.

2.1.2 Fuzzy Logic Operators

Just as in traditional logic and set theory, fuzzy logic utilizes logic operators to perform a diverse number of operations. Given two fuzzy sets A and B, the fuzzy intersection and union are as follows:

$$\mu_{A\cap B} = \mu_A(x) * \mu_B(x) \tag{2.5}$$

$$\mu_{A\cup B} = \mu_A(x) \oplus \mu_B(x) \tag{2.6}$$

where $*$ and $\oplus$ are known as triangular norms (t-norm) and triangular conorm (t-conorm or s-norm) operations respectively defined below:

Definition 2: T-norm [10].

A t-norm operator is a binary operation satisfying, monotonicity, commutativity and associativity axioms and whose boundaries are as follows:

$$\text{t-norm}(a,0) = 0 \tag{2.7}$$

$$\text{t-norm}(a,1) = a \tag{2.8}$$

Definition 3: T-Conorm [10].

A t-conorm operator is a binary operation satisfying, monotonicity, commutativity and associativity axioms and whose boundaries are as follows:

$$\text{t-conorm}(a,0) = a \tag{2.9}$$

$$\text{t-conorm}(a,1) = 1 \tag{2.10}$$

Some common t-norm operations are the minimum t-norm (2.11), and the product t-norm (2.12). Some common s-norm operations are the maximum s-norm (2.13) and the probabilistic sum (2.14).

$$\text{Minimum t-norm:} \quad \mu_{A\cap B} = \min(\mu_A, \mu_b) \tag{2.11}$$

$$\text{Product t-norm:} \quad \mu_{A\cap B} = \mu_A \cdot \mu_b \tag{2.12}$$

$$\text{Maximum s-norm:} \quad \mu_{A\cup B} = \max(\mu_A, \mu_b) \tag{2.13}$$

$$\text{Probabilistic sum s-norm:} \quad \mu_{A\cup B} = \mu_A + \mu_B - \mu_A \cdot \mu_B \tag{2.14}$$

2.1.3 Fuzzy Rules and Inference

Fuzzy rules are logical statements composed of linguistic variables of the following form:

$$\text{if } x \text{ is } A \text{ Then } y \text{ is } B \tag{2.15}$$

where A and B are fuzzy linguistic variables represented by fuzzy membership functions, usually referred to as the premise, and the consequence respectively. Fuzzy rules, given its proximity to human reasoning, is the main method of representing information in fuzzy logic. In the fuzzy rule (2.15) the mathematical operation where if x is A implies that y is B is called the implication and is as follows:

$$\mu_{A\rightarrow B} = \mu_A(x) * \mu_B(x) \tag{2.16}$$

where $*$ is a t-norm.

2.2 Fuzzy Inference Systems

FISs are models created to represent the behaviour of a real-world system utilizing a *rule-base* composed of the aggregation of fuzzy rules of the form (2.15) [11]. FISs are known to be universal approximators, capable of approximating any continuous function within a level of accuracy [12]. Additionally, FISs are transparent and interpretable due to its intuitive linguistic modelling. This makes them useful for modelling, representing and extracting knowledge.

The two main FISs types are Mamdani and Takagi-Sugeno-Kang (TSK). The Mamdani FIS utilize linguistic variables for both the premise and the consequences of the rule-base. TSK FISs, utilize linguistic variables for its premises but consequences are expressed utilizing a function, usually, a linear regression model [13].

2.2.1 Mamdani Fuzzy Inference Systems

The Mamdani FIS is known to be highly interpretable due to its approximation to human natural language and expressing values utilizing linguistic variables for both its premises and consequences. An example of a Mamdani FIS rule-base is shown in Table 2.1. The stages of inference in a Mamdani FIS are: fuzzification, rule firing strength, inference, rule aggregation and defuzzification. Given *R* rules and *P* features a Mamdani FIS can be represented as a 5 layered system as follows:

Table 2.1 Mamdani FIS rule-base

If	$x_1 \, is \, A_{1,1}$	And/Or	$x_1 \, is \, A_{1,1}$	And/Or	…	$x_P \, is \, A_{P,1}$	Then	$y_1 = \mu_1^Q$
If	$x_1 \, is \, A_{1,2}$	And/Or	$x_2 \, is \, A_{2,2}$	And/Or	…	$x_P \, is \, A_{P,2}$	Then	$y_2 = \mu_2^Q$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
If	$x_1 \, is \, A_{1,R}$	And/Or	$x_3 \, is \, A_{3,R}$	And/Or	…	$x_P \, is \, A_{P,R}$	Then	$y_R = \mu_R^Q$

The first layer fuzzifies a crisp input utilizing a fuzzy membership function.

$$O_{r,p}^1 = \mu_{r,p}(x_p) \tag{2.17}$$

The second layer calculates the rule firing strength of each rule according to the logic operation stated in the rule-base, for an "And" logical operator a t-norm ($*$) function is selected, in the case of the "Or" operator an s-norm ($\oplus$) function is utilized:

$$O_r^2 = w_r = \mu_{r,1} * / \oplus \mu_{r,2} \ldots * / \oplus \mu_{r,p-1} * / \oplus \mu_{r,p} \tag{2.18}$$

The third layer is the inference layer which is calculated utilizing a t-norm function.

$$O_r^3 = y_r^Q = w_r * \mu_r^Q \tag{2.19}$$

The fourth layer aggregates the output of the fourth layer utilizing an s-norm

$$O^4 = y^{\hat{Q}} = y_1^Q \oplus y_2^Q \oplus \ldots \oplus y_{R-1}^Q \oplus y_R^Q \tag{2.20}$$

For the final layer, it is necessary to defuzzified the output of the fifth layer. Several functions have been proposed. The one explored in this work is the center of gravity (COG) defuzzification which is as follows:

$$\hat{y} = \frac{\sum_{i=1}^{N} k_i \cdot y(k_i)^{\hat{Q}}}{\sum_{i=1}^{N} y(k_i)^{\hat{Q}}} \tag{2.21}$$

where k_i is variable with strictly increasing values within the specified range.

2.2.2 TSK Fuzzy Inference Systems

The TSK FIS was designed to model the dynamical behaviour of systems [14], utilizing an ensemble of local linear models. The premise of the rule-base creates a partition in the feature space, where each rule represents a local linear model of the described system. The soft boundaries between the rules allows to model a smooth transition between each of the local linear models in order to create an accurate and interpretable non-linear model [13]. An example of a rule-base system of TSK is shown in Table 2.2.

Table 2.2: TSK FIS rule-base.

If	$x_1 \, is \, A_{1,1}$	And/Or	…	$x_P \, is \, A_{P,1}$	Then	$y_1 = f^1(\mathbf{x}) = x_1 b_1^1 + x_1 b_2^1 + \ldots + x_p b_p^1 + b_0^1$
If	$x_1 \, is \, A_{1,2}$	And/Or	…	$x_P \, is \, A_{P,2}$	Then	$y_2 = f^2(\mathbf{x}) = x_1 b_1^2 + x_1 b_2^2 + \ldots + x_p b_p^2 + b_0^2$
⋮	⋮	⋮	⋮	⋮	⋮	⋮
If	$x_1 \, is \, A_{1,R}$	And/Or	…	$x_P \, is \, A_{P,R}$	Then	$y_R = f^R(\mathbf{x}) = x_1 b_1^R + x_1 b_2^R + \ldots + x_p b_p^R + b_0^R$

The overall structure of the TSK rule-base is very similar to that of the Mamdani FIS. The stages of inference in a TSK FIS are: fuzzification, rule firing strength, inference and rule aggregation. The consequences of the TSK are linear functions therefore the output of each rule is a crisp quantity and does not require de-fuzzification. The TSK FIS can be described as a 5-layered system as follows:

The first layer fuzzifies a crisp input utilizing a fuzzy membership function.

$$O_{r,p}^{1} = \mu_{r,p}(x_p) \tag{2.22}$$

The second layer calculates the rule firing strength of each rule according to the logic operation stated in the rule-base, for an "And" and "Or" logical operator a t-norm ($*$) and a s-norm ($\oplus$) functions are selected respectively.

$$O_r^2 = w_r = \mu_{r,1} * / \oplus \mu_{r,2} \ldots * / \oplus \mu_{r,p-1} * / \oplus \mu_{r,p} \tag{2.23}$$

The third layer performs a rule normalization operation.

$$O_r^3 = \bar{w}_r = \sum_{r=1}^{R} \frac{w_r}{\sum_{r=1}^{R} w_r} \tag{2.24}$$

The fourth layer performs the rule inference operation utilizing a t-norm.

$$O_r^4 = y_r = \bar{w}_r * \left(x_1 b_1^r + x_1 b_2^r + \ldots + x_P b_P^r + b_0^r \right) \tag{2.25}$$

The final layer aggregates each of the inferred rules utilizing an s-norm. Because the linear function utilized as the output of the rules in the TSK FIS, it is not necessary to perform a defuzzification operation.

$$O^5 = \hat{y} = y_1 \oplus y_2 \oplus \ldots \oplus y_{R-1} \oplus y_R \tag{2.26}$$

The lack of linguistic variable in the consequences of the rule-base cause the TSK FIS to be less interpretable than Mamdani FIS. The loss in interpretability is compensated by an increase in prediction accuracy and a reduction in computational time.

2.2.3 Single Input Fuzzy Inference Systems

The FISs rule-bases explored so far are composed of a series of statements connected with *AND-OR* operations. Single input FISs rules are composed of a single premise per rule. These systems can describe the individual effect of a feature to the output. Two common single-input FIS are the Single Input Rule Modules (SIRM's) Connected Fuzzy Inference Model [15] and the Single Input Connected (SIC) fuzzy inference method [16].

The SIRM's Connected Fuzzy Inference Model was proposed in [15] to solve the problem of combinatorial rule explosion by creating rules composed of a single premise and a single consequent. Given P features and s_p partitions per feature, the SIRMs rule-base is as follows:

Table 2.3: SIRM rule-base.

$\text{SIRM}_{1,1} = \{\text{if } x_1 = \text{A}_{1,1} \text{ then } y_{1,1} = b_{1,1}\}$
$\text{SIRM}_{1,2} = \{\text{if } x_1 = \text{A}_{1,2} \text{ then } y_{1,2} = b_{1,2}\}$
$\vdots$
$\text{SIRM}_{1,S_p} = \{\text{if } x_1 = \text{A}_{1,S_p} \text{ then } y_{1,S_p} = b_{1,S_p}\}$
$\text{SIRM}_{2,1} = \{\text{if } x_2 = \text{A}_{2,1} \text{ then } y_{2,1} = b_{2,1}\}$
$\vdots$
$\text{SIRM}_{P,S_p} = \{\text{if } x_P = \text{A}_{P,S_p} \text{ then } y_{P,S_p} = b_{P,S_p}\}$

The inference process of the SIRM is as follows. Each feature *p* is partitioned into s_p partitions, the membership function of each partition is calculated utilizing a selected membership function, from the rule-base in Table 2.3 this membership function is as follows:

$$\mu_{p,s_p} = A_{p,s_p}(x_p) \tag{2.27}$$

The inference of each feature is then calculated utilizing the normalized rule strength of the feature partitions as follows:

$$y_p = \frac{\sum_{s_p=1}^{S_p} b_{p,s_p} \cdot \mu_{p,s_p}}{\sum_{s_p=1}^{S_p} \mu_{p,s_p}} \tag{2.28}$$

The final output of the system is calculated as the weighted sum of the features inferences, the weight parameter w_p is selected to give the relative importance of each feature, the parameter can be selected initially from expert knowledge.

$$f(\mathbf{x}) = \sum_{p=1}^{P} w_p y_p \tag{2.29}$$

For the SIC fuzzy inference method utilizes the same rule-base described in Table 2.3, the main difference lies with the system output and inference process, instead of utilizing a weighted sum, it utilizes the normalized rule strength of the feature partitions and the features, the system output can be modelled as follows:

$$f(\mathbf{x}) = \frac{\sum_{p=1}^{P}\sum_{s_p=1}^{S_p} b_{p,s_p} \cdot \mu_{p,s_p}}{\sum_{p=1}^{P}\sum_{s_p=1}^{S_p} \mu_{p,s_p}} \tag{2.30}$$

The simple structure of the SIRM and the SIC fuzzy inference methods are computationally efficient given the low number of operations.

2.2.4 Fuzzy Rule-Base Elicitation

The rule-base which composes a FIS can be created utilizing different methods. The utilization of expert knowledge to derive a FIS is the earliest example of rule-base elicitation [6]. The rule-base is created based on the expert knowledge of a process. With simpler process these rule-bases can create accurate and reliable models. Nowadays is more common to develop rule-bases automatically utilizing a dataset or an information system containing the relevant information required to model a system [13]. Some of the most common methods are grid-partition and cluster base methods.

Grid partition methods are among the earlier FIS automatic rule-base elicitation methods. The features and outputs are divided into partitions creating a grid. The rule-base is composed of a combination of every feature partition and output. These number of rules grows exponentially with the addition of features and partitions, creating what is known as combinatorial rule explosion [17]. An example of a two-dimensional partition is shown in Figure 2.3. To solve the problem of combinatorial rule explosion,

different techniques have been developed; most commonly rule-bases are developed from data clusters or granules that produce more accurate and compact models.

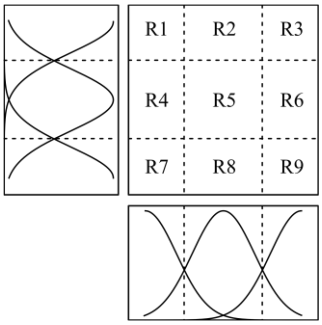


Figure 2.3: Two-dimensional grid-partition with three membership function per feature.

Cluster base methods utilize input and output data from a system or process to identify patterns, each cluster results in the formation of a rule, the size and shape of the membership function are calculated based on the geometry of each one of the clusters or granules obtained from the data, an example of cluster-based rule elicitation is shown in Figure 2.4. Two commonly used clustering algorithms are the Fuzzy C-Means (FCM) clustering algorithm [18]–[23] and the subtractive clustering algorithm [24]. Other alternatives to create initial rule-base based on the input/output information is the utilization of information granulation algorithms[25], [26] and hierarchical clustering [27].

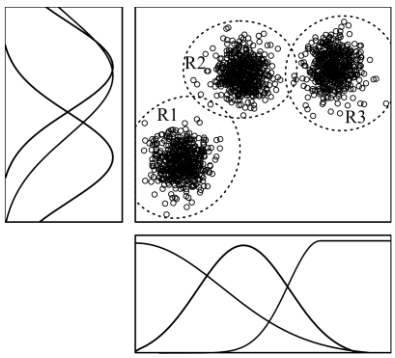


Figure 2.4: Two-dimensional cluster rule-base.

2.3 Neuro-Fuzzy Inference Systems

Eliciting a rule-base utilizing any of the methods previously described does not necessary guarantee an optimal performance of the FIS. In order to improve the performance, it is required to perform a "fine tuning" on the system parameters, such as changing the shape and position of the membership functions. A manual tuning of these rules may become intractable as the complexity increases. In order to tune automatically the parameters of a FIS it is necessary to either utilize global optimization methods such as genetic algorithms (GA) [28] or to implement learning techniques utilized in ANN defined as neuro-FIS [29].

The ANN is a black-box machine learning model known to be universal approximation [30]. On the one hand the main drawback of utilizing any type of black-box models in applications is the lack of transparency. On the other hand neuro-FIS combine the learning capabilities of ANN and the interpretability and transparency of fuzzy logic [29]. Additionally neuro-FIS are also known to be universal approximators [31].

2.3.1 Artificial Neural Networks

The ANN is a mathematical model inspired by the behaviour of neurons in the human brain. The network consists of the arrangement of artificial neurons called perceptron's in different layers to achieve a nonlinear mapping between inputs and outputs. The simplest type is the single-layer feedforward ANN. In a feedforward ANN the information flows in a single direction. In Figure 2.5 a feedforward ANN with a single hidden layer is shown. The mathematical equation is as follows:[32]

$$y(x,w)=\left(\sum_{k=1}^{K} w_k^2 \phi_k \left(\sum_{p=1}^{P} w_{kp}^1 x_p + w_0^1\right) + w_0^2\right) \tag{2.31}$$

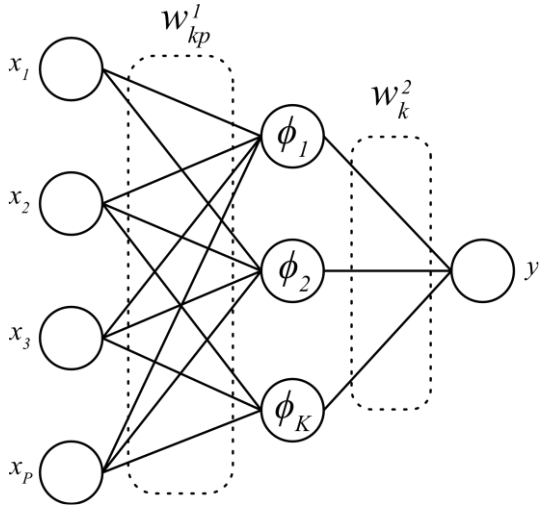


Figure 2.5: One hidden layer feedforward ANN.

where P represents the number of features, K the number of neurons in the hidden layer, ϕ represents the activation function, a common activation function is the sigmoidal function (2.32). The W parameters are called the weights of the ANN, and the w_0's are defined as the bias. These W parameters are usually calculated utilizing a gradient based optimization algorithm in order to minimize an objective function. The most common optimization method is the error-backpropagation algorithm [33].

$$\phi(a) = \frac{1}{1+e^{(-a)}} \tag{2.32}$$

2.3.1.1 The Error-Backpropagation Algorithm

The error-backpropagation algorithm is a gradient-based optimization algorithm implemented in ANN and neuro-FIS to update the parameters of a model and improve

the performance based on an objective function. The objective function utilized in the error-backpropagation is the sum of squared errors:

$$E = \frac{1}{2}\left(\hat{y} - y\right)^2 \tag{2.33}$$

where $\hat{y}$ is the estimated output of a model and y is the real output. The weights and biases of the ANN are updated according to:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \nabla E(\mathbf{w}_t)\alpha \tag{2.34}$$

where $\mathbf{w}$ is the vector containing the weights of the ANN, $\nabla E(\mathbf{w})$ is the gradient of the objective function with respect of the weights and α is the step size.

$$\nabla E(\mathbf{w}_t) = \left[\frac{\partial E}{\partial w_1} \quad \frac{\partial E}{\partial w_2} \quad \cdots \quad \frac{\partial E}{\partial w_N}\right] \tag{2.35}$$

2.3.1.2 Radial Basis Function Networks

Radial basis function networks (RBFN) are a type of ANN with a single hidden layer and the selected activation function is a Gaussian (2.36). While activation functions, such as the sigmoidal function, are supposed to activate the neuron once a threshold is met, RBFN respond to inputs located in certain regions in the feature space.

$$\phi_i = \exp\left(-\frac{1}{2}\left(\frac{\left\|x_i - c_i\right\|}{\sigma_i}\right)^2\right) \tag{2.36}$$

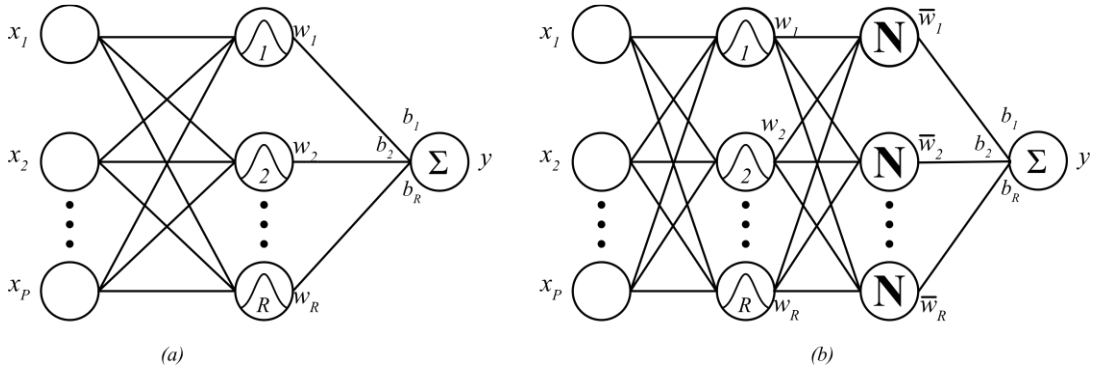


Figure 2.6: Single output RBFN a) weighted sum output and b) weighted average output.

The output of the RBFN can be either a weighted sum (2.37) (Figure 2.6 (a)) or a weighted average (2.38) (Figure 2.6 (b)). The similarities between the weighted average RBFN and the Mamdani FIS are evident, in the following section it will be demonstrated that both can be functional equivalent given certain conditions.

$$f(\mathbf{x}_i) = \sum_{r=1}^{R} b_r \phi_r(\mathbf{x}_i) = \sum_{r=1}^{R} b_r w_r \tag{2.37}$$

$$f(\mathbf{x}_i) = \sum_{r=1}^{R} \frac{b_r \phi_r(\mathbf{x}_i)}{\sum_{i=1}^{R} \phi_r(\mathbf{x}_i)} = \sum_{i=1}^{R} \frac{b_r w_r}{\sum_{i=1}^{R} w_r} \tag{2.38}$$

2.3.2 Neuro Fuzzy Mamdani Fuzzy Inference System

The Mamdani FIS can be functionally equivalent to RBFN under certain conditions [34][35]. The first condition is the selection of a fuzzy Gaussian membership function for the premises. The second condition is to select the algebraic product as the t-norm operation for the calculation the rule firing strength and the implication. The third condition is to aggregate the rules utilizing an algebraic sum operation. Finally by

selecting a singleton membership function (2.4) for the consequents of the rules and selecting the COG defuzzification method results in a function equivalent to the weighted average of the RBFN activation functions outputs (2.38). It is important to note that the algebraic-sum is not an s-norm, such modification result in greater computationally efficiency [10] and in a functional equivalence to the RBFN.

The Mamdani FIS with singleton defuzzification can be described as a four layered system. The first layer fuzzifies the input (2.39), the second layer calculates the rule firing strength (2.40), the fourth layer calculates the inference (2.41). The final layer defuzzified the input utilizing the COG method (2.42).

$$O_{r,p}^{1} = \mu_{r,p} = \exp\left(-\frac{1}{2}\left(\frac{x_{r,p} - c_{r,p}}{\sigma_{r,p}} \right)^{2} \right) \tag{2.39}$$

$$O_{r}^{2} = w_{r} = \prod_{p=1}^{P} \mu_{r,p} \tag{2.40}$$

$$O_{r}^{3} = w_{r} \cdot b_{r} \tag{2.41}$$

$$O_{r}^{4} = \sum_{r=1}^{R} \frac{b_{r} \cdot w_{r}}{\sum_{r=1}^{R} w_{r}} \tag{2.42}$$

The backpropagation algorithm can be utilized for adjusting the singleton membership function position b_r, the spread $\sigma_{r,p}$,and centre $c_{r,p}$ of the Gaussian membership function. The partial derivatives of the objective function (2.33) with respect to the $b, \sigma \text{and} c$ parameters are as follows:

$$\frac{\partial E}{\partial b^{r}} = \left(\hat{y} - y \right) \frac{w_{r}(\mathbf{x})}{\sum_{r=1}^{R} w_{r}(\mathbf{x})} \tag{2.43}$$

$$\frac{\partial E}{\partial \sigma_{r,p}} = \left(\hat{y} - y\right) \frac{b_r - \hat{y}}{\sum_{r=1}^{R} w_r(\mathbf{x})} w_r(x) \left(\frac{\left(x_p - c_{r,p}\right)^2}{\left(\sigma_{r,p}\right)^3} \right) \tag{2.44}$$

$$\frac{\partial E}{\partial c_{r,p}} = \left(\hat{y} - y\right) \frac{b_r - \hat{y}}{\sum_{r=1}^{R} w_r(\mathbf{x})} w_r(x) \left(\frac{x_p - c_{r,p}}{\left(\sigma_{r,p}\right)^2} \right) \tag{2.45}$$

2.3.3 The Adaptive Network Based Fuzzy Inference System

The ANFIS model is based on the TSK FIS [10]. Rules are composed of premises whose membership function usually is selected to be a Gaussian, it utilizes the product t-norm for the conjunction and implication operations and utilizes the algebraic sum for aggregating rules.

The ANFIS model can be described as a five-layered system as shown in Figure 2.7. The first layer fuzzifies the input (2.46), the second layer calculates the firing strength (2.47). The third layer performs a rule normalization operation (2.48). The fourth layer calculates the inference (2.49). The fifth layer aggregates the rules with the algebraic sum operation (2.50).

$$O_{r,p}^1 = \mu_{r,p} = \exp\left(-\frac{1}{2} \left(\frac{x_{r,p} - c_{r,p}}{\sigma_{r,p}} \right)^2 \right) \tag{2.46}$$

$$O_r^2 = w_r = \prod_{p=1}^{P} \mu_{r,p} \tag{2.47}$$

$$O_r^3 = \bar{w}_r = \frac{w_r}{\sum_{r=1}^{R} w_r} \tag{2.48}$$

$$O_r^4 = \bar{w}_r \left(b_{r,0} + b_{r,1} x_1 + \ldots b_{r,P} x_P \right) = \bar{w}_r g_r(\mathbf{x}) \tag{2.49}$$

$$O^5 = \sum_{r=1}^{R} \bar{w}_r g_r(\mathbf{x}) \tag{2.50}$$

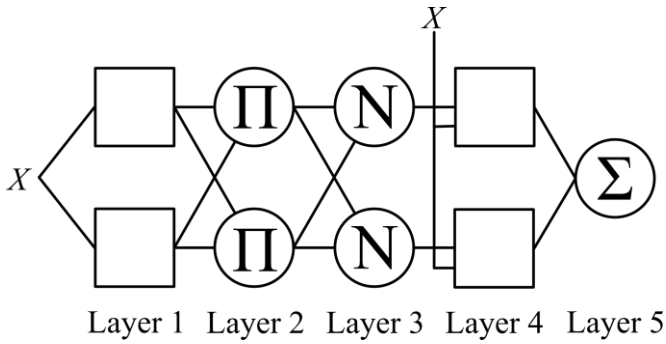


Figure 2.7: ANFIS schematic.

What differentiates the ANFIS from the TSK model is the application of a hybrid learning method. The premises parameters (σ, c) are optimized utilizing the backpropagation algorithm while the consequence parameters $(b_{1,0} \cdots b_{R,P})$ are optimized utilizing a linear least-squares error method. The output of the fifth layer is a weighted sum of the output of a linear regression model (2.50). Therefore, by treating the normalized fired rule strength $\bar{w}_r$ as a constant, it is possible to perform a linear least squares optimization as follows:

$$\underbrace{\begin{bmatrix} \bar{w}_1 & \bar{w}_1 x_1^1 & \cdots & \bar{w}_1 x_P^1 & \bar{w}_2 & \bar{w}_2 x_1^1 & \cdots & \bar{w}_R x_P^1 \\ \bar{w}_1 & \bar{w}_1 x_1^2 & \cdots & \bar{w}_1 x_P^2 & \bar{w}_2 & \bar{w}_2 x_1^2 & \cdots & \bar{w}_R x_P^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{w}_1 & \bar{w}_1 x_1^1 & \cdots & \bar{w}_1 x_P^N & \bar{w}_2 & \bar{w}_2 x_1^N & \cdots & \bar{w}_R x_P^N \end{bmatrix}}_{\Theta} \underbrace{\begin{bmatrix} b_{1,0} \\ b_{1,1} \\ \vdots \\ b_{1,P} \\ b_{2,0} \\ b_{2,1} \\ \vdots \\ b_{R,P} \end{bmatrix}}_{\mathbf{b}} \tag{2.51}$$

$$\mathbf{b}^* = \left(\Theta^T \Theta\right)^{-1} \Theta^T y \tag{2.52}$$

where Θ is called the design matrix, and *N* represents the number of instances in the dataset. The hybrid optimization algorithm alternates the training at each step of the premise parameters $\boldsymbol{\sigma}$ and $\mathbf{c}$,and the consequent parameters $\mathbf{b}$.

2.4 Type-2 Fuzzy Sets

Type-2 fuzzy sets where originally proposed by Zadeh in [36]. In a type-2 membership function, each value of its membership is a type-1 fuzzy set, as shown in Figure 2.8, this increases the ability of a fuzzy membership function to model uncertainties.

The development of a type-2 fuzzy inference system represents several challenges, mainly due to the computational requirements for modelling and performing operations in type-2 fuzzy membership functions [37]. To overcome these limitations, interval type-2 fuzzy sets and membership function were developed [37]. An interval type-2 fuzzy sets is composed of an upper and a lower type-1 fuzzy membership function, representing the region between the membership functions as the footprint of uncertainty as shown in Figure 2.8 a and b. This allows to model uncertainties while reducing the computational requirements of type-2 fuzzy inference systems.

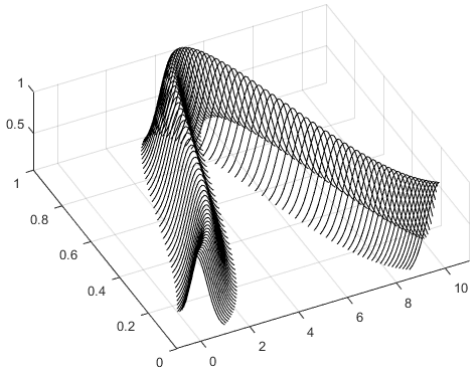
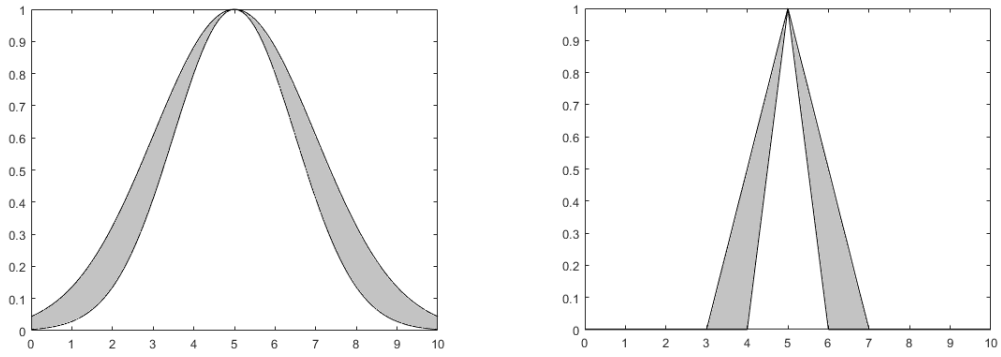


Figure 2.8: Type-2 Gaussian membership function



(a) (b)

Figure 2.9: (a) Interval type-2 Gaussian membership function. (b) Interval type-2 Gaussian membership function and

Type-2 and interval type-2 fuzzy inference systems have been applied to a wide range of fields, including control [38], healthcare [39], and metallurgy [40].

2.5 Rough Sets

Rough sets were developed by Pawlak in [41] to model vagueness and uncertainty. A rough set is composed of two approximations: a lower approximation that contains all the objects that certainly belong to a class and an upper approximation that contains all the objects that may or may not belong to a class. An example of an information table is shown in Table 2.4.

Table 2.4: Information table example.

Object	Feature 1	Feature 2	Feature 3	Output
1	A	C	B	1
2	A	C	B	0
3	B	A	C	0
4	B	A	A	1
5	A	A	C	0
6	B	A	A	1

In an information system given any subset *D* of *P* conditional features indiscernibility is assessed as follows [41]:

$$IND(D) = \{(x, y) \in U^2 \mid \forall d \in D, d(x) = d(y)\} \tag{2.53}$$

Therefore, two objects are indiscernible if they contain the same feature values, for the features in *D*. For example in the information system shown in Table 2.4 the indiscernible objects of the following subsets $D_1 = \{Feature1, Feature2, Feature3\}$, $D_2 = \{Feature1, Feature2\}$ and $D_3 = \{Feature3\}$ are as follows:

$$IND(D_1) = \{\{1,2\},\{3,5\},\{4,6\}\} \tag{2.54}$$

$$IND(D_2) = \{\{1,2\},\{3,4,6\},\{5\}\} \tag{2.55}$$

$$IND(D_3) = \{\{1,2\},\{3,5\},\{4,6\}\} \tag{2.56}$$

Indiscernible objects are treated as a single information granule and represented by a set $[x]_p$. The lower and upper approximation are respectively:

$$\underline{PX} = \{x \mid [x]_p \subseteq X\} \tag{2.57}$$

$$\overline{PX} = \{x \mid [x]_p \cap X \neq \varnothing\} \tag{2.58}$$

The tuple $\langle \underline{PX}, \overline{PX} \rangle$ is defined as the Rough set. A graphical representation of a rough sets is shown in Figure 2.10.

The positive, negative and boundary regions of a rough set given two sets of attributes P and Q are as follows:

$$POS_P(Q) = \bigcup_{X \in U/Q} \underline{P}X \tag{2.59}$$

$$NEG_P(Q) = U - \bigcup_{X \in U/Q} \bar{P}X \tag{2.60}$$

$$BND_P(Q) = \bigcup_{X \in U/Q} \bar{P}X - \bigcup_{X \in U/Q} \underline{P}X \tag{2.61}$$

The positive region contains all the objects of U that can be classified to a class U/Q given the information contained in the attributes P. The boundary region contains the set of objects that can't be classified with absolute certainty, and the negative region contains the objects that certainly cannot be classified. In the example shown in Table 2.4, the positive regions of D_1, D_2 and D_3 given $Q = Output$ are as follows:

$$\underset{1}{POS_{D_1}}(Q) = \{\underset{0}{\{4,6\}}, \{3,5\}\} \tag{2.62}$$

$$\underset{1}{POS_{D_2}}(Q) = \{\underset{0}{\{\varnothing\}}, \{5\}\} \tag{2.63}$$

$$\underset{1}{POS_{D_3}}(Q) = \{\underset{0}{\{4,6\}}, \{3,5\}\} \tag{2.64}$$

From the Positive region of D_1 instances {1,2} do not form part of any class, the reason for this is the conflict in the output Q, it is not possible to determine whether a such feature values would determine a precise output, therefore instances {1,2} are considered inconsistent. From D_2 it is seen a decrease in the size of the sets. That is because the removal of features, especially Feature 3, makes it impossible to discern between objects and to classify the output appropriately. Additionally, it is seen from the results, that D_3 contains the same number of objects in its positive region as D_1. The feature dependency can be measured as follows:

$$\gamma_D(Q) = \frac{|POS_D(Q)|}{|U|} \tag{2.65}$$

The feature dependency is a measure of how well a set of features can describe the output. For the subsets D_1, D_2 and D_3 from the example of Table 2.4 the feature dependency is the following:

$$\gamma_{D_1}(Q) = \frac{|\{3,4,5,6\}|}{|\{1,2,3,4,5,6\}|} = 0.6666 \tag{2.66}$$

$$\gamma_{D_2}(Q) = \frac{|\{5\}|}{|\{1,2,3,4,5,6\}|} = 0.1666 \tag{2.67}$$

$$\gamma_{D_3}(Q) = \frac{|\{3,4,5,6\}|}{|\{1,2,3,4,5,6\}|} = 0.6666 \tag{2.68}$$

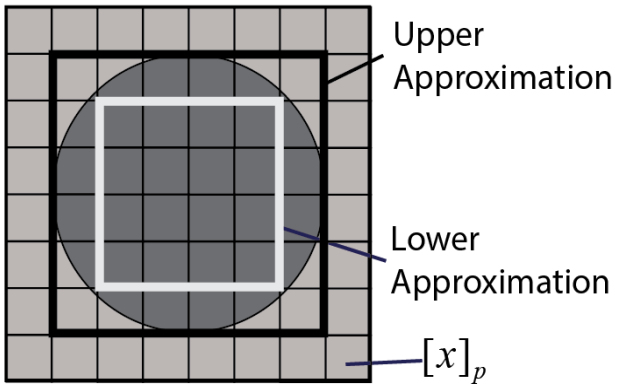


Figure 2.10: Rough set representation.

Rough sets were applied for a diverse number of applications such as knowledge discovery [42] and clustering [43]. Where rough sets have been most successfully applied has been in the development of feature selection algorithms [44]–[47]. Rough sets suffer from the limitation of being only applicable to qualitative datasets, thus limiting its applicability considerably given that most real-world datasets are composed of mixed valued data. To solve this problem, the development of fuzzy-rough sets

hybrids were developed [9]. Fuzzy rough sets are capable of modelling mixed datasets given the continuous degree of membership of fuzzy sets.

2.5.1 Fuzzy Rough Set Theory

The fuzzy rough sets hybrids were initially proposed by Dubois and Prade in [9], the method consists of developing fuzzy partitions in the dataset. The fuzzy rough lower and upper approximations are estimated as follows:

$$\mu_{\underline{P}X}(F_i) = \inf_x \max\left\{1-\mu_{F_i},(x),\mu_X(x)\right\} \tag{2.69}$$

$$\mu_{\overline{P}X}(F_i) = \sup_x \max\left\{\mu_{F_i},(x),\mu_X(x)\right\} \tag{2.70}$$

where F_i is a fuzzy equivalence class and $\mu_X(x)$ denotes the degree to which *x* belongs to fuzzy equivalence class X [30]. The main drawback with Dubois and Prade's fuzzy rough sets is the exponential increase in computations required with the addition of features and fuzzy partitions. An alternative fuzzy-rough set elicitation method was introduced by Radzikowska and Kerre in [48]. Instead of measuring the indiscernibility relationship between objects a measure of their similarity is calculated using a fuzzy tolerance relationship, μ_{R_p}. The fuzzy-rough lower and upper approximations are as follows:

$$\mu_{\underline{R_p}X}(x) = \inf_{y\in U} I(\mu_{R_p}(x,y),\mu_X(\mathrm{y})) \tag{2.71}$$

$$\mu_{\overline{R_p}X}(x) = \sup_{y\in U} T(\mu_{R_p}(x,y),\mu_X(y)) \tag{2.72}$$

$$\mu_{R_p}(x,y) = \bigcap_{p\in P}\{\mu_{R_p}(\mathrm{x},\mathrm{y})\} \tag{2.73}$$

where *T* is a t-norm, and *I* is a fuzzy implicator. μ_{Rp} is a similarity measure between objects *x* and *y* for a feature *p*. Jensen and Chen [49] proposed the application of the

Łukasiewicz t-norm (2.74) , the Łukasiewicz implicator (2.75) and proposed the following fuzzy similarity relations (2.76)-(2.78).

$$T = \left(\max\left(x + y - 1, 0\right)\right) \tag{2.74}$$

$$I = \left(\min\left(1 - x + y, 1\right)\right) \tag{2.75}$$

$$\mu_{R_p}(x, y) = \exp\left(-\frac{(p(x) - p(y))^2}{2\sigma_p^2}\right) \tag{2.76}$$

$$\mu_{R_p}(x, y) = 1 - \frac{\left|p(x) - p(y)\right|}{\left|p_{\max} - p_{\min}\right|} \tag{2.77}$$

$$\mu_{R_p}(x, y) = \max\left(\min\left(\frac{p(y) - \left(p(x) - \sigma_p\right)}{p(x) - \left(p(x) - \sigma_p\right)}, \frac{\left(p(x) + \sigma_p\right) + p(y)}{\left(p(x) + \sigma_p\right) + p(x)}\right)\right) \tag{2.78}$$

where σ_p^2 is the variance of feature *p*.

The positive region and feature dependency of a fuzzy-rough sets are calculated as follows:

$$\mu_{POS_{R_p}(Q)}(X) = \sup_{X \in U/Q} \mu_{\underline{R_p X}}(x) \tag{2.79}$$

$$\gamma_p^{\mu}(Q) = \frac{\sum_{x \in U} \mu_{POS_{RP}(Q)}(x)}{\left|U\right|} \tag{2.80}$$

Rough set theory and fuzzy rough set theory have been implemented successfully in different fields such as in pattern recognition [45], attribute selection [44], [45], [47], [49]–[52], rule induction [53], classification [47], [54] and knowledge discovery [42], [47].

2.6 Complex Fuzzy Sets and Logic

CFS theory was first developed by Ramot et al. [8] [55]. A CFS *S* in a universe of discourse *U* is defined as follows:

$$\mu_S(x) = r_S e^{j\omega_S(x)} \tag{2.81}$$

where $j = \sqrt{-1}$, r_S and ω_S are the *magnitude* and the *phase* of the CFS respectively. While traditional type-1 fuzzy sets lie within the interval $[0,1]$ the CFS lies within a unit circle. The magnitude r_S represents a type-1 fuzzy sets and the phase ω_S is a relative quantity that assigns the "context". This makes the type-1 fuzzy set a special case of the CFS when all phases are equal to zero.

2.6.1 Complex Fuzzy Operations

According to [8] [55], the magnitude and the phase of the CFS are two separate identities, and therefore the operations applied to one should not affect the other. In the case for the complex fuzzy union and intersection, given two complex membership functions *A* and *B*, the resultant membership function of the union operation $A \cup B$ and intersection operation $A \cap B$ is given as follows:

$$\mu_{A\cup B}(x) = \left[r_A(x) \oplus r_B(x)\right] \cdot e^{j\omega_{A\cup B}(x)} \tag{2.82}$$

$$\mu_{A\cap B}(x) = \left[r_A(x) * r_B(x)\right] \cdot e^{j\omega_{A\cap B}(x)} \tag{2.83}$$

where $\oplus$ represents any *t-conorm* function and $*$ represents any *t-norm* function. The following equations (2.84)-(2.90) are proposed for both the union and intersection of the phase [8], [55]:

$$\omega_{A\cup B} = \omega_A + \omega_B \tag{2.84}$$

$$\omega_{A\cup B} = \max\left(\omega_A, \omega_B\right) \tag{2.85}$$

$$\omega_{A\cup B} = \min\left(\omega_A, \omega_B\right) \tag{2.86}$$

$$\omega_{A\cup B} = \omega_A - \omega_B \tag{2.87}$$

$$\omega_{A\cup B} = \begin{cases} \omega_A & r_A > r_B \\ \omega_B & r_B < r_A \end{cases} \tag{2.88}$$

$$\omega_{A\cup B} = \frac{r_A \cdot \omega_A + r_B \cdot \omega_B}{r_A + r_B} \tag{2.89}$$

$$\omega_{A\cup B} = \frac{\omega_A + \omega_B}{2} \tag{2.90}$$

The characteristic operator of the CFS is the complex fuzzy aggregator which is also called vector aggregation [8] [55]. CFSs are composed of a magnitude and a phase, therefore CFSs exhibit “wave-like” properties, when two or more CFS are aggregated the magnitude of the resultant vector will depend on the phase alignment of the CFSs. The definition of the complex fuzzy aggregation [55] is as follows:

Definition 4 [55]: Let $A_1, A_2, ..., A_n$ be CFS defined on the universe of discourse U. vector aggregation on $A_1, A_2, ..., A_n$ is defined by a function v as follows:

$$v : \left\{a \middle| a \in \mathbb{C}, |a| \le 1\right\}^n \to \left\{b \middle| b \in \mathbb{C}, |b| \le 1\right\} \tag{2.91}$$

The function *v* produces an aggregate fuzzy set A by operating on the membership grades of $A_1, A_2, ..., A_n$ for each $x \in U$. For all $x \in U$, v is given by:

$$\mu_A(x) = v\left(\mu_{A_1}(x), \mu_{A_2}(x), ..., \mu_{A_n}(x)\right) = \sum_{i=1}^{n} w_i \mu_{A_i} \tag{2.92}$$

With $w_i \in \{a \mid a \in \mathbb{C}, |a| \leq 1\}$ for all *i*, and $\sum_{i=1}^{n} |w_i| = 1$.

The definition of the vector aggregator operation is intended to be as general as possible and the calculation of the complex weights w_i are problem-dependent [55].

For the implication operator, the proposed function is the algebraic product (2.93).

$$\mu_{A \to B}(x, y) = \mu_A(x) \cdot \mu_B(y) \tag{2.93}$$

where:

$$r_{A \to B}(x, y) = r_A(x) \cdot r_B(y) \tag{2.94}$$

$$\omega_{A \to B}(x, y) = \omega_A(x) + \omega_B(y) \tag{2.95}$$

2.6.2 Complex Fuzzy Sets With and Without Rotational Invariance

The magnitude and the phase of the CFS proposed in [8] and [55] have separate identities, and the operations performed on one should not have an effect on the other. Dick [56] defines this CFS as one "with rotational invariance". A rotational invariant CFS has several limitations, and most importantly, Dick demonstrates that "the algebraic product cannot be used as a conjunction operation" [57]. in a rotational invariant CFS, even though Ramot et al. utilizes the product function as implication [56], [57]. To resolve these limitations Dick proposes a CFS "without rotational invariance" based on vector logic, where the magnitude and phase are not separate identities. Dick proves that in a CFS without rotational invariance the algebraic product can be used as a conjunction operation [56], [57].

2.6.3 Pure Complex Fuzzy Sets

Tamir [58] expands the original idea of CFS devised by Ramot et al. [8] [55] and proposes a "pure CFS". The rotational invariant CFS only conveys the fuzzy information in the magnitude, in a pure CFS both the magnitude and the phase convey the fuzzy information; the pure CFS can be alternatively represented in rectangular form. In a pure CFS either the real or the imaginary part (alternatively the magnitude and the phase) represents a fuzzy set, while the other represents a fuzzy class. Fuzzy classes [59] are sets of fuzzy sets, therefore a pure CFS represents the membership of an object in a fuzzy class and a fuzzy set.

2.6.3.1 Other Complex Fuzzy Sets

The field of CFS and logic is relatively new, and more research and applications are being developed. With that a whole new development of different CFS, including those based in Atanassov intuitionistic fuzzy sets [60], which include the Pythagorean fuzzy sets [61], and the complex intuitionistic fuzzy sets [62]. Complex neutrosophic sets have also been proposed [63].

In [64] the authors make a comparison between the CFS and a type-2 fuzzy sets, among their conclusions, it is of importance to denote the following:

1) The CFS conveys an extra dimension of information while a type-2 fuzzy set is used to represent uncertainty.

2) In 3 dimensions a type-2 fuzzy sets represent a surface while the CFS represents a trajectory.

Additional work on type-2 and interval valued complex fuzzy sets can also be found in [65]–[67]. A comprehensive review of the state of the art of CFS can be found in [57]

2.6.4 Complex Fuzzy Inference Systems

Complex fuzzy inference systems (CFISs) are a set of FIS based on the CFS with rotational invariance proposed by Ramot et al. in [8] and the CFS without rotational invariance proposed by Dick in [56]. These CFISs are not to be confused with complex valued fuzzy inference systems which are not based on CFS but are based on either complex fuzzy numbers or the application of complex valued information in the FIS [68]–[72]. The CFISs developed so far are the Adaptive Neuro Fuzzy Complex Inference System (ANCFIS) [73], the Complex Neuro Fuzzy System (CNFS) [74], and the Adaptive Complex Neuro Fuzzy Inferential System (ACNFIS) [75].

2.6.4.1 The Adaptive Neuro Fuzzy Complex Inference System

The first CFIS developed was the ANCFIS [73]. The ANCFIS is a six-layered system (Figure 2.11) based on the ANFIS architecture [76] designed specifically to model time series data utilizing CFSs without rotational invariance [56]. Compared with most FIS the ANCFIS utilizes a sinusoidal membership function; It is known from the Fourier theorem that any periodic function can be approximated with a series of sums of sines and cosines, therefore it is proposed in [73] a sinusoidal membership function to capture the most important frequencies and model the approximate periodic behaviour of an input window.

The sinusoidal membership function is as follows:

$$r_s(\theta) = d\sin(a\theta + b) + c \tag{2.96}$$

where, r and θ are the magnitude and phase of the CFS respectively, the parameters a, b, c and d modify the frequency, phase shift, vertical shift and amplitude respectively.

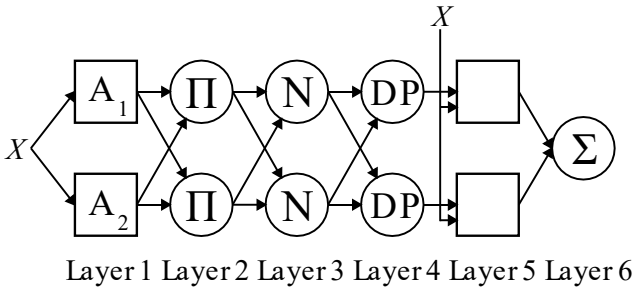


Figure 2.11: ANCFIS schematic.

The first layer of the ANCFIS convolves an input window time series dataset. The second layer calculates the firing strength of the rules utilizing the algebraic product. The third layer normalizes the firing strength, during normalization only the magnitudes of the CFSs are normalized and the phases are left unchanged. The fourth layer is an additional layer not present in the ANFIS model, called the rule interference layer, instead of utilizing the vector aggregation proposed by Ramot et al., the interferences are created by applying a dot product between the rules; the output of the fourth layer is a real valued scalar. The fifth layer calculates the consequent parameters and multiplies the output of the fourth layer. The sixth layer is the output layer where the scalar output of each rule are summed.

The ANCFIS model utilizes an input window instead of delayed inputs; this reduces the number of rules to the number of input windows creating a compact FIS. The parameters are optimized utilizing a hybrid optimization algorithm, for the forward pass a least squares algorithm is used to update the consequences, the backward pass utilizes a combination of complex back propagation [77] and derivative free optimization to update the premise parameters.

Variations on the ANCFIS input type, architecture and operations have been explored throughout its development and the author encourages the reader to research the work done for the ANCFIS model.

The ANCFIS model has been applied to different datasets: The Wolfer sunspot numbers [73], [78], [79], the Mackey-Glass 17 [73], [78], the Santa Fe laser dataset [73], [78], stellar brightness [78], wave heights [78], [79], Photovoltaic power dataset [80]. The ANCFIS has also been implemented successfully in modelling multivariate time series, such as a Motel monthly occupancy [81] [82], Flour monthly price[81], [82] Monthly precipitation in different areas in Tennessee [81], [82], and NASDAQ [82]. A variation on the training algorithm to incorporate extreme learning machines was applied to four different software reliability growth datasets [83] . The reported results obtained from the ANCFIS are comparable with other models while maintaining a compact model, utilizing in some circumstances fewer than 3 rules to model complex datasets and chaotic time series.

2.6.4.2 The Complex Neuro-Fuzzy System

The CNFS is based on the ANFIS architecture [76] and CFSs with rotational invariance [8], the system utilizes a complex Gaussian membership function. A hybrid learning algorithm is applied for the training which consists of a least squares algorithm for consequences and a derivative free optimization algorithm for the premises. The model output is a complex number with a real and imaginary part, defined as the dual output property. The real part is generally used as the final output of the system, with the dual output property is explored in [84] and [85].

Two different types of complex Gaussian membership function are utilized. Initially in [86]–[88] the membership function used is the Gaussian membership function represented in rectangular form:

$$cGauss = \exp\left(\frac{-(h-m)^2}{2\sigma^2}\right) + j\frac{-(h-m)}{2\sigma^2}\exp\left(\frac{-(h-m)^2}{2\sigma^2}\right) \qquad (2.97)$$

In subsequent papers [84], [85], [89], the complex Gaussian membership function is modified to add a term λ called the frequency factor which multiplies the phase of the membership function, and the polar representation is utilized:

$$cGaussian(x) = r_S e^{j\omega_S(x)} \qquad (2.98)$$

$$r_s(x,c,\sigma) = \exp\left(-0.5\left(\frac{x-c}{\sigma}\right)^2\right) \qquad (2.99)$$

$$\omega_s(x,c,\sigma,\lambda) = -\exp\left(-0.5\left(\frac{h-m}{\sigma}\right)^2\right)\left(\frac{h-m}{\sigma^2}\right)\lambda \qquad (2.100)$$

The first layer of the CNFS calculates the value of the complex membership utilizing either (2.97) or (2.98). The second layer calculates the firing strengths according to (2.83) utilizing the product operation of the magnitudes and the addition operation for the phases (2.84). The third layer normalizes the whole complex number. The fourth layer calculates the linear consequences and multiplies the normalized weights from the third layer. The fifth layer calculates the output by summing the signals of the network, the real part is used as the final output. The imaginary part can also be used as an output in certain circumstances.

CNFSs have been applied for function approximation [74], noise cancelling [86], time series prediction [87], [89], knowledge discovery [88]. The dual output property is explored in [84] for financial purposes to calculate both the opening and closings of

the NASDAQ and in another instance to calculate simultaneously the TAIEX index and the Dow Jones with the real and the imaginary part of the complex output.

2.6.4.3 The Adaptive Complex Neuro–Fuzzy Inferential System

The ACNFIS [75] is a 5 layer FIS (Figure 2.12) based on the ANFIS model [76] and utilized a CFS with rotational invariance [8]. The ACNFIS utilizes two Gaussian functions as the magnitude and phase membership function, because "a complex valued function cannot be both analytical and bounded unless is a constant" [75], the complex membership function utilizes two real valued functions to bound the complex membership within the unit circle. The complex membership function is as follows:

$$\mu(x) = \exp\left(-\left(\frac{x-c_{A_j}}{a_{A_j}}\right)^2\right) \angle \left(2\pi \exp\left(-\left(\frac{x-c_{P_j}}{a_{P_j}}\right)^2\right)\right) \qquad (2.101)$$

where A is the magnitude and P is the phase.

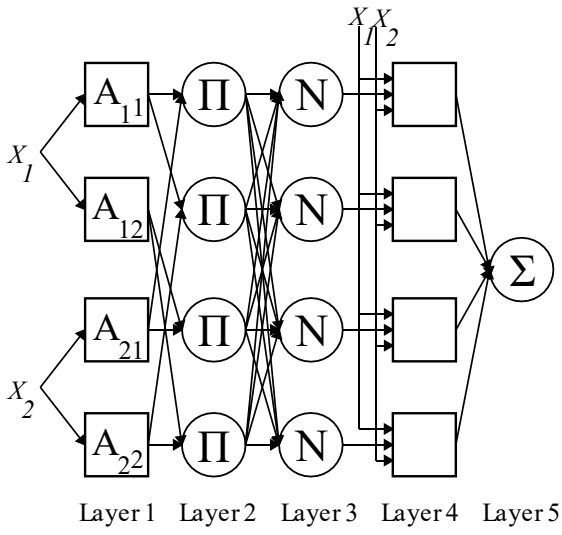


Figure 2.12: ACNFIS schematic.

The first layer of the system calculates the complex membership function according to (2.101). The second layer calculates the firing strengths according to (2.83) utilizing the product operation of the magnitudes and the addition operation for the phases (2.84) , utilizing the product operation of the magnitudes and the addition operation for the phases. The third layer normalizes the magnitude of the complex number. The fourth layer calculates the linear consequences, in the ACNFIS two linear consequences per rule are calculated, one for the real part and one for the imaginary part. The real part is utilized as the final output. The system utilizes a Levenberg-Marquardt (LM) optimization algorithm for training.

2.7 Interpretability and Transparency

Interpretability and transparency are subjective properties of models which definition varies from different sources. According to Mencar and Fanelli [90] Transparency is a property of a system to represent the relationship between features and output variables, interpretability is a subjective property related to the representation and transmission of knowledge trough symbols and characters (e.g. linguistic variables and rules of a fuzzy system) [90].

For Lipton [91] the interpretability of a model can be composed of two main properties, *transparency* and *post-hoc explanations*. Transparency is the property of a model to explain how a model works, by its entirety and its individual components. Post-hoc explanations relate to the representation of information to extract knowledge about a process.

Other authors consider interpretability and transparency not to be closely related properties. In [92] the author considers all FIS to be interpretable given its proximity to natural language, while transparency is a property that measures the reliability and robustness of a model.

2.7.1 Interpretability and Transparency in Fuzzy Inference Systems

The main advantage of utilizing FIS over others is the interpretability and transparency that fuzzy logic provides. Good performance and generalization properties have been shown, with the additional advantage already explained in previous section of soft boundaries. There exists no clear mathematical definition of interpretability and transparency, regardless a few guidelines [93], and measurements [7], [93]–[96] can be taken into consideration to better develop interpretable and transparent FISs.

In [93], the authors develop a taxonomy to classify the proposed interpretability measures and techniques to improve interpretability in linguistic (Mamdani) rule-base FIS (Table 2.5). These sets of measures and techniques are grouped into a double helix "complexity versus sematic interpretability" and "rule-base versus fuzzy partition". The four quadrants (Q) are Q1 Complexity at the rule-base level, Q2 Complexity at the level of fuzzy partitions, Q3 Semantics at the rule-base level, Q4 Semantics at the fuzzy partition level.

The first quadrant relates to the number of rules and the number of conditions per rule. Maintaining a parsimonious model is essential to be interpretable. It is known in psychology that humans struggle processing more than seven information objects. In [97] the number of information objects that a human can process was found to be 7 ± 2. Therefore, it is important to maintain rule-base systems with no more than 9 premises per rule [93].

The second quadrant relates to the number of features and the number of membership functions per feature. The limit of humans to process information was mentioned in the previous paragraph [93].

The third quadrant relates to the consistency of a rule-base, and the number of rules fired at the same time. A rule-base is considered consistent when there are no contradictory rules [93].

The fourth quadrant is related to the completeness, normalization and distinguishability of the membership functions describing the FIS. Completeness is a property in which given any combination of feature values at least one rule is fired, that requires that for any feature the universe of discourse is covered by at least a membership function value greater than 0. A membership function is considered normal when its maximum value is equal to 1. Distinguishability relates to the ability of a human to properly distinguish between the membership functions, this requires for the membership functions to be properly separated between each other, with little overlap between the membership functions partitions of a rule or feature [93].

For TSK FISs interpretability is considerably reduced given that the consequents of the rule-base are composed of linear regression models and not linguistic variables. The TSK FIS is a local linear model, linear regression models are transparent, given that it is possible to assess the impact of each feature on the output, these same properties allows to for the models to be interpretable to some extent. Therefore, a TSK FIS can be locally interpretable. In order to maintain the interpretability of the TSK FIS some authors have developed learning algorithms to maintain a local -global performance [98], [99].

Table 2.5: Taxonomy To Classify Interpretability [93].

	Rule base level	**Fuzzy partition level**
Complexity-based interpretability	(Q1) Complexity at the rule-base level.	(Q2) Complexity at the fuzzy partition level.
Semantic-based interpretability	(Q3) Semantics at the rule-base level.	(Q4) Semantics at the fuzzy partition level.

2.8 Summary

Fuzzy sets and logic were developed to model the complexity and vagueness of human natural language [5]. Fuzzy statements are arranged in the form of if-then rules, capable of modelling natural phenomena intuitively. This arrangement of if-then rules is defined as a FIS. The two main type of FIS are Mamdani [6] and TSK [14]; Mamdani FIS are more interpretable given it only utilizes linguistic variables to form its rule-base. TSK are more accurate, given its consequences are composed of linear functions.

The rule-base of a FIS can be generated manually utilizing expert knowledge or automatically, by generating either a grid-partition of the dataset or applying clustering algorithms. The number of rules in a grid-partition method increase exponentially with the addition of features and partitions. Clustering algorithms solve this problem by creating a partition in the feature space instead.

The performance of FIS can be further enhanced by applying learning algorithms utilized in ANN. These neuro-FISs merge the prediction accuracy of ANN and the interpretability of fuzzy logic. A special type of ANN called the RBFN can be functionally equivalent to a Mamdani FIS given certain conditions [34].

To model different phenomena several expansions to the type-1 fuzzy set have been developed, these include, fuzzy rough sets and CFS. Rough sets are composed of two approximations to represent the possible membership of an object, fuzzy rough sets expand the applicability of rough sets to add vagueness and soft boundaries to membership values. CFS add context and time to linguistic variables.

So far only three CFISs have been developed to date, these are the ANCFIS [73], CNFIS [74] and ACNFIS [75]. Results obtained from these CFISs are comparable with other known FISs such as RBFN and ANFIS. The ANCFIS was designed for time series

prediction, compared with most FIS, the ANCFIS utilize a sinusoidal membership function, the rule interference is performed by a dot product operation. Both the CNFIS and ACNFIS neglect, for the most part, the effect and meaning of the imaginary component of the CFS, furthermore neglecting the effect of the rule interference operation. None of the CFISs developed to date address the problem of interpretability.

Interpretability is a property of FIS given its proximity to human natural language. Transparency and interpretability are related but do not mean the same [90]. A mathematical definition of interpretability does not exist, but rather a set of guidelines can be implemented to evaluate the interpretability of a FIS [93].

Chapter 3

Selected Datasets for Algorithms Validation

The models elicited in this work will utilize four real world datasets. The first two are industrial datasets obtained from material testing. The third is a dataset obtained from a clinical study. The fourth is publicly available dataset.

3.1 Brief Overview of Mechanical Properties of Steel

Metallurgy is a branch of material sciences that studies the behaviour metals. The field of metallurgy is divided into two main branches, ferrous-metallurgy and nonferrous metallurgy. Ferrous metals are those metals whose main alloying element is iron. Among them, one of the most important alloys is steel whose main components are iron and carbon [100].

Metals are composed of microscopic crystal grains. Crystals are classified according to the arrangement of the atoms composing them. Iron, the main component of steel, can take three different structures, ferrite, austenite or martensite. The macrostructural properties of steel rely on the microscopic structure and arrangement of these crystals. The production, treatments and addition of alloying elements to steel change the structure and arrangements of the crystals changing its properties [100].

3.2 Charpy Impact Test

The Charpy impact test is used to measure the fracture energy absorbed by a material. A sample is placed in the Charpy impact test machine where a pendulum strikes the sample and fractures it, registering the loss of potential energy of the

pendulum as the energy absorbed by the material [101]. To facilitate a fracture, samples are machined to add a notch which creates a triaxial state of stress in the centre of the sample [101]. The resistance to fracture is called “notch toughness” [101]. Fractures can be classified as ductile or brittle, ductile fractures are associated with a higher absorption of energy compared with brittle ones [102].

The body-cantered cubic lattice structure, characteristic of iron at low temperature and present in plain carbon and low-alloy steels causes the material to become brittle at low temperatures, therefore it is observed a reduced fracture energy at those temperatures. To characterize the change from ductile to brittle fracture the Charpy impact test is performed at different temperatures. The obtained measurements are used to calculate the Charpy impact energy curves. These curves have an “S” shape as shown in Figure 3.1. The temperature range at which the materials exhibit brittle and ductile fracture are called the lower and upper shelf region respectively, the temperature of the transition region is called the Ductile to Brittle Transition Temperature (DBTT) [103].

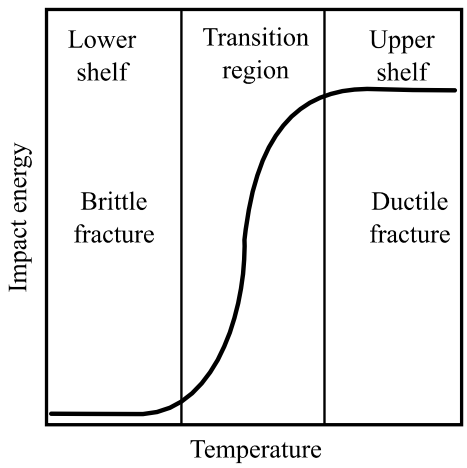


Figure 3.1: Charpy impact test DBTT curve.

The Charpy impact test presents a difficulties for modelling mainly due to the scatter in measurements [104] and the amount of inconsistencies [105], inconsistencies are

related to samples with the same or similar feature parameters and different outputs. The inconsistencies present in the dataset are attributed to features not measured in the dataset. Features, such as grain size and other micro-scale material properties, are time consuming and/or expensive to measure [106] and therefore it is not uncommon for these variables not to be found in the datasets.

The Charpy impact dataset utilized in this work consist of 1661 records, 16 features and one output which corresponds to the measured Charpy impact energy, a summary of the dataset information is shown in Table 3.1. Additionally, a partial correlation plot is shown in Figure 3.2.

Table 3.1 Charpy Impact Dataset information.

Continuous Variables	**Mean**	**Median**	**Range**
Test Dept	20.8	12.7	5.5-146.05
Sample size	172.49	155	11-381
C	0.3942	0.42	0.13-0.52
Si	0.2548	0.25	0.11-0.38
Mn	0.8409	0.82	0.41-1.75
S	0.0167	0.019	0.0008-0.052
Cr	1.0752	1.08	0.11-3.25
Mo	0.2394	0.23	0.02-0.98
Ni	0.3683	0.2	0.03-4.21
Al	0.027	0.026	0.003-0.047
V	0.0077	0.005	0.001-0.26
Hardening Temperature	864.02	860	810-980
Tempering Temperature	647.19	650	190-730
Impact Temperature	-5.7869	-10	-53 - 23
Charpy Energy	89.642	89.333	3.46-245.33
Categorical variables	**Number of categories**		
Site	3		
Cooling Medium	5		

The effects of materials alloying, and processing are highly-nonlinear, therefore from the partial correlation plot only a few conclusions can be made such as the effect of Carbon, Tempering and Impact Temperature. The effect of Carbon in steel is well known, an increase percentage of carbon increases its strength and its brittleness. Tempering is a heat treatment process in which a material is heated at certain

temperature and then cooled at a controlled temperature; tempering increases the ductility of ferrous materials.

The addition of other alloys and process are harder to measure and quantify, some alloys such Chromium and Nickel are added to a material to increase its resistance to corrosion, and therefore it is important to understand the relationship to perform a cost-benefit analysis or a trade-off between different desired material properties.

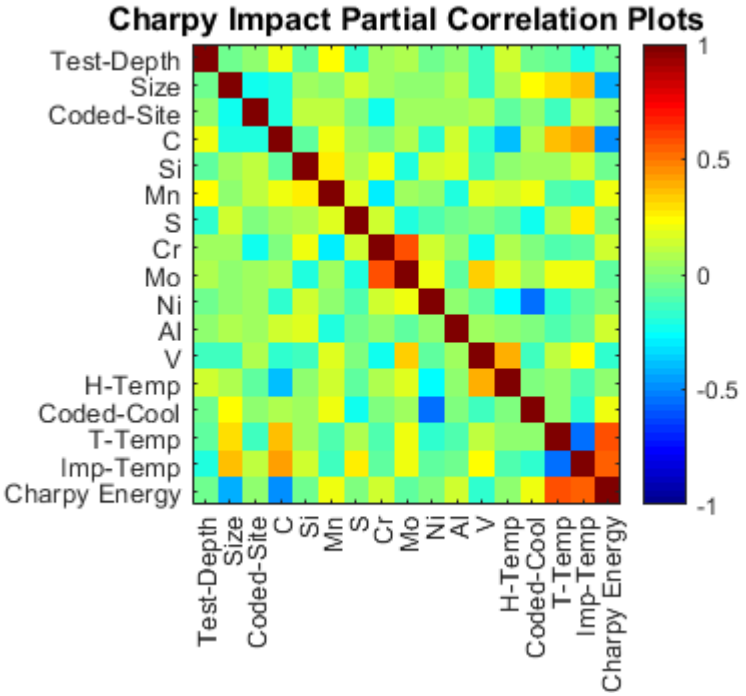


Figure 3.2: Charpy Impact partial correlation plot.

3.3 Ultimate Tensile Strength

The UTS is common measure of a material strength. In order to measure the UTS a sample is placed in a tensile test machine which applies a load at a constant speed, the deformation and required force is measured and the data is used to obtain stress-strain

curves. The UTS is defined as the maximum engineering stress and corresponds to the maximum stress measured in a stress-strain curve [101].

The UTS dataset consists of 3760 records, 15 features and one output which correspond to the UTS value. The characteristics of the dataset are shown in Table 3.2. Additionally 12 data points are used for validation, these 12 data points are outliers and therefore used to validate the generalization properties of a model [40].

Table 3.2: UTS dataset information.

Continuous Variables	**Mean**	**Median**	**Range**
Test Depth	16.08	12.7	4-140
Sample Size	156.93	150	8-381
C	0.3902	0.41	0.12-0.62
Si	0.2546	0.25	0.11-0.35
Mn	0.7524	0.73	0.35-1.72
S	0.021	0.023	0.0005-0.21
Cr	1.053	1.07	0.05-3.46
Mo	0.2631	0.23	0.01-1
Ni	0.8039	0.25	0.02-4.16
Al	0.036	0.027	0.005-1.08
V	0.0075	0.005	0.001-0.27
Hardening Temperature	856.81	850	820-980
Tempering Temperature	604.18	610	170-730
UTS	932.09	912.9	516.2-1842
Categorical variables	**Number of Categories**		
Site	6		
Cooling Medium	3		

The partial correlation plot is shown in Figure 3.3. From the partial correlation plot, similarly to the Charpy impact test, only a few conclusions can be drawn given the nonlinear relationship between alloying elements, process and material properties. It is well known that while the content of carbon increases its brittleness, it does increase its strength as well. Tempering increases the ductility of a carbon steel while decreasing strength.

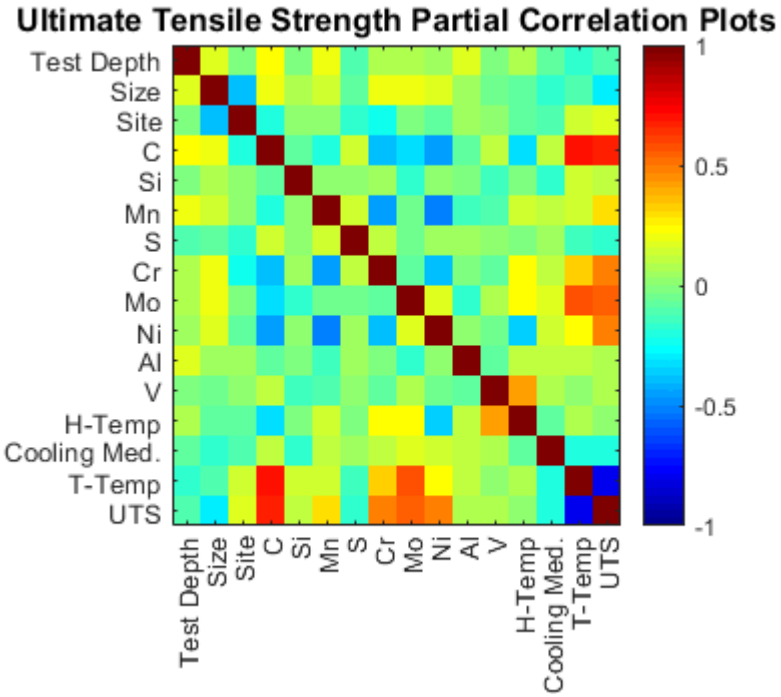


Figure 3.3: Ultimate Tensile Strength Partial correlation plot.

3.4 Bladder Cancer

Patients diagnosed with cancer are often given an estimate of the risk of death/relapse from the disease. The risk estimation is based on the lifetime expectancy after the diagnose, a common practice is to classify as high risk of mortality patients whose death may occur within the next 5 years, and low risk those patients whose life expectancy is superior to 5 years [40].

Such estimations are usually made by medical professionals, more recently prediction models are being used to assist in the diagnosis. Eliciting prediction models for medical purposes is considered a challenging task due to the presence of "censored data" [107]. In medical studies is common for patients to withdraw before completion, for the patients to die due to unrelated events, or for the patients to outlive the period of observation, when such circumstances occur, the records are marked as "right

censored" [107]. An example of the records and censoring is shown in Figure 3.4. The branch of statistics that studies time-to-event data is called survival analysis.

The dataset consists of the records obtained from 2918 patients who suffer from bladder cancer; the dataset contains 16 features and 1 output, which corresponds to time of death or last observed time. Out of the 2918 patients records, 613 are marked as non-censored. The dataset used in this work consists of the non-censored records as well as those right censored records whose last-observed time surpassed the threshold of 60 months. The resulting dataset consists of the records of 1581 patients. A summary of the dataset is shown in Table 3.3.

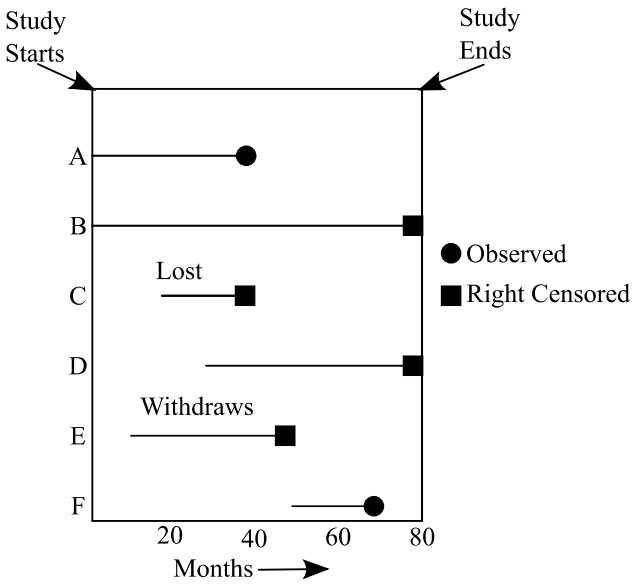


Figure 3.4: [39] Illustration of right censoring: Patients A and B, outlived the study, Patient C was lost due to an unrelated event, patient E withdrew from the study. The records of patient A and F are the only ones not censored as the time of death from the event of interest occurred within the duration of the study. The recorded time is equal to the observed time only. In this example patient C last observed time is 20 months, as the observation period begun at 20th month and was lost at the 40th month.

Patients whose last observed time is superior to 60 months are labelled as "1", while non-censored patients whose last observed time is below the 60-month threshold are

labelled as “0”. This is a simple solution that does not require application of survival analysis methods [108], which are out of the scope of this work. The dataset will be utilized as a least square problem.

Table 3.3 Bladder Cancer dataset information.

Continuous Variables	**Median**	**Mean**	**Range**
Age (years)	72.7	71.6	21.3–101
Stage	4.03	4.02	0.00–9.00
Urothelium	2	3.42	0.00–6.00
Nodes detail	4	3.94	0.00–4.00
Categorical Variables	**Values**	**Number of Patients**	**Percentage**
Sex	Male	2129	0.7296
	Female	789	0.2704
Tumour grade	Good	736	0.2522
	Moderate	956	0.3276
	Poor	1226	0.4202
Squamous	No	2789	0.9558
	Yes	129	0.0445
CIS Present	No	2548	0.8732
	Yes	370	0.1268
SPB	Solid	492	0.1686
	Papillary	1856	0.6361
	Both	570	0.1953
Vascular invasion	No	2701	0.9256
	Yes	217	0.0744
Muscle invasion	No	816	0.2796
	Yes	2102	0.7204
Cystectomy	No	2886	0.989
	Yes	32	0.011
Radiotherapy	No	2854	0.9781
	Yes	64	0.0219

3.5 Superconductivity

Superconductors are materials known to have near zero resistance when their temperature is bellow a critical temperature [109]. The superconductivity dataset consists of 21263 instances, 80 features and 1 output which corresponds to the critical temperature of such semiconductors [109], [110].

3.6 Summary

A brief overview of the datasets explored on this work has been presented, each dataset present different challenges. The partial correlation plots for the Charpy impact test and UTS were able to describe, certain behaviour that has been well understood in material science. It is clear the limitations of utilizing linear statistical methods for knowledge extraction.

The Bladder cancer dataset present difficulties given the number of censored data present in clinical studies. A modelling approach is presented that do not require the application of statistical survival analysis tools, allowing to model the dataset utilizing a least squares algorithm.

The super conductivity dataset contains a large number of features and instances, therefore the results obtained would validate the application of the developed algorithms for large datasets.

Given the known difficulties of modelling the Charpy impact test dataset, this set will be analysed and tested in greater detail in comparison with the other datasets to demonstrate the capabilities of the models and tools developed.

Chapter 4

The Single Input Complex Fuzzy Inference System Model

4.1 Introduction

Complex Fuzzy Logic (CFL) and CFSs expand the traditional type-1 fuzzy sets and logic to the unit circle. CFS and logic was initially developed by Ramot et al. who proposed the utilization of CFS to model periodic data [8] [55].

Most of the developed CFIS so far have explored the ability to represent approximate periodic data with CFS and have produced highly accurate results. Regardless of the achievements of these CFIS, the problem of interpretability has not been fully addressed though.

According to Ramot et al. [55] the development of a CFL should retain the properties of traditional fuzzy logic and benefit from the use of complex numbers; the authors point out to the following properties: 1) The framework should handle numerical data and linguistic knowledge. 2) CFL system must remain simple and intuitive. 3) Rules should be fired in parallel for efficiency [55].

The proposed Single Input Complex Fuzzy Inference System (SICFIS) model was developed in accordance with these three requirements. In order to create an interpretable CFIS the structure needs to remain as simple as possible: the SICFIS model represent a single-feature-partition-per-rule CFIS where the premises are composed of type-1 fuzzy Gaussian membership functions and the consequences are complex fuzzy singleton membership functions. This simple structure allows the user

to identify the relationship between features partitions based on the phase difference of the consequences, additionally the system is capable of handling continuous, categorical and linguistic data.

The simple structure of the SICFIS presents several advantages: 1. The number of parameters grows linearly with the number of features in the dataset. 2. The combinatorial rule explosion problem is avoided. 3. It is not necessary to execute a clustering algorithm or the assistance of expert knowledge to create an initial rule-base. Therefore, training time is reduced considerably since the number of operations and parameters are lower than traditional FISs. Additionally, a parsimonious model should be able to reduce the probability of overfitting [111].

In this chapter the SICFIS model is tested on three different datasets. The first dataset is used for the prediction of a Charpy impact test in steel. The second dataset is used for prediction of the UTS of steel. The third dataset consists on predicting the risk of mortality for bladder cancer patients. Results obtained from the three different datasets show an equivalent level of accuracy as RBFN, ANFIS models, simple ANN as well as other type-1 and type-2 FISs. An interpretability analysis applied to the Charpy impact test will demonstrate that the knowledge extracted from the model is consistent with what is known in the literature.

4.2 The Single Input Complex Fuzzy Inference System Model

Most of the applications of CFS, as originally proposed in [8], have mainly focused on modelling datasets which contain approximately periodic data. However, to illustrate the applicability as well as the advantage of CFSs in generic data modelling problems, Ramot et al. proposed an application where CFSs are used to predict voter turnout in an election [55] through the use of the two rules shown in Table 4.1.

According to Ramot et al. while each individual rule when true provides a *high* and *very high* voter turnout, when both of them are true, the voter turnout is in fact *Low* [55]. This phenomenon can be easily and elegantly modelled by assigning different phases to each rule in order to cause a destructive interference. The proposed SICFIS model expands on this idea to create a compact model capable of modelling the complex interaction between feature partitions.

The SICFIS model is a single-feature-partition-per-rule CFIS. Compared with traditional rule-base FIS the SICFIS model utilizes CFS to represent each consequence as a two-dimensional vector. Because each vector has a direction and a magnitude, it is possible to model the interaction between partitions as interferences, thus avoiding the problem of combinatorial rule explosion [17], or the need to apply a clustering or granulation algorithm to derive an rule-base for the system.

Table 4.1: Complex fuzzy rule-base to determine voter turnout in an election.

Premise	Consequence
Rule 1: IF “Confidence in Democracy” is “High”	THEN “Voter Turnout” is “High”
Rule 2: IF “Disenchantment with Leaders” is “High”	THEN “Voter Turnout” is “Very High”

While previously developed CFISs have focused on compactness and accuracy, none of them addresses the problem of interpretability of CFS.

A similar model was proposed in [112]. Although the proposed methods are similar, the authors of [112] fail to provide any results. Additionally the equations presented are identical as the ones presented in the real-valued SIRM model proposed in [15]. Therefore, due to the lack of results and evidence provided in [112] , the SICFIS model proposed in this work is the first interpretable CFIS.

4.2.1 The Single Input Complex Fuzzy Inference System Membership Function

The SICFIS model utilizes a real valued Gaussian membership function for the premises, for a feature p and a partition s_p the membership function is as follows:

$$\mu_{p,s_p} = \exp\left(-\frac{1}{2}\left(\frac{x_p - c_{p,s_p}}{\sigma_{p,s_p}}\right)^2\right) \tag{4.1}$$

where c and σ are the centre and the spread of the Gaussian membership function respectively.

For the consequences a complex singleton membership function is used as follows:

$$\zeta_{p,s_p} = \beta_{p,s_p} \cdot e^{j\varphi_{p,s_p}(x)} \tag{4.2}$$

$$\zeta_{p,s_p}^{\mathrm{Re}} = \beta_{p,s_p} \cos(\varphi_{p,s_p}) \tag{4.3}$$

$$\zeta_{p,s_p}^{\mathrm{Im}} = \beta_{p,s_p} \sin(\varphi_{p,s_p})\, j \tag{4.4}$$

where β represents the magnitude and φ represents the phase. Equations (4.3) and (4.4) show rectangular coordinates of the singleton membership function; both parameters β and φ are real-valued scalars.

4.2.2 The Single Input Complex Fuzzy Inference System Model Architecture

The SICFIS is a Mamdani CFIS with singleton defuzzification, therefore the architecture resembles that of a RBFN model. The implication operation is the *algebraic product and* the aggregation operation is the *vector aggregation method*. For the vector aggregation the complex weights will be eliminated.

Each feature p is to be partitioned into s_p partitions (e.g. Low-Medium-High) and, each partition will be assigned a real valued Gaussian membership function, (4.1). The rule consequences are represented by the complex singleton membership function, (4.2). The parameters in the model are real-valued therefore traditional optimization methods can be implemented.

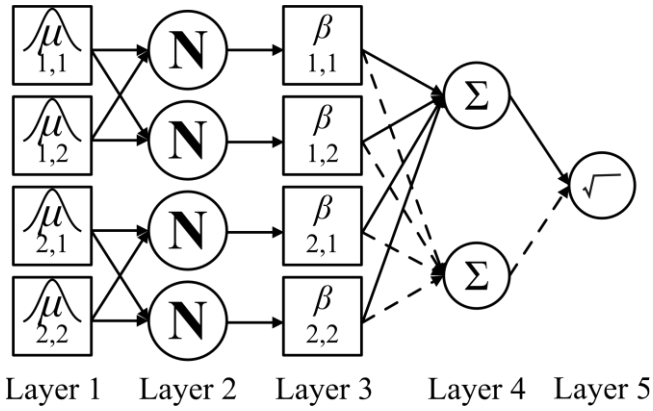


Figure 4.1: The SICFIS schematic.

The SICFIS is a 5-layer model as shown in Figure 4.1. The first layer is the fuzzification layer which assigns a degree of membership to a partition s_p of a feature p, according to:

$$O^1_{p,s_p} = \mu_{p,s_p} = \exp\left(-\frac{1}{2}\left(\frac{x_p - c_{p,s_p}}{\sigma_{p,s_p}} \right)^2 \right) \tag{4.5}$$

The second layer performs a normalization operation for the s_p partitions of a feature p as follows:

$$O^2_{p,s_p} = \frac{\mu_{p,s_p}}{\sum_{s_p=1}^{S_p} \mu_{p,s_p}} \tag{4.6}$$

The third layer performs the implication operation. The algebraic product is selected as the implication operation. The output of the second layer (4.6) multiplies the complex singleton membership function, (4.2). The rectangular form of the complex singleton membership function is used in order to facilitate calculations as follows:

$$O^3_{\mathrm{Re},p,s_p} = O^2_{p,s_p} \cdot \cos(\varphi_{p,s_p}) \cdot \beta_{p,s_p} \tag{4.7}$$

$$O^3_{\mathrm{Im},p,s_p} = O^2_{p,s_p} \cdot \sin(\varphi_{p,s_p}) \cdot \beta_{p,s_p} \tag{4.8}$$

The third layer is the vector aggregation (or rule interference) layer in which the real and imaginary parts are added respectively as follows:

$$O^4_{\mathrm{Re}} = \sum_{p=1}^{P} \sum_{s_p=1}^{S_p} O^3_{\mathrm{Re},p,s_p} \tag{4.9}$$

$$O^4_{\mathrm{Im}} = \sum_{p=1}^{P} \sum_{s_p=1}^{S_p} O^3_{\mathrm{Im},p,s_p} \tag{4.10}$$

The fifth layer calculates the magnitude and the phase of the resultant vector as follows:

$$O^5 = \left|O^4\right| \angle \arg\left(O^4\right) \tag{4.11}$$

The magnitude of the resultant vector is utilized as the final output of the model to evaluate its performance; the phase may be used as additional information to improve the interpretability of the system. Particularly, as it will be demonstrated in this work.

4.3 Model Initialization

In order to improve the results from the optimization it is important to select a valid initial model since a randomly or an inadequately initialized model is more likely to drive the optimization algorithm into a sub-optimal solution. The initialization of the model works as follows: for the premises a grid partition of the data is performed, each feature p will be divided into s_p partitions (Figure 4.3) , each partition will have a centre and Standard Deviation (SD) as is recommended in [113], where the membership values are continuous and the partition intersect at approximately *0.5* membership value as shown in Figure 4.2 a. For the complex consequences a phase is assigned to each membership function, with the values of the phases being linearly spaced between *0* and 2π as shown in Figure 4.2 b. The initial values β are obtained from the coefficients of a partial correlation (PC) analysis as follows:

$$PC = \frac{N\sum_{i=1}^{N}\varepsilon_{X,i}\varepsilon_{Y,i} - \sum_{i=1}^{N}\varepsilon_{X,i}\sum\varepsilon_{Y,i}}{\sqrt{N\sum_{i=1}^{N}\varepsilon_{X,i}^2 - \left(\sum_{i-1}^{N}\varepsilon_{X,i}\right)^2}\sqrt{N\sum_{i=1}^{N}\varepsilon_{Y,i}^2 - \left(\sum_{i-1}^{N}\varepsilon_{Y,i}\right)^2}} \tag{4.12}$$

where ε are the residuals obtained from a linear regression and X,Y are the datasets. The process is shown in Algorithm 4.1.

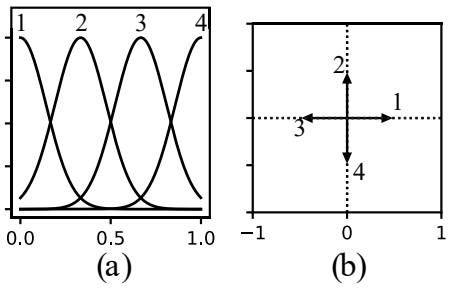


Figure 4.2: (a) Initial grid partition for a feature p. (b) Initial vector assigned to the output of a rule, with a length equal to β_{p,s_p} and an phase equal to φ_{p,s_p}.

Algorithm 4.1: SICFIS initialization.

Inputs: Number of features P, Number of partitions s_p for each feature p, partial correlation analysis PC (4.12).
Outputs: Rule output parameter β, consequent membership function parameter φ, premise membership function parameter σ, premise membership function parameter c.

$p \leftarrow 1$
while $p < P$
  $k \leftarrow 1$
  while $k < s_p$
    $\beta_{p,k} \leftarrow PC_p$
    $\varphi_{p,k} \leftarrow (k-1)(2\pi)/s_p$
    $\sigma_{p,k} \leftarrow 1/(2.3333(s_j - 1))$
    $c_{p,k} \leftarrow 1/(k(s_j - 1))$
    $k \leftarrow k+1$
  $p \leftarrow p+1$
  $k \leftarrow 1$

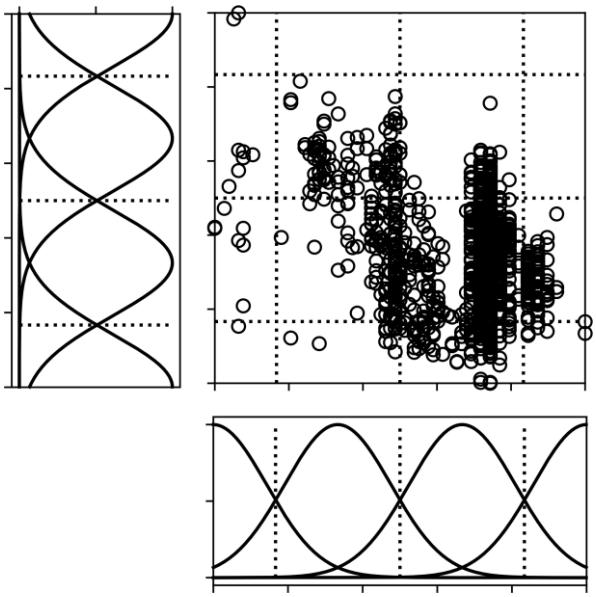


Figure 4.3: Example of a grid partition of a two-dimensional dataset.

4.4 Interpretability, Transparency and Knowledge Extraction

4.4.1 Interpretability Concepts and Comparisons with Traditional Fuzzy Rule-Base Models

The SICFIS model has several advantages over traditional fuzzy rule-base systems. In order to highlight these advantages as well as some considerations to be made for assessing interpretability the taxonomy introduced in section 2.7.1 will be used.

4.4.1.1 First Quadrant: Complexity at the Rule-Base Level:

The number of rules of the SICFIS is much lower than that of grid-partition based methods; the combinatorial rule explosion problem is avoided given that the number of rules grow linearly with the addition of features and partitions. Given that the number of rules is equal to the number of features and partitions, the number of rules for the SICFIS can be greater than that of cluster-based methods.

The number of conditions per rule is clearly reduced since the SICFIS model is a single feature partition per rule FIS. The number of conditions per rule in both grid-partition and cluster-based methods is usually equal to the number of features in the dataset.

4.4.1.2 Second Quadrant: Complexity at the Level of Fuzzy Partitions:

The number of conditions per feature is considerably reduced in the SICFIS. While the number of conditions per feature in cluster-based methods is equal to the number of clusters or rules, and the number of conditions per features in gird partition methods is equal to the size of the grid, it will be demonstrated in the following sections that a

superior performance be achieved with the SICFIS model in comparison with traditional FIS models with as few as 3 partitions per feature.

4.4.1.3 Third Quadrant: Semantics at the Rule-Base Level:

The problem of two or more contradictory rules being fired at the same time is avoided completely, given that a rule corresponds to the behaviour of a specific feature partition, the concept of contradiction does not apply to the SICFIS model. Additionally, the main characteristic of the SICFIS model is the ability to model the interaction between feature partitions as interferences.

4.4.1.4 Fourth Quadrant Semantics at the Fuzzy Partition Level:

In a traditional fuzzy rule-base model completeness in the system is achieved only if all features are complete. In a SICFIS model incompleteness in the system would require incompleteness in all the features. Additionally incompleteness in a feature would signal a lack of effect in such region in the overall output of the model, therefore incompleteness would not be considered as an error entirely, but such assumptions would require analysing the results, as it is possible that the incompleteness is due to a lack of data points in such regions.

4.4.2 Knowledge Representation with the SICFIS Model

It is well known that the visual representation of machine learning and AI models facilitates the extraction of knowledge of a system and increases its interpretability.

The SICFIS model specific properties allows for the representation of knowledge in different forms, presenting an additional advantage over traditional fuzzy rule-base models. In the following subsections different forms of representing knowledge will be

introduced, an accompanying mock-up example will be used to demonstrate these representation forms.

4.4.2.1 Magnitude-Phase Plots

The Magnitude-Phase plots are composed of the resultant magnitude and phase of each individual feature *p* for a specific range of operation. The calculation of the magnitude (4.13) and the phase (4.14) for a feature *p* is calculated as follows:

$$Mag_p = \left| \sum_{s_p=1}^{S_p} \left(\bar{\mu}_{p,s_p}(k_p) + \cos(\varphi_{p,s_p}) \cdot \beta_{p,s_p} \right) + \sum_{s_p=1}^{S_p} \left(\bar{\mu}_{p,s_p}(k_p) + \sin(\varphi_{p,s_p}) \cdot \beta_{p,s_p} \right) j \right| \quad (4.13)$$

$$Ph_p = \arg\left(\sum_{s_p=1}^{S_p} \left(\bar{\mu}_{p,s_p}(k_p) + \cos(\varphi_{p,s_p}) \cdot \beta_{p,s_p} \right) + \sum_{s_p=1}^{S_p} \left(\bar{\mu}_{p,s_p}(k_p) + \sin(\varphi_{p,s_p}) \cdot \beta_{p,s_p} \right) j \right) \quad (4.14)$$

where $\bar{\mu}_{p,s_p}(k_p)$ is the normalized firing rule strength of a feature *p* and partition s_p which corresponds to the output of the second layer of the SICFIS model (4.6), k_p is a continuous variable with strictly increasing values within the specified range of operation of a feature *p*. The transparency of the system can be demonstrated utilizing the information contained in the magnitude-phase plots, as the behaviour of the system for any combinations of values within a range of operation can be assessed and measured. An example of a magnitude-phase plot is shown in Figure 4.6.

4.4.2.2 Fuzzy Rules-Base Derived From SICFIS

Even though the SICFIS is not a traditional rule-base it can however represent one. A grid partition rule-base can be created by measuring the resultant vector of all possible combinations of feature partitions. The problem of combinatorial rule explosion can be avoided by creating short rules [114] utilizing only the most important

feature partitions which can be easily assessed by measuring the magnitude of each feature partition. This provides an additional level of control over the granularity and interpretability of the model. Table 4.2 shows an example of a small SICFIS rule-base and Table 4.3 shows the derived grid-partition rule-base from the SICFIS rule-base.

Table 4.2: Example of a SICFIS rule-base.

Premise	Consequence	
If A_1 is "High"	Then	*B1*
If A_1 is "Low"	Then	*B2*
If A_2 is "High"	Then	*B3*
If A_2 is "Low"	Then	*B4*

Table 4.3: Example of the derived grid-partition rule-base from the SICFIS rule-base.

Premise	Consequence	
If A_1 is "High" and A_2 is "High"	Then	$\lvert B1+B3\rvert$
If A_1 is "High" and A_2 is "Low"	Then	$\lvert B1+B4\rvert$
If A_1 is "Low" and A_2 is "High"	Then	$\lvert B2+B3\rvert$
If A_1 is "Low" and A_2 is "Low"	Then	$\lvert B2+B4\rvert$

4.4.2.3 Vector Partition Plot

The vector partition plots shows two different graphs, the first one shows how a feature *p* is partitioned into the different membership function μ_{p,s_p} for $s_p = 1,\cdots,S_p$, the second graph represents graphically the consequence corresponding to the partitions of the feature *p* as a two dimensional vector with a magnitude β_{p,s_p} and an phase φ_{p,s_p} . The vector partition plot presents the rules premises and consequences in an orderly manner. This allows the user to identify and measure the interaction between different partitions corresponding to different features. An example of the vector partition plot of three features is shown in Figure 4.4.

4.4.2.4 Cosine Distance Matrix Plot

The cosine distance matrix plot represents the level interference between each two partition consequences, with a number within [-1,1], to represent degree to which an interference is destructive or constructive respectively. The cosine distance matrix information, can be used just as a Pearson correlation matrix plot to derive knowledge, compared with the correlation matrix, the cosine distance matrix is able to represent the non-linear relationship between the different partitions. An example of the cosine distance matrix plot is shown in Figure 4.5.

4.4.3 Example of the Application of the SICFIS to Model Material Properties

Interferences can be used to model the complicated relationship between material alloys and process to the properties of the materials. In order to demonstrate how the SICFIS model can be used to model features as interference a simple mock-up example can be stated as follows:

It is known that increasing the percentage of carbon in steel improves its strength until a threshold is met, any addition of carbon beyond this threshold will decrease its strength as the material becomes too brittle. The content of carbon can be labelled as low (L), medium (M), high (H) and very high (VH). For this example, two more features are included, one is the content of iron, and finally let's assume that a process "X" is applied to the material in order to improves its properties. For simplicity let's assume the effect of the content of iron and the process "X" is the same for the whole range of possible input values, and therefore a feature partition of the iron and the process "X" will not be created as it is in the case of carbon.

It is assumed that as the content of carbon increases from L to M, the strength increases, therefore there is a constructive interference between the content of iron and carbon for these partitions. The threshold in which the addition of steel becomes

detrimental to its strength is met when the content is H, and is completely detrimental when it reaches VH, then we can infer that the output vector H is orthogonal to that of the content of iron output vector, and for VH a destructive interference occurs. Further let's suppose that the process "X" is known to improve the strength of high carbon steel and has little or no effect for low, medium or very high carbon steel, meaning that a constructive interference occurs with the H carbon partition, and for the rest of the input values little or no interference occurs.

The SICFIS rule-base is shown in Table 4.4, the corresponding grid partition rule-base is shown in Table 4.5. It is clear how the SICFIS rule-base contains fewer rules than that of the grid partition, the difference becomes greater as more feature partitions are created as the number of rules grows exponentially for the grid partition fuzzy rule-base system and linearly for the SICFIS model.

Figure 4.4 shows the vector partition plot of this model. As mentioned previously, the carbon content is partitioned into 4 membership functions, and the corresponding output of each rule is shown below. No feature partition is implemented for the iron content and the process 'X', therefore only one output vector is assigned. Figure 4.5 shows the cosine distance matrix plot which shows the degree of interference between the different feature partitions. Figure 4.6 shows the corresponding magnitude-phase plots, which represents the magnitude and phase values of the feature vector for all the possible values within the range of operation.

Table 4.4: SICFIS rule-base.

	Premise				Consequence	
1)	If	C	is	L	Then	B1
2)	If	C	is	M	Then	B2
3)	If	C	is	H	Then	B3
4)	If	C	is	VH	Then	B4
5)	If	Fe	is	y	Then	B5
6)	If	X	is	x	Then	B6

Table 4.5: Grid partition rule-base.

	Premise		Consequence
1)	If C is L and Fe is y	Then	B1
2)	If C is M and Fe is y	Then	B2
3)	If C is H and Fe is y	Then	B3
4)	If C is VH and Fe is y	Then	B4
5)	If C is L and Fe is y and X is x	Then	B5
6)	If C is M and Fe is y and X is x	Then	B6
7)	If C is H and Fe is y and X is x	Then	B7
8)	If C is VH and Fe is y and X is x	Then	B8

Figure 4.7 shows the results given three different scenarios: a) the total strength of a high carbon steel when the process “X” is not applied, b) the total strength of high carbon steel when the process “X” is applied and c) the total strength of medium carbon steel when the process “X” is applied. From the results it can be confirmed that the process “X” increases the strength of high carbon steel and has little effect on medium carbon steel. Additionally, the high carbon steel with the process “X” has the same strength as medium carbon steel. It is demonstrated with this simple example how the CFS can be used to model the complex interaction between alloying elements and process utilizing interferences.

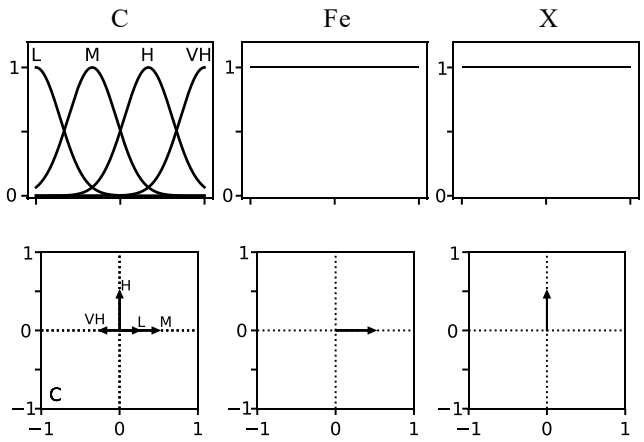


Figure 4.4: Vector partition plot for Carbon (C), Iron (Fe) and the process “X”.

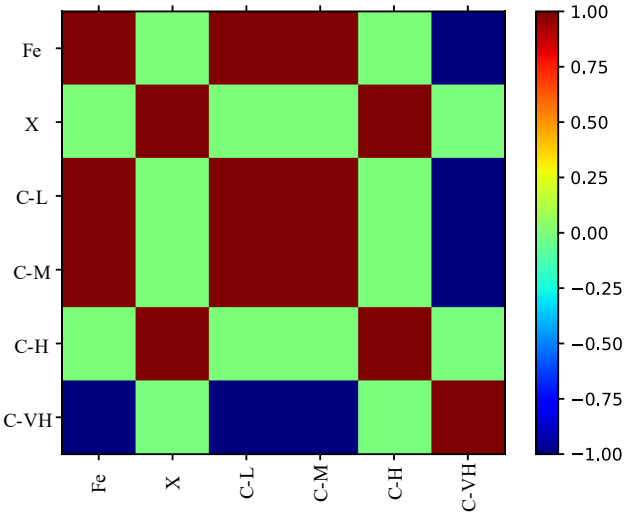


Figure 4.5: Cosine distance matrix plot for Carbon (C), Iron (Fe) and the process “X”.

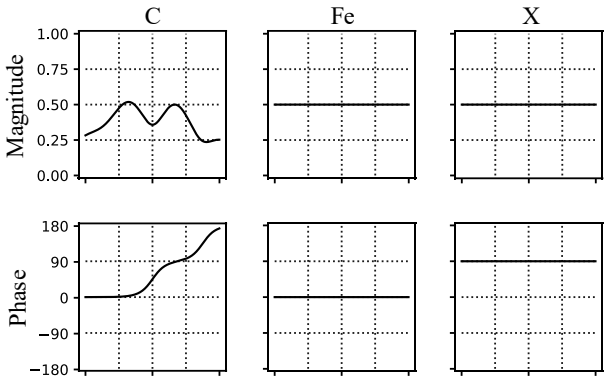


Figure 4.6: Magnitude Phase plots for Carbon (C), Iron (Fe) and the process “X”.

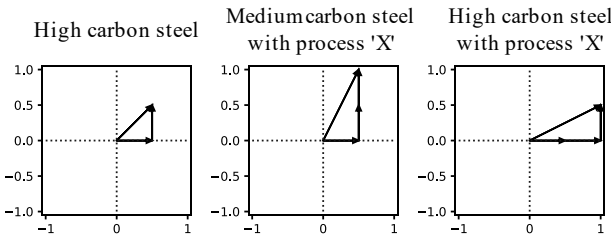


Figure 4.7: Resultant vector for high carbon steel, medium carbon steel with process “X” and high carbon steel with process “X”.

4.5 Optimization

In order to improve the system performance, the parameters are to be updated utilizing a gradient-based learning algorithm. The error-backpropagation algorithm utilizes squared error as an objective function. The derivatives of the parameters with respect of the function are as follows:

$$\frac{\partial f}{\partial \beta_{p,s_p}} = \left(\frac{\partial f}{\partial h_{\mathrm{Re}}} \frac{\partial h_{\mathrm{Re}}}{\partial \beta_{p,s_p}} + \frac{\partial f}{\partial h_{\mathrm{Im}}} \frac{\partial h_{\mathrm{Im}}}{\partial \beta_{p,s_p}} \right) \tag{4.15}$$

$$\frac{\partial f}{\partial \varphi_{p,s_p}} = \left(\frac{\partial f}{\partial h_{\mathrm{Re}}} \frac{\partial h_{\mathrm{Re}}}{\partial \varphi_{p,s_p}} + \frac{\partial f}{\partial h_{\mathrm{Im}}} \frac{\partial h_{\mathrm{Im}}}{\partial \varphi_{p,s_p}} \right) \tag{4.16}$$

$$\frac{\partial f}{\partial \sigma_{p,s_p}} = \left(\frac{\partial f}{\partial h_{\mathrm{Re}}} \frac{\partial h_{\mathrm{Re}}}{\partial \mu_{p,s_p}} \frac{\partial \mu_{p,s_p}}{\partial \sigma_{p,s_p}} + \frac{\partial f}{\partial h_{\mathrm{Im}}} \frac{\partial h_{\mathrm{Im}}}{\partial \mu_{p,s_p}} \frac{\partial \mu_{p,s_p}}{\partial \sigma_{p,s_p}} \right) \tag{4.17}$$

$$\frac{\partial f}{\partial c_{p,s_p}} = \left(\frac{\partial f}{\partial h_{\mathrm{Re}}} \frac{\partial h_{\mathrm{Re}}}{\partial \mu_{p,s_p}} \frac{\partial \mu_{p,s_p}}{\partial c_{p,s_p}} + \frac{\partial f}{\partial h_{\mathrm{Im}}} \frac{\partial h_{\mathrm{Im}}}{\partial \mu_{p,s_p}} \frac{\partial \mu_{p,s_p}}{\partial c^{s}_{p,s_p}} \right) \tag{4.18}$$

The parameters to be updated can be stored in a single vector $\mathbf{w}$ as follows:

$$\mathbf{w} = \left[\beta_{1,1} \cdots \beta_{P,S_P} \quad \varphi_{1,1} \cdots \varphi_{P,S_P} \quad \sigma_{1,1} \cdots \sigma_{P,S_P} \quad c_{1,1} \cdots c_{P,S_P} \right] \tag{4.19}$$

Three different backpropagation algorithms are implemented to evaluate their performance measured using the Root Mean Squared Error (RMSE), the first one is a recursive backpropagation algorithm, the second one a batch backpropagation algorithm and finally the LM optimization algorithm.

4.5.1 Recursive Backpropagation

The recursive backpropagation algorithm is specially utilized in dynamical systems. The parameters are updated utilizing the information of each new sample obtained. For information tables or datasets, the recursive backpropagation calculates the gradient of the squared error of each sample in the dataset.

Each iteration of the algorithm is called an epoch. The process is repeated until an end condition is met, such as conditions may include reaching a maximum number of epochs or a local minimum.

Results from a recursive backpropagation for the Charpy impact test dataset is shown in Figure 4.8. The model was trained for 50 epochs, taking a total of 1937 seconds to be computed in a computer Windows 10 with a processor intel i5-9400F @ 2.90 GHz with an installed memory RAM of 8.00GB, and a Graphic Processing Unit (GPU) NVIDIA 1660 6GB.

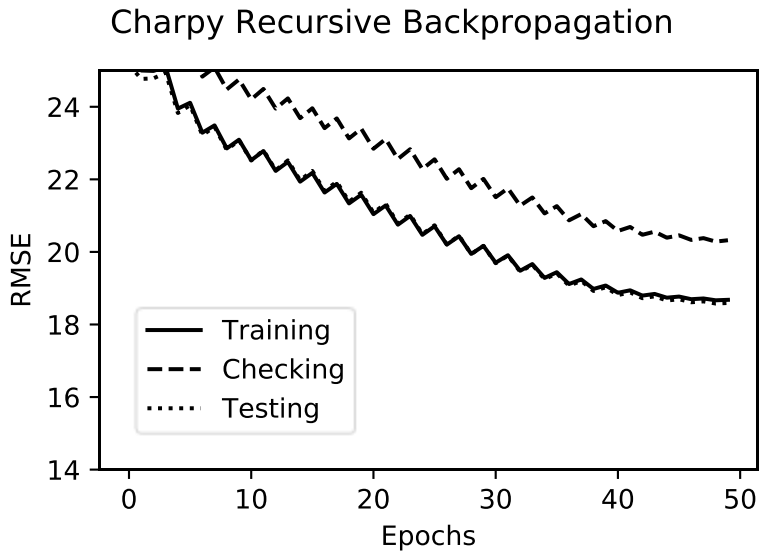


Figure 4.8: Charpy recursive backpropagation RMSE at each epoch.

4.5.2 Batch Backpropagation

Recursive backpropagation algorithms are effective for real-time applications. For other tasks it is more computationally efficient to calculate the Jacobian matrix of the parameters for all the records in the dataset; this is defined as batch backpropagation. The algorithm is more efficient if parallel computation with GPUs is implemented. Results from a batch backpropagation for the Charpy impact test dataset is shown Figure 4.9. The model was trained for 2000 epochs, taking a total of 72 seconds to be computed in the same computer mentioned in the previous section, the GPU was utilized for parallel computing for both algorithms. Given that the batch backpropagation calculates the Jacobian matrix in a single operation, an exponential reduction in computing time can be observed.

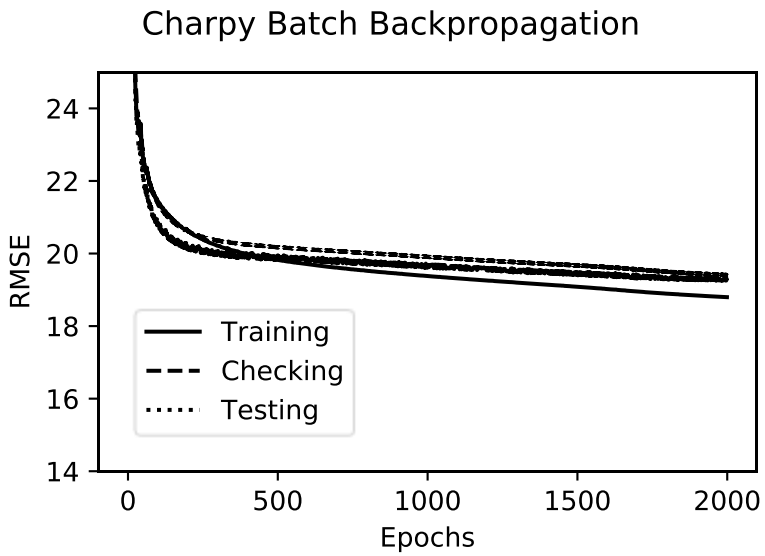


Figure 4.9: Charpy batch backpropagation RMSE at each epoch.

4.5.3 Levenberg-Marquardt Optimization

The recursive and batch backpropagation utilizes the information of the first derivatives to find the local minima. It is possible to improve the optimization model by including the information obtained from the second derivative. Algorithms that utilize the second derivative are known as Newton-Raphson methods, and require the computation of the Hessian matrix. For large models computing the Hessian matrix becomes intractable [115]. The LM algorithm [116] utilizes an approximation of the Hessian matrix utilizing the Jacobian that results in a fast and efficient optimization algorithm shown in Algorithm 4.2.

Figure 4.10 shows the training performance of the LM algorithm for each epoch applied to the Charpy impact test dataset. The model was trained for 40 epochs, taking a total of 2.8 seconds to be computed in the same computer mentioned in the previous section. The LM shows superior performance compared with both the recursive and batch backpropagation algorithms, a further exponential reduction in computing time is achieved by parallel computing and utilizing the approximation to the Hessian matrix.

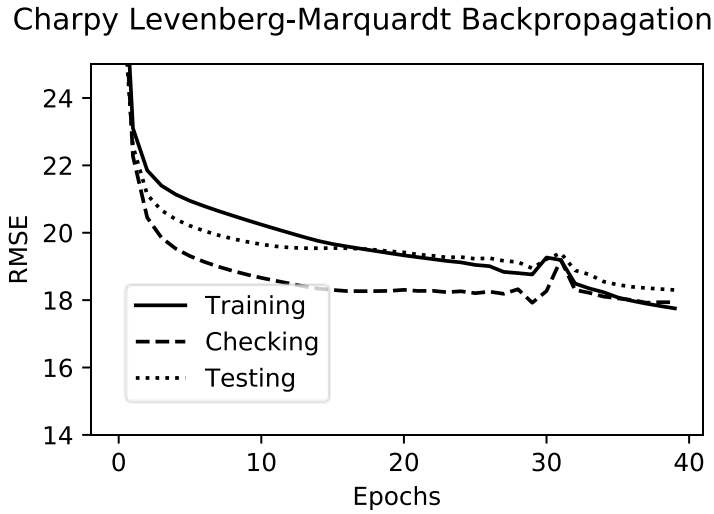


Figure 4.10: Charpy LM RMSE at each epoch.

Algorithm 4.2: Levenberg-Marquardt optimization

Inputs: initial parameter vector $\mathbf{w}_0$, dataset input vector $\mathbf{x}$,dataset output vector $\mathbf{y}$, SICFIS output vector $\hat{\mathbf{y}} = [f(x_i, \mathbf{w})]^{\mathbf{T}}$ for $i = 1, 2, \cdots, N$, LM coefficient γ, LM coefficient modifiers $\delta_a < 1 < \delta_b$, identity matrix $\mathbf{I}$, end condition threshold η, maximum number of epochs m.

Outputs: optimized parameter matrix $\mathbf{w}$

While $t < m$ or $\left| \frac{1}{2} \mathbf{e}_{t+1} \mathbf{e}^{\mathbf{T}}{}_{t+1} - \frac{1}{2} \mathbf{e}_t \mathbf{e}^{\mathbf{T}}{}_t \right| < \eta$

Compute Jacobian matrix $\mathbf{J}_t$

$\Delta \mathbf{w}_t \leftarrow -\left[\mathbf{J}_t^{\mathbf{T}} \mathbf{J}_t + \gamma_t \mathbf{I} \right]^{-1} \mathbf{J}_t^{\mathbf{T}} \mathbf{e}_t$

$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \Delta \mathbf{w}_t$

$\hat{\mathbf{y}}_{t+1} \leftarrow \mathbf{f}(\mathbf{x}, \mathbf{w}_{t+1})$

$\mathbf{e}_{t+1} \leftarrow (\hat{\mathbf{y}}_{t+1} - \mathbf{y})$

If $\frac{1}{2} \mathbf{e}_{t+1} \mathbf{e}^{\mathbf{T}}{}_{t+1} < \frac{1}{2} \mathbf{e}_t \mathbf{e}^{\mathbf{T}}{}_t$

Then: $\gamma_{t+1} = \delta_a \cdot \gamma_t$

Else: $\gamma_{t+1} = \delta_b \cdot \gamma_t$

$t \leftarrow t + 1$

4.6 A Faster SICFIS Model

The SICFIS model presented in this Chapter presents a simple architecture, being possible to train models within a few seconds with the addition of parallel computing. By making a few modifications to the model, it is possible to obtain a faster SICFIS model, reducing the training times even further. It is possible to obtain an equivalent model, which maintains the advantages presented in section 4.4, by removing the normalization operation in the second layer. This modification reduces the number of operations considerably, especially for larger datasets. The fast-SICFIS model is a 4-layer system as observed in Figure 4.11.

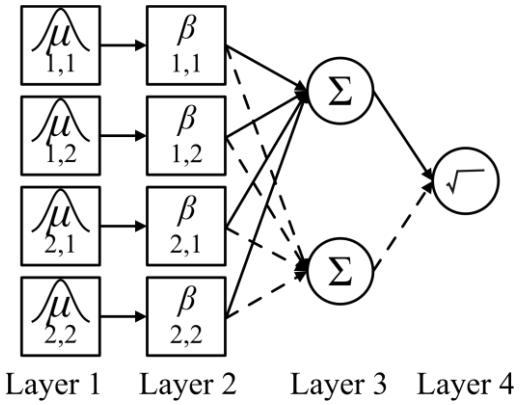


Figure 4.11: The fast-SICFIS schematic.

The first layer is the fuzzification layer which assigns a degree of membership to a partition s_p of a feature p, according to

$$O^1_{p,s_p} = \mu_{p,s_p} \tag{4.20}$$

The second layer is the implication operation, which multiplies the premises and the consequences. The rectangular form of the complex singleton membership function is used in order to facilitate calculations as follows:

$$O^2_{\mathrm{Re},p,s_p} = O^1_{p,s_p} \cdot \cos(\varphi_{p,s_p}) \cdot \beta_{p,s_p} \tag{4.21}$$

$$O^2_{\mathrm{Im},p,s_p} = O^1_{p,s_p} \cdot \sin(\varphi_{p,s_p}) \cdot \beta_{p,s_p} \tag{4.22}$$

The third layer is the vector aggregation (or rule interference) layer in which the real and imaginary parts are added respectively as follows:

$$O^3_{\mathrm{Re}} = \sum_{p=1}^{P} \sum_{s_p=1}^{S_p} O^2_{\mathrm{Re},p,s_p} \tag{4.23}$$

$$O^3_{\mathrm{Im}} = \sum_{p=1}^{P} \sum_{s_p=1}^{S_p} O^2_{\mathrm{Im},p,s_p} \tag{4.24}$$

The fourth layer calculates the magnitude and the phase of the resultant vector as follows:

$$O^4 = \left|O^3\right| \angle \arg\left(O^3\right) \tag{4.25}$$

4.6.1 Performance Comparison Between the Normalized-SICFIS and the Fast-SICFIS

The Charpy impact dataset will be utilized to compare the normalized-SICFIS model and the fast-SICFIS model training times and performance. The LM algorithm presented in previous section provided the best results and will be the one selected for this analysis. Each feature is partitioned into three partitions. The models are trained from 20 to 70 epochs, the RMSE is utilized to measure the performance, a 5 k-fold cross validation is applied; the mean RMSE at each epoch is recorded. The results are shown in Figure 4.12.

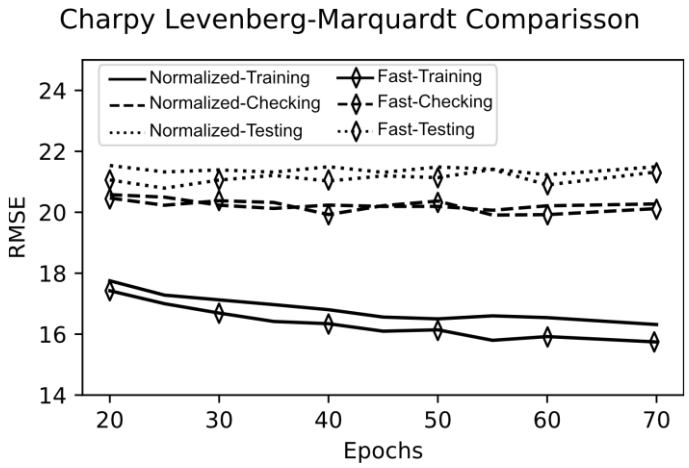


Figure 4.12 Charpy impact dataset, training, checking and testing performance for different number of epochs for the normalized and fast SICFIS models.

Figure 4.13 shows the required computation time for different number of epochs. It is observed a linear increase of computational time with the addition of epochs, although with different slopes. For the 210 epochs the training times were 12.12 and

6.61 seconds for the normalized and the fast SICFIS model respectively, roughly twice the computation time required. The difference between the RMSE is minimal and may be attributed to random effects. Further comparison of the performance between models will be presented in the results sections for the three real world datasets.

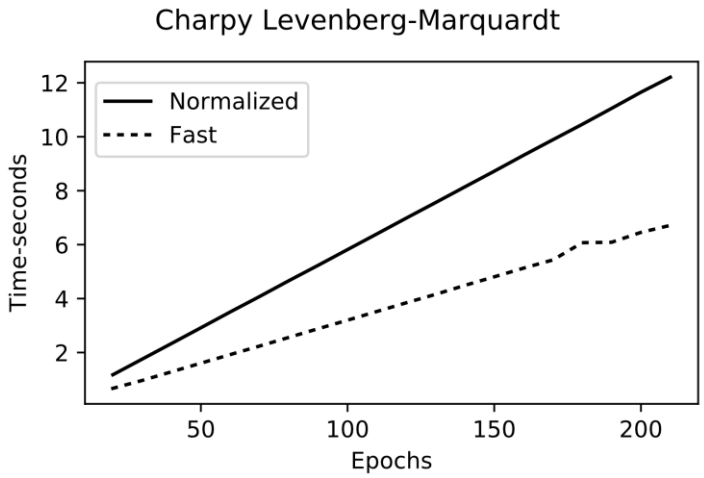


Figure 4.13 Charpy impact dataset, training times for the normalized and fast SICFIS models for different number of epochs.

4.7 Results

In Order to validate the generalization properties of the normalized-SICFIS and the fast-SICFIS introduced in this Chapter, the four real-world datasets presented in Chapter 3 are utilized. A parameter grid search is performed on each of the datasets, and the performance of each combination of parameters from the grid is recorded.

4.7.1 Charpy Impact Dataset Results

For the Charpy impact dataset the parameter grid is shown in Table 4.6. The RMSE is used to measure the performance of the models. A summary of the results of the normalized-SICFIS and the fast-SICFIS models are shown in Table 4.7 and Table 4.8

respectively. The best models obtained from both the normalized and the fast model are shown in Table 4.9. The regression plots of the best performing models for the fast and normalized-SICFIS models are shown in Figure 4.14 and Figure 4.15 respectively.

Table 4.6: Charpy impact dataset parameter grid.

Parameter	Values
Models	{Normalized-SICFIS, Fast-SICFIS}
Optimization Method	LM
Number of membership functions per feature	{2,3,4,5,6}
Initial LM coefficient	{20,40,60,80,100}
Number of k-fold cross validation per model	5
Maximum number of epochs	70
Training-Checking-Testing partitions	[65-18-17]

Table 4.7 Charpy Impact Normalized-SICFIS Results Summary.

No. mF*	Training		Checking		Testing		All	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
2mF	18.64	0.39	20.52	1.07	21.29	0.75	19.47	0.22
3mF	16.45	0.48	20.16	1.14	19.94	1.04	17.81	0.35
4mF	15.92	0.48	21.04	1.57	20.11	1.27	17.72	0.33
5mF	16.08	0.50	20.57	0.92	20.94	1.46	17.87	0.41
6mF	15.69	0.45	20.12	1.00	21.37	2.15	17.64	0.65

*mF: membership function

Table 4.8: Charpy Impact Fast-SICFIS Results Summary.

No. mF*	Training		Checking		Testing		All	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
2mF	16.96	0.54	21.28	1.80	20.49	1.37	18.46	0.41
3mF	16.22	0.44	21.25	1.78	20.57	1.32	18.03	0.33
4mF	16.30	0.44	22.05	2.39	20.83	1.53	18.31	0.66
5mF	15.70	0.51	20.66	1.60	20.46	1.48	17.58	0.56
6mF	15.70	0.28	21.57	1.50	20.65	1.15	17.80	0.42

*mF: membership function

For comparison purposes four additional models are shown in Table 4.10. The first model is a Mamdani FIS with singleton defuzzification, which is equivalent to a RBFN. It is a 9 rule FIS, the input partition is 56.25-18.75-25 for training, checking and testing respectively [25]. The second model is an ANFIS model with a quantum membership function [117]. It is a 6 rule FIS created utilizing a fuzzy c-means clustering algorithm and the input partition is 55-15-30 for training, checking and testing respectively [117]. The third and fourth model are a 6 and 8 rule Interval Type-2 TSK FISs (IT2-Squared)

respectively, as proposed in [40] for UTS predictions. The data partition is 60-20-20 for training, checking and testing respectively.

Table 4.9: Charpy Impact SICFIS Best Results.

No. mF ‡	Training		Checking		Testing		All	
	Norm*	Fast†	Norm*	Fast†	Norm*	Fast†	Norm*	Fast†
2mF	18.12	16.90	19.76	20.35	21.01	19.21	18.94	17.97
3mF	15.87	16.31	18.31	20.57	20.52	18.10	17.20	17.46
4mF	15.26	15.82	20.99	20.91	19.04	18.05	17.10	17.23
5mF	15.23	15.38	21.12	19.63	19.75	18.52	17.25	16.77
6mF	15.41	15.82	19.45	18.71	17.98	19.89	16.66	17.12

*Norm: Normalized-SICFIS model. †Fast: Fast-SICFIS model. ‡mF: Membership Function.

The differences between the normalized and the fast SICFIS performance are comparable, with a lower SD between the results of the normalized-SICFIS model, would mean the results are more consistent. The best results of both models are similar.

Table 4.10 Charpy Impact Results Comparison.

Model	Training	Checking	Testing	All
RBFN [25]	14.66	21.24	20.42	17.33
ANFIS [117]	17.75	18.84	18.17	18.03
IT2-Squared 6 rules	16.41	19.4	19.65	17.65
IT2-Squared 8 rules	15.74	20.73	19.83	17.55

In comparison with other models, the mean performance of both SICFIS models is comparable with the best models registered in Table 4.10. Demonstrating the superiority of the SICFIS model, in both performance and computation time.

The computation times measured in seconds of 9 different models are shown in Table 4.11. These results were obtained on a Windows 10 64 bit desktop computer with a processor Intel Core i5-4590 CPU @3.30 GHz and an installed memory RAM of 8.00 GB. The initial FISs for the RBFN and the ANFIS were obtained by utilizing the Subtractive Clustering algorithm from the MATLAB 2018b fuzzy toolbox, each model having been trained for 20 epochs. For optimization, the RBFN utilizes a

backpropagation algorithm and the ANFIS a hybrid backpropagation-least-squares algorithm.

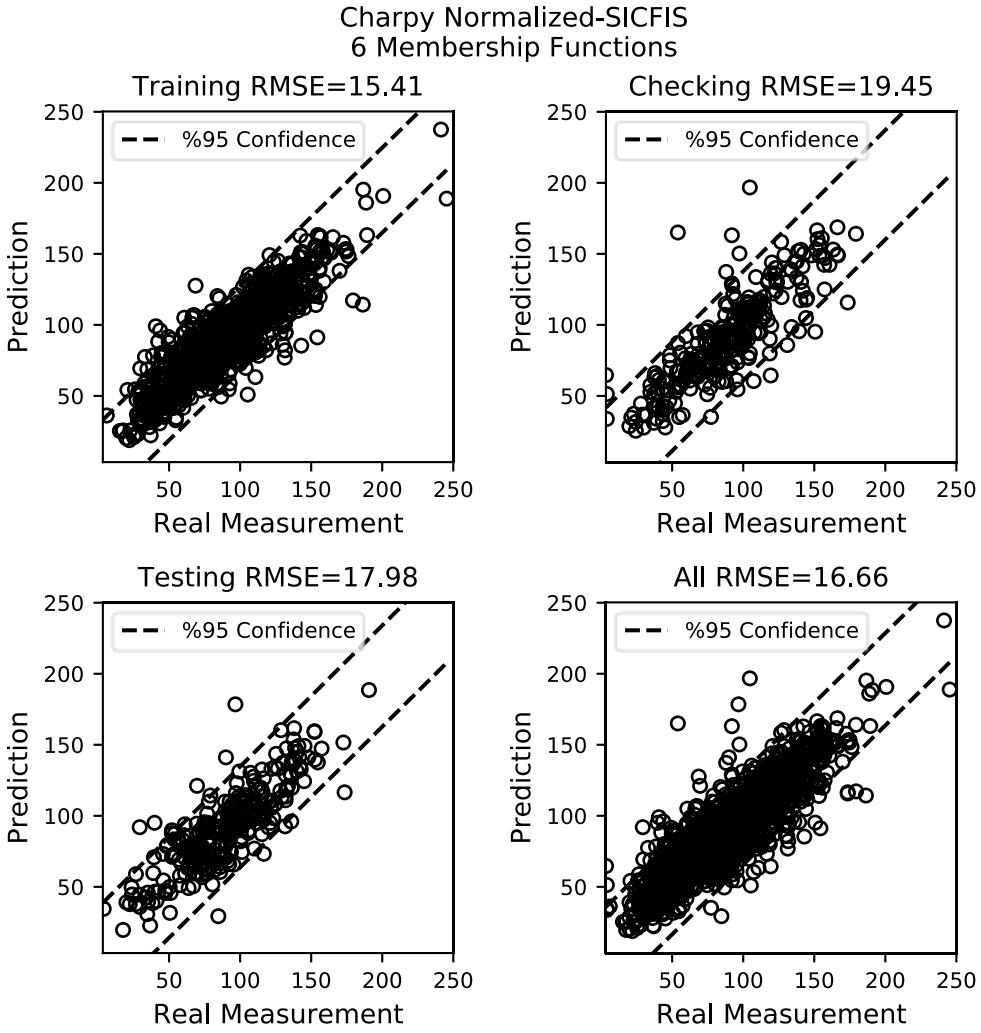


Figure 4.14: Charpy Impact test, results regression plot, normalized-SICFIS model with 6 membership functions partitions per feature.

Table 4.11: Charpy Impact, initial FIS and training computation times in seconds.

	RBFN 9 Rules	ANFIS 9 Rules	Normalized-SICFIS 2mF	Fast-SICFIS 2mF
Initial FIS	1.694s	1.728s	0.018s	0.017s
Training	1.44s	4.727s	2.52	0.134s
	RBFN 10 Rules	ANFIS 10 Rules	Normalized-SICFIS 3mF	Fast-SICFIS 3mF
Initial FIS	1.772s	1.772s	0.017s	0.017s
Training	1.618s	5.508s	2.53	0.18s
	RBFN 11 Rules	ANFIS 11 Rules	Normalized-SICFIS 4mF	Fast-SICFIS 4mF
Initial FIS	1.847s	1.824s	0.019s	0.016s
Training	1.759s	6.554s	2.52	0.238s

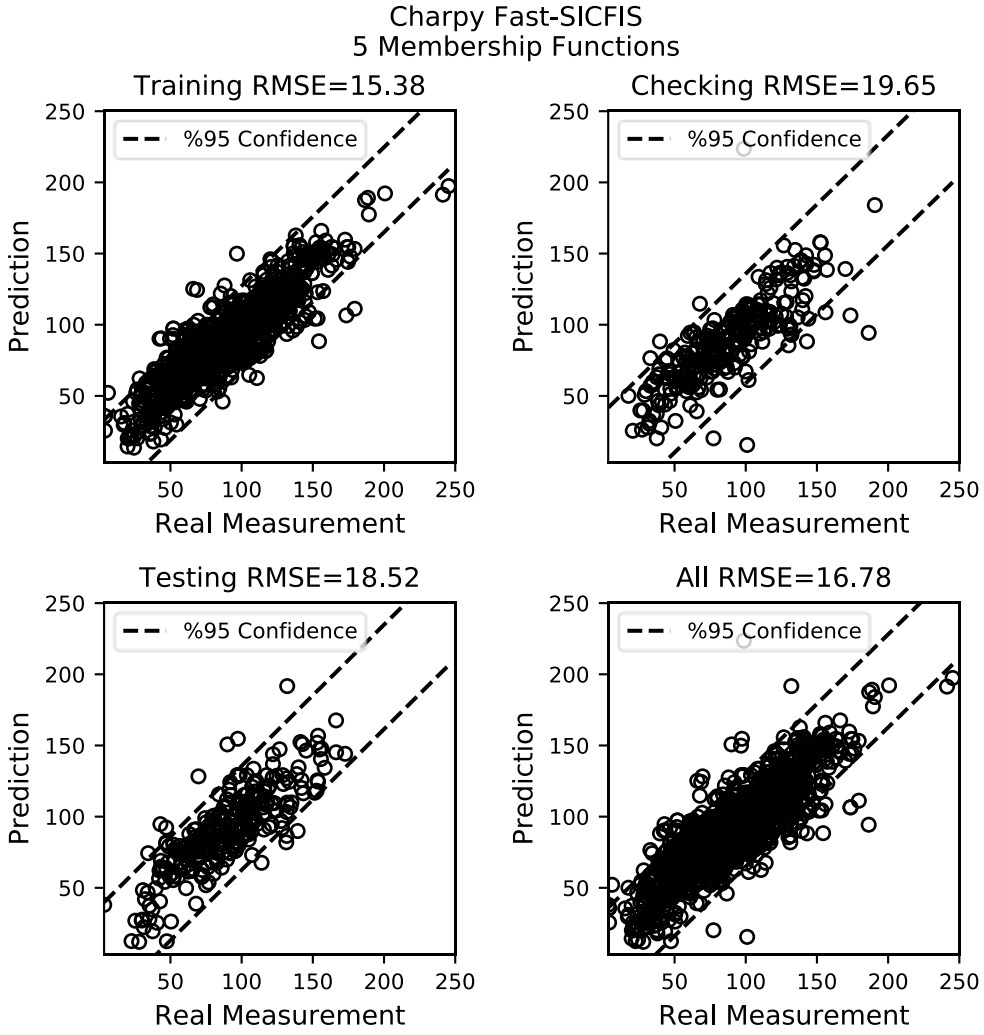


Figure 4.15: Charpy Impact test, results regression plot, fast-SICFIS model with 5 membership functions partitions per feature.

Table 4.12: Charpy Impact inference computation time in seconds.

	RBFN 9 Rules	ANFIS 9 Rules	Normalized-SICFIS 2mF	Fast-SICFIS 2mF
Inference	0.1094s	0.1563s	0.01001s	0.0020s
	RBFN 10 Rules	ANFIS 10 Rules	Normalized-SICFIS 3mF	Fast-SICFIS 3mF
Inference	0.2188s	0.2231s	0.01003s	0.0025s
	RBFN 11 Rules	ANFIS 11 Rules	Normalized-SICFIS 4mF	Fast-SICFIS 4mF
Inference	0.2344s	0.2434s	0.01004s	0.0030s

4.7.2 Ultimate Tensile Strength Results

The dataset includes two categorical features with 3 and 6 categories, a membership function per category will be used for these features. The data partition is 70-30 for training and testing respectively, the validation consists of 12 data points. The parameter grid is shown in Table 4.13: UTS parameter grid.. The UTS results summary containing the mean and SD results from the normalized and fast SICFIS model are shown in Table 4.14 and Table 4.15 respectively.

Table 4.13: UTS parameter grid.

Parameter	Values
Models	{Normalized-SICFIS, Fast-SICFIS}
Optimization Method	LM
Number of membership functions per feature	{3,4,5,6,7,8}
Initial LM coefficient	{20,40,60,80,100}
Number of k-fold cross validation per model	5
Maximum number of epochs	70
Training-Testing partitions	[70-30]

Table 4.14: UTS normalized-SICFIS UTS results summary.

No. mF*	Training		Testing		Validation		All	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
3mF	36.03	0.90	38.81	1.54	54.46	6.66	36.97	0.75
4mF	37.51	1.93	41.72	2.24	63.36	4.39	38.93	1.91
5mF	36.58	1.69	41.29	1.58	65.73	8.87	38.19	1.35
6mF	36.69	1.84	42.24	2.29	64.04	7.34	38.56	1.83
7mF	37.75	3.53	44.65	4.29	79.14	18.41	40.16	3.62
8mF	36.55	1.83	43.33	2.70	62.78	7.04	38.82	1.83

*mF: membership function,

The best models obtained from both the normalized and the fast model are shown in Table 4.16. The regression plots of the best performing models for the fast and normalized-SICFIS models are shown in Figure 4.16 and Figure 4.17 and respectively.

Table 4.15: UTS fast-SICFIS UTS results summary.

No. mF*	Training		Testing		Validation		All	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
3mF	37.24	1.02	40.19	1.42	59.67	5.03	38.25	0.75
4mF	35.78	0.93	41.12	3.77	57.90	4.43	37.58	1.46
5mF	37.31	1.83	45.49	4.18	65.62	4.80	40.07	2.37
6mF	37.75	1.97	46.65	3.99	67.19	5.00	40.77	2.24
7mF	37.16	1.59	45.04	3.97	69.25	7.43	39.85	2.04
8mF	34.80	1.85	48.54	6.60	60.81	6.30	39.61	2.76

*mF: membership function,

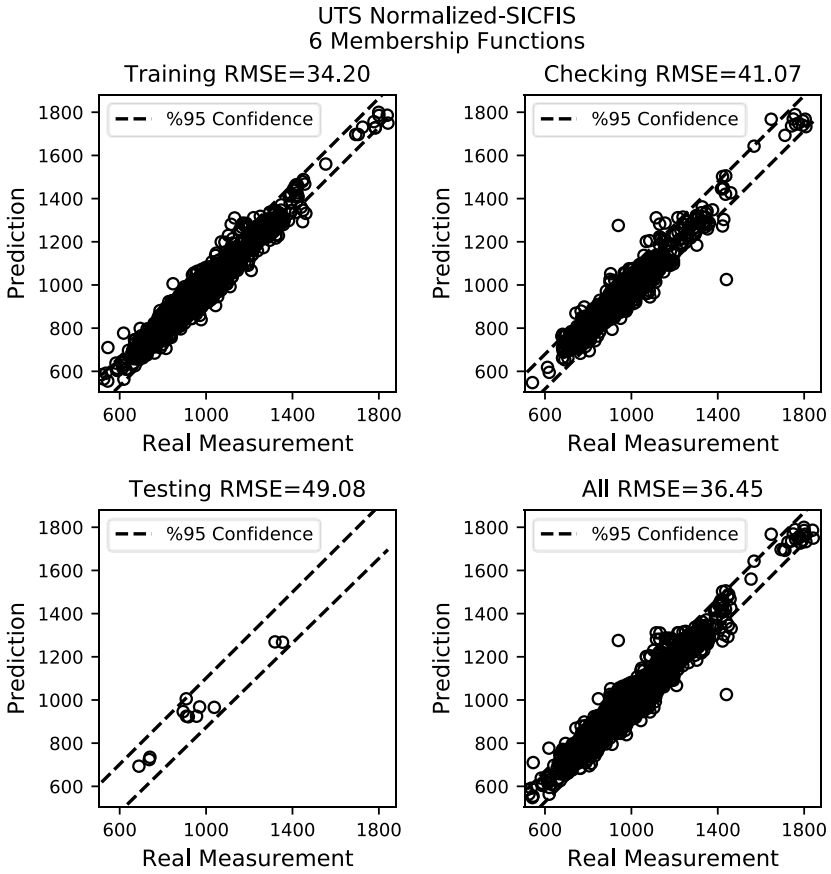


Figure 4.16: UTS test, results regression plot, normalized-SICFIS model with 6 membership functions partitions per feature.

Table 4.16: UTS Normalized and Fast SICFIS UTS Best Results.

No. mF	Training		Testing		Validation		All	
	Norm*	Fast†	Norm*	Fast†	Norm*	Fast†	Norm*	Fast†
3mF	35.36	35.13	41.07	36.93	49.80	51.96	37.22	35.74
4mF	35.64	33.69	36.25	39.26	55.05	52.30	35.91	35.52
5mF	35.22	33.14	39.65	40.43	59.91	50.19	36.70	35.54
6mF	35.97	34.71	42.40	38.19	63.76	53.86	38.12	35.86
7mF	34.20	33.88	41.07	39.26	49.08	68.87	36.45	35.74
8mF	32.23	34.49	41.15	38.89	57.25	65.87	35.24	36.00

*Norm: Normalized-SICFIS model. †Fast: Fast-SICFIS model. ‡mF: Membership Function.

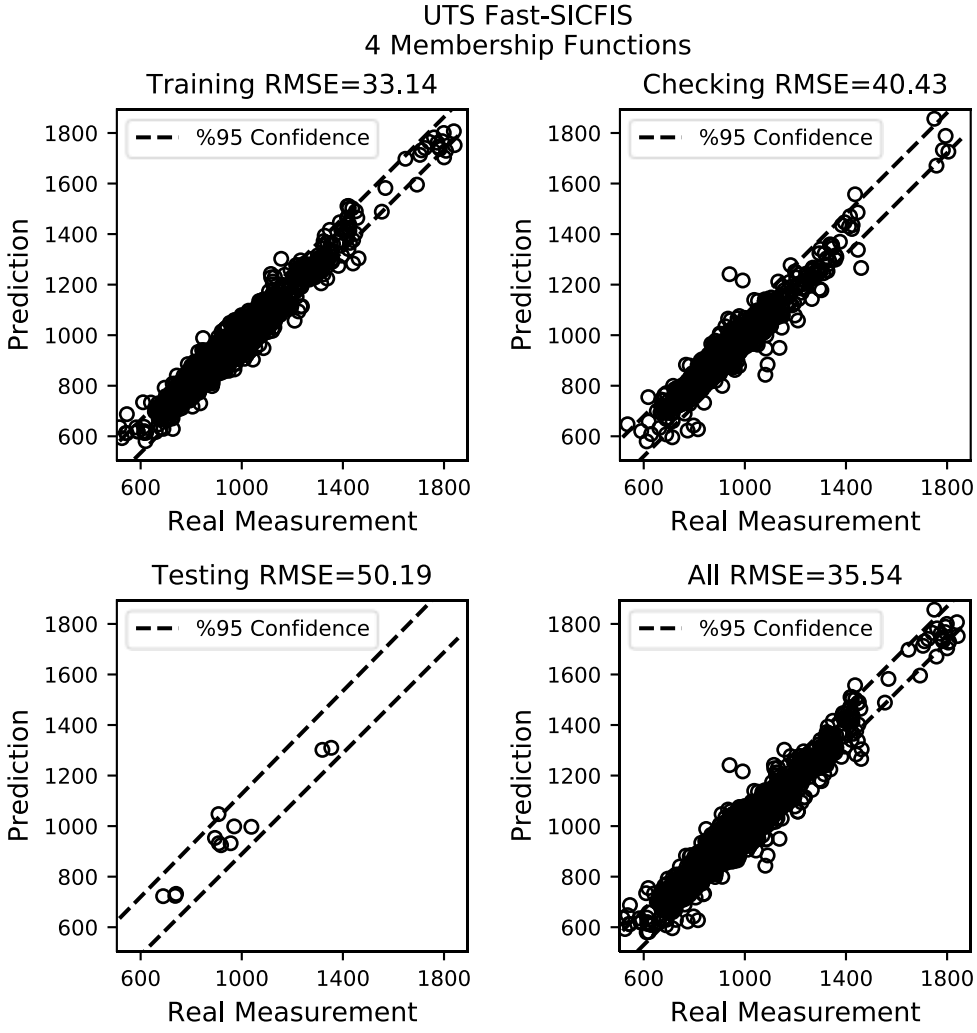


Figure 4.17: UTS test, results regression plot, fast-SICFIS model with 5 membership functions partitions per feature.

For comparison purposes the results of three different FISs is shown, the IT2-Squared and the Multi- Objective Interval Type-2 Fuzzy Modelling (MOIT2FM) [118] are type-2 FISs, the IMOFM-M [118] is a Mamdani type-1 FIS, all are composed of 6 rules. The RMSE is used to measure the performance and results are shown in Table 4.17. The results are mixed, as the normalized and fast SICFIS model outperform the training partition, the testing partition performance is equivalent to the IT2-Squared and MOIT2FM, while the validation partition underperforms in comparison.

Table 4.17: UTS results comparison.

Model	Training	Testing	Validation
IT2-Squared [40]	34.45	38.76	37.34
MOIT2FM [118]	36.33	40.52	34.77
IMOFM-M [118]	46.47	45.52	49.87

4.7.3 Bladder Cancer Results

The Bladder cancer dataset includes mostly categorical features. Three continuous features contain integer values and therefore will be considered as being categorical in this study. Therefore, only one feature is treated as being continuous. The Area Under the Curve (AUC) is used to measure performance to compare with other models. The AUC is calculated with the same dataset as the one in this work, that is the resulting dataset of the records of the non-censored patients and the records of the right-censored patients whose last observed time surpasses the 60-month threshold.

The Bladder cancer parameter grid is shown in Table 4.18. A summary of the results obtained by the normalized and fast SICFIS models are shown in Table 4.19 and Table 4.20 respectively. The best results obtained are shown in Table 4.21.

For comparison 5 other models are shown in Table 4.22, these models being: The Cox regression model, a logistic regression model (LoR), an ANN and two FISs. The

FISs shown in Table 4.22 have been integrated into the Cox regression model in order to perform a risk prognosis analysis. The first is a type-1 FIS with 20 Fuzzy Mamdani type rules and the second a type-2 FIS also composed of 20 Fuzzy Mamdani type rules. Further information regarding these models can be found in [39].

Table 4.18: Bladder Cancer Parameter Grid.

Parameter	Values
Models	{Normalized-SICFIS, Fast-SICFIS}
Optimization Method	LM
Number of membership functions per feature	{2,3,4}
Initial LM coefficient	{20,40,60,80,100}
Number of k-fold cross validation per model	5
Maximum number of epochs	70
Training -Testing partitions	[65-35]

Table 4.19 Normalized-SICFIS Bladder Cancer Results Summary.

No. mF*	Training		Testing		All	
	Mean	SD	Mean	SD	Mean	SD
2mF	0.9022	0.0076	0.8726	0.0084	0.8918	0.0057
3mF	0.9027	0.0064	0.8710	0.0081	0.8915	0.0041
4mF	0.9037	0.0060	0.8725	0.0084	0.8928	0.0036

*mF: membership function

Table 4.20: Fast-SICFIS Bladder Cancer Results Summary.

No. mF*	Training		Testing		All	
	Mean	SD	Mean	SD	Mean	SD
2mF	0.9057	0.0084	0.8747	0.0087	0.8952	0.0047
3mF	0.9065	0.0054	0.8763	0.0132	0.8963	0.0030
4mF	0.9046	0.0090	0.8751	0.0145	0.8945	0.0069

*mF: membership function.

The Receiver Operating Characteristic (ROC) curves of the best results for the normalized and fast SICFIS models are shown in Figure 4.18 and in Figure 4.20 respectively. The corresponding scatter plots of the Scores are shown in Figure 4.19 and in Figure 4.21 respectively. The optimum point is selected as the point in which the

prediction accuracy it's at its maximum. The confusion matrix corresponding to such optimum point is shown in Table 4.23 and in Table 4.24.

Table 4.21:Normalized and Fast SICFIS Bladder Cancer Best Results.

No. mF‡	Training		Testing		All	
	Norm*	Fast†	Norm*	Fast†	Norm*	Fast†
2mF	0.9115	0.9190	0.8824	0.8652	0.9005	0.9001
3mF	0.9119	0.9022	0.8795	0.8985	0.8971	0.9011
4mF	0.9060	0.9015	0.8852	0.8998	0.8976	0.9010

*Norm: Normalized-SICFIS model. †Fast: Fast-SICFIS model. ‡mF: Membership Function.

Table 4.22: Bladder Cancer Results Comparison.

Model	Training	Testing	All
Cox [39]	0.83	0.82	0.83
LoR [39]	0.76	0.74	0.75
ANN [39]	0.88	0.84	0.87
T1 FIS [39]	0.88	0.83	0.86
T2 FIS [39]	0.92	0.91	0.92

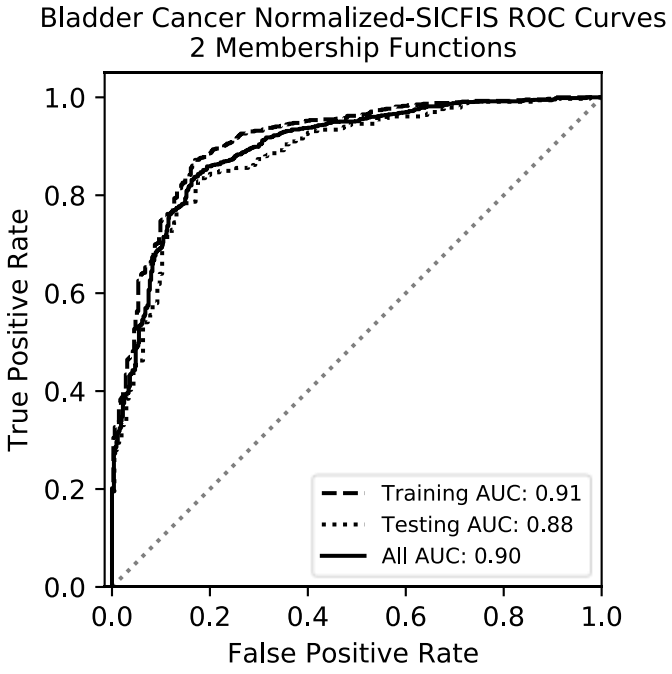


Figure 4.18: Normalized-SICFIS 2 membership functions ROC curves.

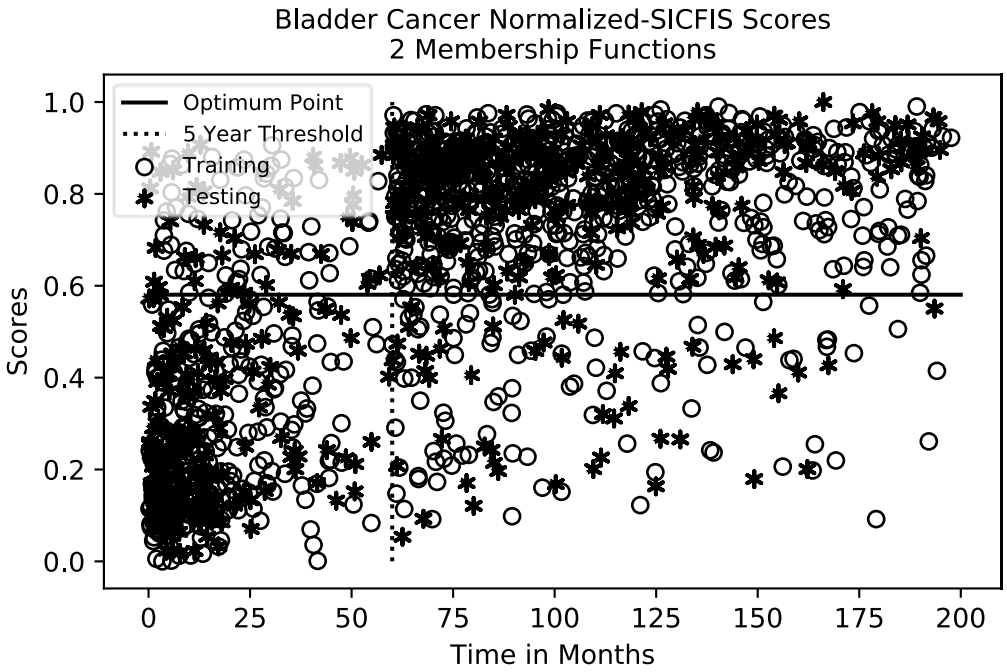


Figure 4.19: Normalized-SICFIS 2 membership functions scores scatter plot.

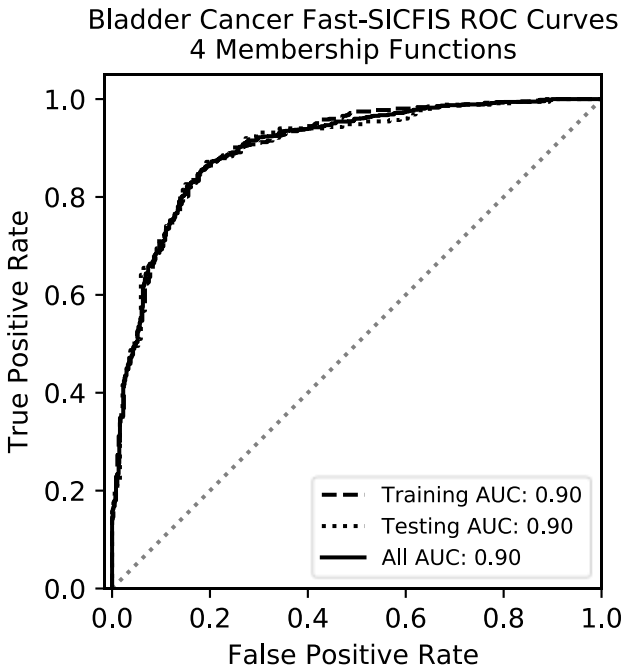


Figure 4.20: Fast-SICFIS 4 membership functions ROC curves.

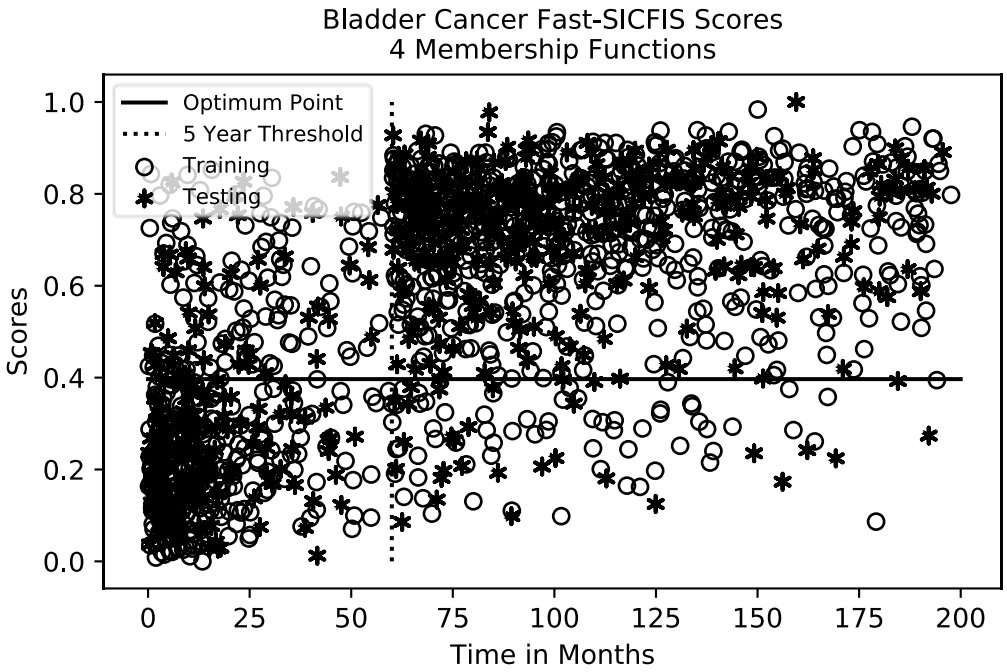


Figure 4.21: Fast-SICFIS 4 membership functions scores scatter Plot.

Table 4.23: Normalized-SICFIS 2 membership functions Confusion Matrix.

		True Class			
		Low Risk		High Risk	
		Training	Testing	Training	Testing
Predicted Class	Low Risk	272	149	94	44
	High Risk	62	51	599	310

Table 4.24: Fast-SICFIS Confusion 4 membership functions Matrix.

		True Class			
		Low Risk		High Risk	
		Training	Testing	Training	Testing
Predicted Class	Low Risk	298	162	67	32
	High Risk	79	47	583	313

4.7.4 Superconductivity Results

A summary of the results obtained from the superconductivity data set are shown in Table 4.25 and Table 4.26. The best results obtained given a number of membership functions per feature is shown in Table 4.27. A results comparison is shown in Table 4.28, five different models are shown: a linear regression model, an XG-Boost model, an ANFIS model and two ANNs. Both the linear regression and XG-Boost results are obtained from [109]. The training partitions for the linear regression and the XG-Boost model is 2/3 for training and 1/3 for testing, the reported results are only for the out-of-sample data and no information is available for the remaining partitions. The data partition for the ANFIS model and the two ANN is 65-18-17 for training, checking and testing respectively. The ANFIS model is composed of 8 rules, while the two ANN are composed of 10 and 20 hidden layers.

Table 4.25: Superconductivity Normalized-SICFIS Results Summary.

No. mF*	Training		Checking		Testing		All	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
2mF	14.72	0.128	14.90	0.315	15.02	0.150	14.80	0.067
3mF	14.22	0.116	14.65	0.396	14.54	0.247	14.35	0.113
4mF	13.79	0.157	14.50	0.381	14.34	0.286	14.02	0.106

*mF: membership function

Table 4.26: Superconductivity Fast-SICFIS Results Summary.

No. mF*	Training		Checking		Testing		All	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
2mF	13.99	0.137	14.46	0.358	14.61	0.185	14.18	0.133
3mF	13.80	0.088	14.40	0.286	14.57	0.182	14.04	0.081
4mF	13.45	0.103	14.55	0.633	14.47	0.246	13.83	0.149

*mF: membership function

Table 4.27: Superconductivity SICFIS Best Results.

No. mF ‡	Training		Checking		Testing		All	
	Norm*	Fast†	Norm*	Fast†	Norm*	Fast†	Norm*	Fast†
2mF	14.76	13.79	14.55	13.92	14.81	14.37	14.73	13.91
3mF	14.21	13.60	14.07	14.86	14.34	14.15	14.21	13.93
4mF	13.45	13.19	14.25	14.05	14.46	14.65	13.77	13.61

*Norm: Normalized-SICFIS model. †Fast: Fast-SICFIS model. ‡mF: Membership Function.

Table 4.28 Superconductivity Results Comparison.

Model	Training	Checking	Testing	All
Linear Regression [109]	NA	NA	17.6	NA
XG-Boost [109]	NA	NA	9.4	NA
ANFIS 8 Rules	13.37	16.27	16.08	14.42
ANN 10 hidden layers	13.42	13.50	14.23	13.58
ANN 20 hidden layers	12.54	13.39	12.93	12.76

4.8 Interpretability Analysis: Example of the Charpy Impact Dataset

The interactions of processes and alloying elements and their effect on the material properties are complex and are often difficult to represent. For the purpose of this analysis, the magnitude-phase plots of a selected number of features is shown in Figure 4.23. Because the Charpy impact dataset is known for the scattered measurements this diagram is obtained from a SICFIS model trained with the complete dataset. For validation, the information in [119] will be utilized. This information contains a comprehensive summary of the effect of alloying elements to notch toughness.

A scatter plot of the results is shown in Figure 4.22. The plot shows the whole complex number coordinates. It is shown that most of the predictions are located within the second and third quadrants.

As already stated, the Charpy impact test measures the notch toughness of a material and characterizes the DBTT. The impact temperature is an important variable in the model, and it is known that at low temperatures the material becomes brittle and at high

temperatures the material becomes ductile. Carbon is the main alloying element in steel, of which a high concentration of carbon causes the material to become brittle and therefore an increase of carbon in steel is associated with a decrease in impact energy [119].

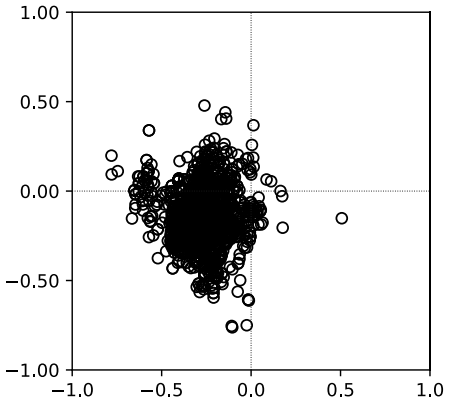


Figure 4.22: Two-dimensional magnitude and phase scatter plot of results.

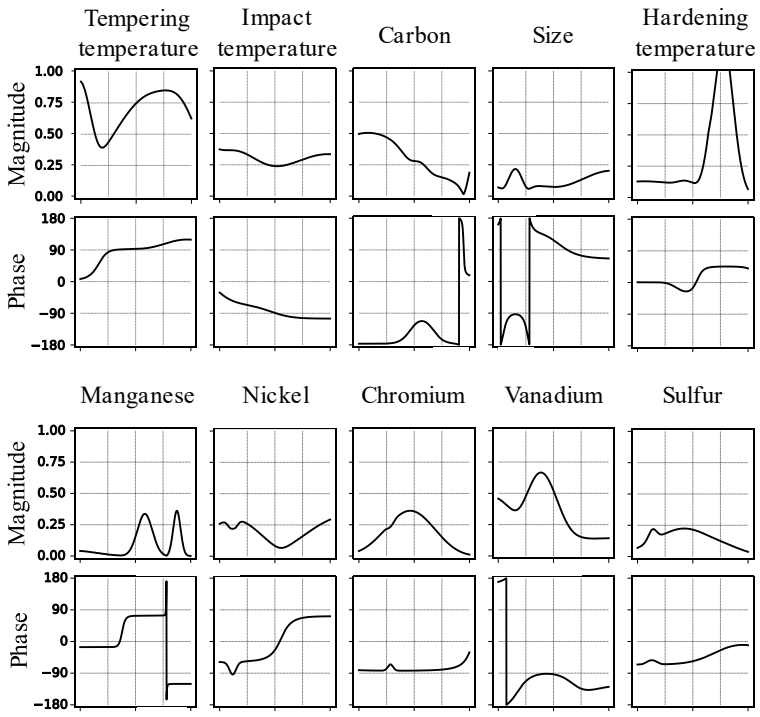


Figure 4.23: Charpy impact test magnitude-phase plots.

Given the known effect of both impact temperature and Carbon, it is possible in general to associate a positive effect on impact energy to angles within the second and third quadrant, and a negative effect to angles within the first and fourth quadrant. There are exceptions to this however, and this would depend mostly on the interaction with other alloying elements and the process [119].

Increasing the Manganese content reduces the transition temperature and improves the upper shelf energy in low carbon steel. A lesser effect is observed in medium carbon steel and has little effect on high carbon steels. Manganese can have the opposite effect on tempered and hardened steel. In the magnitude and phase plot of these alloying elements it is observed that a high content of Manganese is detrimental to high carbon steel, high hardening temperatures and to tempering, while being beneficial to some extent to low carbon steel [119].

Nickel is used to improve the materials properties at low temperatures but is also known to have a negative effect on the upper shelf energy while Chromium is known to increase the upper shelf energy. It is shown that a high content of nickel has a 180° phase difference with a high impact temperature, hence creating a negative interference, and it remains mostly orthogonal with a low impact temperature. Chromium's phase, however, would produce a positive interference with a high impact temperature and is orthogonal to a low impact temperature, which means its effect is mostly on the upper shelf energy [119].

Vanadium improves notch toughness [120], while the addition of Sulphur has a negative effect in notch toughness [119]. This can be emphasized by the fact that Sulphur is located within the fourth quadrant and vanadium in the second and third quadrants.

4.9 Summary

To the authors' best knowledge, the SICFIS model is the first interpretable CFIS based on CFSs. It was demonstrated that the SICFIS model performs equivalently to other well-known models with as little as 2 partitions per feature. Computational times are reduced exponentially due to its simple structure and the application of GPU parallel computing.

The results obtained from the Charpy impact test are superior to other FISs. SICFIS was shown to be transparent and interpretable. The interpretability analysis performed on the magnitude-phase plots is consistent with what is currently known in the literature. Given the single input-partition-per-rule architecture of SICFIS it is possible to determine the individual effect of each alloying element and process. Moreover, eliciting an initial SICFIS is approximately 100 times faster than traditional FISs utilizing a subtracting Clustering algorithm. The training time is 10 and 30 times faster compared with the RBFN and the ANFIS models respectively. The fast-SICFIS model can improve the computation times even further with a more computational efficient architecture and the power of parallel computing.

The results obtained from the UTS dataset for the training and testing partition produce equivalent performance to the other FIS methods, for the 12 validation points the results perform are sub-optimal, and more work is required to improve upon the results.

The results obtained from the Bladder cancer prediction were superior to the other models, excluding the type-2 FIS. It should be mentioned even better results may have been obtained by modifying the model to perform a proper survival analysis, which is out of the scope of this work. The fact that it demonstrated a superior performance compared with state-of-the-art models shows promise of utilizing SICFIS for other

medical applications. For this dataset, the normalized-SICFIS model performed considerably worse than the fast-SICFIS model, this differences in performance are attributed to the negative influence of the rule normalization operation when a large number of categorical features are present.

The results obtained from the superconductivity dataset are comparable with the ANN and ANFIS models. Demonstrating the capabilities of the normalized and fast SICFIS models to perform predictions with large datasets.

The normalized and the fast SICFIS models provide similar results. The slight reduction in the standard deviation obtained from the result summary may indicate a more consistent performance from the normalized-SICFIS model. The fast-SICFIS model can train models around two times faster than the normalized SICFIS model, this reduction in computational time may become more significant for larger datasets. Therefore, the trade-off between computational speed and consistent results should be taken into consideration depending on the application, such as in real-time applications, it would be of great benefit a considerable reduction in computational times. For datasets with a large number of categorical features the fast-SICFIS model would in theory be a better choice as demonstrated by the results obtained by the Cancer dataset.

In addition to the superior performance obtained from both SICFIS models it was demonstrated the interpretability and transparency of the models. Among the different knowledge representation methods it can be argued that the magnitude-phase plots provide crucial information for the validation of the model and the extraction of knowledge, moreover, its interpretability is not affected when there is overlap between partitions or when the number of partitions increases as it may be the case with the vector partition plots and the cosine distance matrix plot.

In comparison with type-1 and type-2 FIS, the SICIFS model provides better insight of the individual effects of a feature in the overall performance of a model. Additionally, the SICFIS rule-base can be represented utilizing the traditional type-1 fuzzy rule-base, with an additional control over the granularity of the information presented.

Chapter 5

The Adaptive Neuro Fuzzy Inference System with Single Input Complex Fuzzy Inference System Consequences

5.1 Introduction and Background

The TSK FIS is a rule-base model whose premises are composed of linguistic variables and the consequences are composed of functions, which are most commonly linear regression models. Each rule represents a region of the dataset that can be approximated by a local linear model, this divide-and-conquer strategy allows to model complex nonlinear systems as a combination of interpretable linear models. Defining fuzzy boundaries allows for a better representation of the local model and improves the prediction accuracy for data points located within the boundaries of two or more local regions. Dividing a large and complex problem into local interpretable models may become a problem as the complexity increases. The larger and the more complex a dataset is the more rules are required to model its behaviour, hence decreasing its interpretability.

In order to improve the prediction accuracy of TSK models and reduce the number of rules some authors have devised different adaptations to the TSK architecture. Models such as the *neural networks designed on approximate reasoning architecture* [121], and *the co-active neuro fuzzy inference system* [122] embed ANNs to the TSK FIS architecture with the objective of combining the interpretability of FISs and the prediction accuracy of ANNs. Embedding ANN into FIS reduces considerably, if not

at all, its interpretability as ANN are black-box models. These models are not to be confused with the popular ANFIS model [76], the ANFIS is a TSK FIS and does not embed ANN to its architecture but rather utilizes backpropagation learning algorithms to improve its accuracy while maintaining its interpretability.

Other strategies developed to reduce the number of rules and improve upon the accuracy of the results while maintaining the transparency and interpretability of the system include replacing the consequences of a TSK FIS with nonlinear functions. The number of rules is reduced considerably given that the overall architecture of the system is local-nonlinear which can describe a larger region of the dataset more accurately compared with linear models. These methods have been applied for control applications. Rajesh [123] include sinusoidal functions to improve accuracy of a controller. Sala and Ariño [124] utilize polynomials from Taylor series expansion. Tanaka [125] utilizes a sum of squares for modelling non-linear dynamical systems. Dong [126] utilizes local nonlinear TSK rules for the design of a controller.

In this work it is proposed to replace the linear consequence of the TSK with the fast-SICFIS model. In Chapter 4 the interpretability properties of the SICFIS was demonstrated, its superior accuracy compared with other models, and the considerable reduction in training times, especially in the case of the fast-SICFIS model were also shown. These properties make it an ideal candidate for improving upon the accuracy of the ANFIS model while retaining its interpretability. The Results obtained are comparable with ensembles of ANN. Training times are comparably lower than other more complex methods, while maintaining its interpretability.

5.2 The ANFIS-SICFIS Model

The ANFIS-SICFIS model is a neuro fuzzy inference system based on the popular ANFIS architecture. The ANFIS-SICFIS premise is composed of a traditional type-1

rule-base and the consequences are composed of SICFIS models. The ANFIS-SICFIS fuzzy rule-base is given in Table 5.1.

Table 5.1: ANFIS-SICFIS Rule-base.

Premise	Consequence
$IF\ x_1\ is\ A_1^1\ AND\ x_2\ is\ A_2^1 \cdots AND\ x_P\ is\ A_P^1\ THEN\ y = is\ h^1(\mathbf{x})$	
$IF\ x_1\ is\ A_1^2\ AND\ x_2\ is\ A_2^2 \cdots AND\ x_P\ is\ A_P^2\ THEN\ y = is\ h^2(\mathbf{x})$	
$\vdots$	
$IF\ x_1\ is\ A_1^R\ AND\ x_2\ is\ A_3^R \cdots AND\ x_P\ is\ A_P^R\ THEN\ y = is\ h^R(\mathbf{x})$	

where x_p represents the input value for a feature p A_p^r represents a type-1 fuzzy membership function for a rule r and a feature p and h^r represents the output of a local SICFIS model corresponding to the rule output.

5.2.1 ANFIS-SICFIS Premises

The premise of the ANFIS-SICFIS can be represented as a three layered system, the first layer fuzzifies the input utilizing a Gaussian membership function (5.1), the second layer calculates the rule strength utilizing the product t-norm (5.2), finally the third layer normalizes the fired rule strength (5.3).

$$O_{r,p}^1 = \mu_{r,p} = \exp\left(-\frac{1}{2}\left(\frac{x_p - c_{r,p}^{RB}}{\sigma_{r,p}^{RB}}\right)^2\right) \tag{5.1}$$

$$O_r^2 = w_r = \prod_{p=1}^{P} \mu_{r,p} \tag{5.2}$$

$$O_r^3 = \bar{w}_r = \frac{w_r}{\sum_{r=1}^{R} w_r} \tag{5.3}$$

The premises of the rules correspond to a region of the dataset. The rules may be defined by an expert or by utilizing a clustering algorithm. The clustering algorithm allows to identify the associations between the inputs and the output in the dataset [127]. The most common fuzzy clustering algorithm and the one utilized in this work is the Fuzzy C-Means (FCM) algorithm [20]. The FCM algorithm is as follows:

Algorithm 5.1 Fuzzy C-Means clustering algorithm

Inputs: Dataset *x*, fuzzy partition exponent $m>1$ end condition threshold η, maximum number of epochs *q*, number of centers *C*.
Output: fuzzy partition matrix *u*, fuzzy cluster center positions *v*

Assign initial values for prototypes $c_1, c_2, ..., c_C$
While $t < q$ or $|J_t - J_{t-1}| < \eta$

Calculate fuzzy partition matrix $u_{ij} \leftarrow \sum_{k=1}^{C} \left(\frac{\left\| x_i - c_j^{FCM} \right\|}{\left\| x_i - c_k^{FCM} \right\|} \right)^{-\frac{2}{m-1}}$

Update prototypes $c_j^{FCM} \leftarrow \sum_{i=1}^{N} u_{ij}^m x_i \Big/ \sum_{i=1}^{N} u_{ij}^m$

compute objective function $J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ic}^m \left\| x_i - c_j^{FCM} \right\|^2$

where $m>1$ is the fuzzy partition exponent, c^{FCM} are the initial values for the prototypes and *N* is the total number of instances in the dataset. From the FCM clustering algorithm it is possible to create a rule-base utilizing the *c* prototypes and the fuzzy partition matrix *u*. The centers of the Gaussian membership functions for the rule-base $c_{r,p}^{RB}$ are equal to the projections of the prototypes c^{FCM} of the FCM algorithm. The calculation of the spreads $\sigma_{r,p}^{RB}$ are calculated utilizing the fuzzy covariance matrix [27] as follows:

$$Cov_i = \frac{\sum_{j=1}^{N} (u_{ij})^m \left(\mathbf{a}^k - \mathbf{v}_i \right) \left(\mathbf{a}^k - \mathbf{v}_i \right)^T}{\sum_{j=1}^{n} (u_{ij})^m} \tag{5.4}$$

$$\sigma_{r,p}^{RB} = [Diag(Cov_i)]^{1/2} \tag{5.5}$$

The fuzzy partition exponent determines the "fuzziness" of the clustering algorithm. It can be shown that when *m=1* the FCM algorithms produces "hard" partitions of the dataset [128]. The degree of fuzziness or overlap between partitions can be measured utilizing the partition coefficient shown in (5.6) [20]. The partition coefficient approaches 1 as the partition become "harder". Similarly, a partition coefficient of the rule-base can be measured utilizing normalized rule strength $\bar{w}_r$ (5.3), instead of the fuzzy partition matrix *u* as shown in (5.7).

$$Partition\,Coefficient(\text{FCM}) = \sum_{i=1}^{N}\sum_{j=1}^{C}\left(u_{ij}\right)^2 \Big/ N \tag{5.6}$$

$$Partition\,Coefficient(\text{Rule-Base}) = \sum_{i=1}^{N}\sum_{r=1}^{R}\left(\bar{w}_{i,r}\right)^2 \Big/ N \tag{5.7}|$$

where *N* is the total number of instances *C* is the number of clusters and R is the number of rules. Figure 5.1 shows the partition coefficient value as the fuzzy partition exponent *m* increases for both the FCM and the rule-base. A sharp decline in the FCM partition coefficient is observed as *m* increases with lesser effect in the rule-base partition coefficient. Therefore, it can be inferred that to obtain distinguishable local interpretable models it is important to choose a partition coefficient value between 1 and 2.

5.2.2 ANFIS-SICFIS Consequences

The fast-SICFIS model already explored in Chapter 4 is a 4 layered complex-FIS. The SICFIS present several advantages over traditional FIS and other machine-learning models, these advantages include its interpretability, its low complexity and fast computation. In order to differentiate between the premise membership functions of the ANFIS-SICFIS rule-base and the SICFIS the symbol μ will be used for the former and

the symbol ν for the latter. The SICFIS will be represented as a nonlinear function, the architecture of the model was presented in Chapter 4.6 and the equations are summarized below:

$$h^r(\mathbf{x}) = \sqrt{\left(g^r_{\mathrm{Re}}\right)^2 + \left(g^r_{\mathrm{Im}}\right)^2} \tag{5.8}$$

$$g^r_{\mathrm{Re}} = \sum_{p=1}^{P}\sum_{s=1}^{S_p} \nu^r_{p,s_p} \cdot \cos(\varphi^r_{p,s_p}) \cdot \beta^r_{p,s_p} \tag{5.9}$$

$$g^r_{\mathrm{Im}} = \sum_{p=1}^{P}\sum_{s=1}^{S_p} \nu^r_{p,s_p} \cdot \sin(\varphi^r_{p,s_p}) \cdot \beta^r_{p,s_p} \tag{5.10}$$

$$\nu^r_{p,s_p} = \exp\left(-\frac{1}{2}\left(\frac{x_p - c^{CFR,r}_{p,s_p}}{\sigma^{CFR,r}_{p,s_p}}\right)^2\right) \tag{5.11}$$

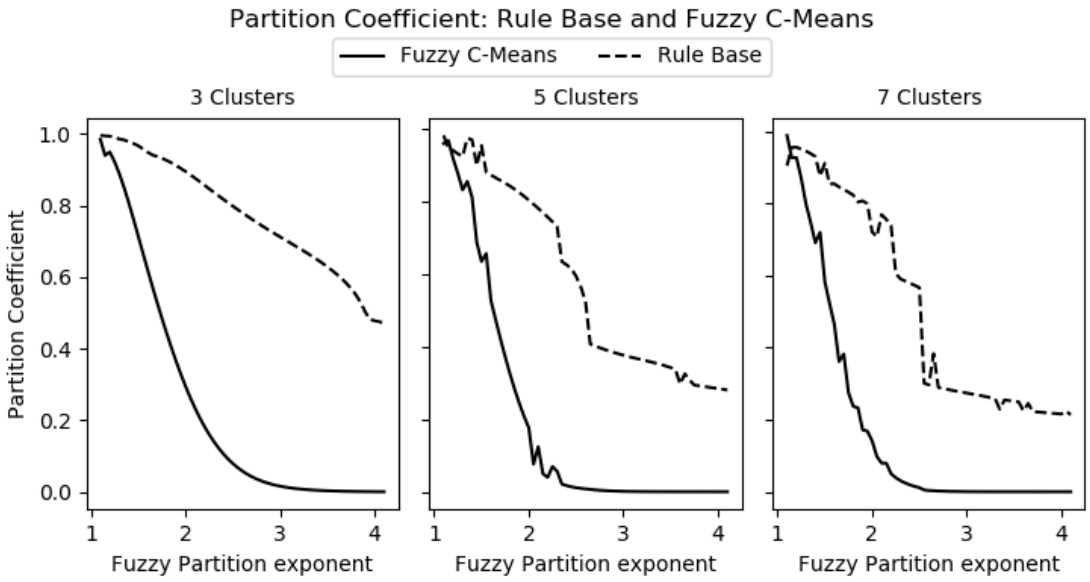


Figure 5.1 Fuzzy partition coefficient values given different clusters and changing the fuzzy partition exponent value.

The output of the SICFIS model is a complex number, with a phase and a magnitude, something referred to as the "dual output property"; in Chapter 4 the magnitude of the SICFIS model was used to asses its performance and the phase was utilized as additional information utilized during the interpretability analysis. To adequately address the dual output property of the SICFIS within the context of the ANFIS-SICFIS model, two different approaches will be explored. The first one passes the output of the SICFIS as real-valued, that means, only the magnitude information of the output. The second approach passes the output of the SICFIS as complex-valued.

The first approach is relatively straightforward, as the last layers of the ANFIS-SICFIS simply perform an algebraic product between the normalized rule strength and the real-valued consequent later to sum the outputs of each rule, the output of this model is real-valued, therefore from here after this approach will be defined as the *real-ANFIS-SICFIS* model. The second approach would require and additional layer, a second rule interference layer, which would calculate interference between the rules, the final output of this model is complex-valued, therefore the magnitude is utilized to assess its performance and the phase can be used as additional information, from here after this approach will be defined as the *complex-ANFIS-SICFIS* model.

5.2.3 Real-ANFIS-SICFIS

The first three layers of the real-ANFIS-SICFIS represent the premise rule-base of the system described by the equations (5.1)-(5.3). The fourth layer of the real-ANFIS-SICFIS is the magnitude of the local SICFIS for a rule *r* as follows:

$$O_r^{\text{Real},4} = h^r(\mathbf{x}_i) = \sqrt{\left(g_{\text{Re}}^r\right)^2 + \left(g_{\text{Im}}^r\right)^2} \tag{5.12}$$

The final output aggregates the inference between the premises and the consequences of each rule as follows:

$$O^{\text{Real},5} = \sum_{r=1}^{R} \bar{w}_r \cdot h^r \tag{5.13}$$

A schematic of the real-ANFIS-SICFIS is shown in Figure 5.2. The architecture of this model resembles closely the one of Jang's ANFIS model [76].

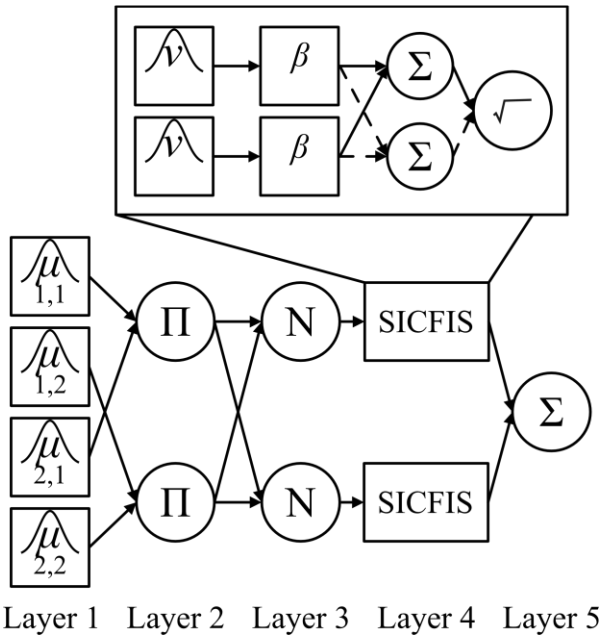


Figure 5.2: The real-ANFIS-SICFIS schematic.

5.2.3.1 Real-ANFIS-SICFIS Training

For the optimization the LM optimization algorithm explored in section 4.5 is utilized. To differentiate between the premise and the consequences parameters a

superscript with the symbol μ is used for the premise parameters of the type-1 fuzzy rule-base. A superscript v_r is used for the parameters of the SICFIS model, corresponding to a rule (local model) *r*.

Premise parameters:

$$\left[\sigma_{r,p}^{\mu} \quad c_{r,p}^{\mu}\right] \tag{5.14}$$

Consequence parameters:

$$\left[\beta_{p,s_p}^{r} \quad \varphi_{p,s_p}^{r} \quad \sigma_{p,s_p}^{v_r} \quad c_{p,s_p}^{v_r}\right] \tag{5.15}$$

Derivatives:

$$\frac{\partial f}{\partial \sigma_{r,p}^{\mu}} = \left(\frac{\partial f}{\partial \mu_r}\frac{\partial \mu_r}{\partial \sigma_{r,p}^{\mu}}\right) \tag{5.16}$$

$$\frac{\partial f}{\partial c_{r,p}^{\mu}} = \left(\frac{\partial f}{\partial \mu_r}\frac{\partial \mu_r}{\partial c_{r,p}^{\mu}}\right) \tag{5.17}$$

$$\frac{\partial f}{\partial \beta_{p,s_p}^{r}} = \frac{\partial f}{\partial h^r}\left(\frac{\partial h^r}{\partial g_{\mathrm{Re}}^r}\frac{\partial g_{\mathrm{Re}}^r}{\partial \beta_{p,s_p}^{r}} + \frac{\partial h^r}{\partial g_{\mathrm{Im}}^r}\frac{\partial g_{\mathrm{Im}}^r}{\partial \beta_{p,s_p}^{r}}\right) \tag{5.18}$$

$$\frac{\partial f}{\partial \varphi_{p,s_p}^{r}} = \frac{\partial f}{\partial h^r}\left(\frac{\partial h^r}{\partial g_{\mathrm{Re}}^r}\frac{\partial g_{\mathrm{Re}}^r}{\partial \varphi_{p,s_p}^{r}} + \frac{\partial h^r}{\partial g_{\mathrm{Im}}^r}\frac{\partial g_{\mathrm{Im}}^r}{\partial \varphi_{p,s_p}^{r}}\right) \tag{5.19}$$

$$\frac{\partial f}{\partial \sigma_{p,s_p}^{v_r}} = \frac{\partial f}{\partial h^r}\left(\frac{\partial h^r}{\partial g_{\mathrm{Re}}^r}\frac{\partial g_{\mathrm{Re}}^r}{\partial v_{p,s_p}^{r}}\frac{\partial v_{p,s_p}^{r}}{\partial \sigma_{p,s_p}^{v_r}} + \frac{\partial h^r}{\partial g_{\mathrm{Im}}^r}\frac{\partial g_{\mathrm{Im}}^r}{\partial v_{p,s_p}^{r}}\frac{\partial v_{p,s_p}^{r}}{\partial \sigma_{p,s_p}^{v_r}}\right) \tag{5.20}$$

$$\frac{\partial f}{\partial c_{p,s_p}^{v_r}} = \frac{\partial f}{\partial h^r}\left(\frac{\partial h^r}{\partial g_{\mathrm{Re}}^r}\frac{\partial g_{\mathrm{Re}}^r}{\partial v_{p,s_p}^{r}}\frac{\partial v_{p,s_p}^{r}}{\partial c_{p,s_p}^{v_r}} + \frac{\partial h^r}{\partial g_{\mathrm{Im}}^r}\frac{\partial g_{\mathrm{Im}}^r}{\partial v_{p,s_p}^{r}}\frac{\partial v_{p,s_p}^{r}}{\partial c_{p,s_p}^{v_r}}\right) \tag{5.21}$$

5.2.4 Complex-ANFIS-SICFIS

A schematic of the complex-ANFIS-SICFIS model is shown in Figure 5.3. The first three layers represent the premise rule-base of the system described in the equations

(5.1)-(5.3). The output of the fourth layer of the complex-ANFIS-SICFIS utilizes the real and the imaginary output of the SICFIS for a rule *r* as follows:

$$O_{r,\mathrm{Re}}^{\mathrm{Complex},4} = g_{\mathrm{Re}}^{r} = \sum_{p=1}^{P}\sum_{s=1}^{S_p} \nu_{p,s_p}^{r} \cdot \cos(\varphi_{p,s_p}^{r}) \cdot \beta_{p,s_p}^{r} \tag{5.22}$$

$$O_{r,\mathrm{Im}}^{\mathrm{Complex},4} = g_{\mathrm{Im}}^{r} = \sum_{p=1}^{P}\sum_{s=1}^{S_p} \nu_{p,s_p}^{r} \cdot \sin(\varphi_{p,s_p}^{r}) \cdot \beta_{p,s_p}^{r} \tag{5.23}$$

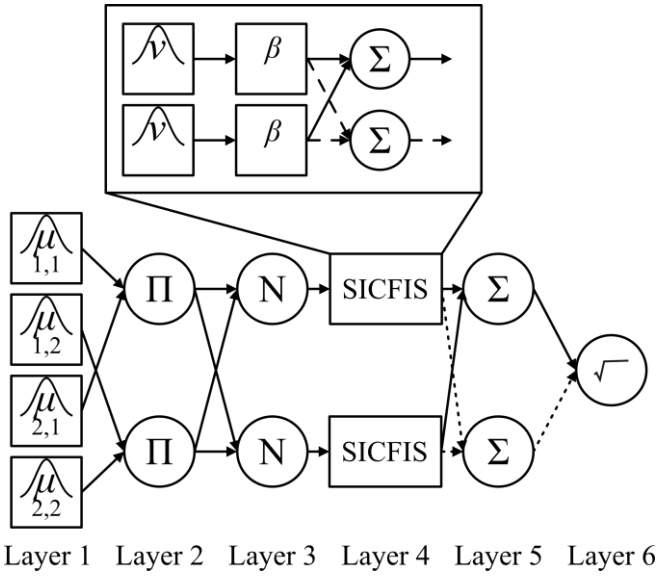


Figure 5.3: The complex-ANFIS-SICFIS schematic.

Given that the output of the fourth layer is a complex number, the complex-ANFIS-SICFIS includes an additional layer, which measures the interference between the rules, additionally, each rule consequent is multiplied by the normalized rule strength. The output of the fifth layer is also a complex quantity with a real and an imaginary part as follows:

$$O_{\mathrm{Re}}^{\mathrm{Complex},5} = \sum_{r=1}^{R} \overline{w}^{r} \cdot g_{\mathrm{Re}}^{r} \tag{5.24}$$

$$O_{\mathrm{Im}}^{\mathrm{Complex},5} = \sum_{r=1}^{R} \overline{w}^{r} \cdot g_{\mathrm{Im}}^{r} \tag{5.25}$$

The output of the sixth layer calculates the magnitude and the phase of the real and imaginary quantities obtained from the output of the fifth layer. The magnitude is utilized to make the predictions and measure the performance of the system while the phase is utilized for additional information.

$$O^{\text{Complex},6} = \left|O^{\text{Complex},5}\right| \angle \arg\left(O^{\text{Complex},5}\right) \tag{5.26}$$

5.2.4.1 Complex-ANFIS-SICFIS Training

For the optimization the LM optimization algorithm explored in section 4.5 is utilized. To differentiate between the premise and the consequences parameters a superscript with the symbol μ is used for the premise parameters of the type-1 fuzzy rule-base. A superscript v_r is used for the parameters of the SICFIS model, corresponding to a rule local model *r*.

Premise parameters:

$$\left[\sigma_{r,p}^{\mu} \quad c_{r,p}^{\mu}\right] \tag{5.27}$$

Consequence parameters:

$$\left[\beta_{p,s_p}^{r} \quad \varphi_{p,s_p}^{r} \quad \sigma_{p,s_p}^{v_r} \quad c_{p,s_p}^{v_r}\right] \tag{5.28}$$

Derivatives:

$$\frac{\partial f}{\partial \sigma_{r,p}^{\mu}} = \left(\frac{\partial f}{\partial h_{\text{Re}}}\frac{\partial h_{\text{Re}}}{\partial \mu_r}\frac{\partial \mu_r}{\partial \sigma_{r,p}^{\mu}} + \frac{\partial f}{\partial h_{\text{Im}}}\frac{\partial h_{\text{Im}}}{\partial \mu_r}\frac{\partial \mu_r}{\partial \sigma_{r,p}^{\mu}}\right) \tag{5.29}$$

$$\frac{\partial f}{\partial c_{r,p}^{\mu}} = \left(\frac{\partial f}{\partial h_{\text{Re}}}\frac{\partial h_{\text{Re}}}{\partial \mu_r}\frac{\partial \mu_r}{\partial c_{r,p}^{\mu}} + \frac{\partial f}{\partial h_{\text{Im}}}\frac{\partial h_{\text{Im}}}{\partial \mu_r}\frac{\partial \mu_r}{\partial c_{r,p}^{\mu}}\right); \tag{5.30}$$

$$\frac{\partial f}{\partial \beta_{p,s_p}^{r}} = \left(\frac{\partial f}{\partial h_{\text{Re}}}\frac{\partial h_{\text{Re}}}{\partial g_{\text{Re}}^{r}}\frac{\partial g_{\text{Re}}^{r}}{\partial \beta_{p,s_p}^{r}} + \frac{\partial f}{\partial h_{\text{Im}}}\frac{\partial h_{\text{Im}}}{\partial g_{\text{Im}}^{r}}\frac{\partial g_{\text{Im}}^{r}}{\partial \beta_{p,s_p}^{r}}\right) \tag{5.31}$$

$$\frac{\partial f}{\partial \varphi^{r}_{p,s_p}} = \left(\frac{\partial f}{\partial h_{\mathrm{Re}}} \frac{\partial h_{\mathrm{Re}}}{\partial g^{r}_{\mathrm{Re}}} \frac{\partial g^{r}_{\mathrm{Re}}}{\partial \varphi^{r}_{p,s_p}} + \frac{\partial f}{\partial h_{\mathrm{Im}}} \frac{\partial h_{\mathrm{Im}}}{\partial g^{r}_{\mathrm{Im}}} \frac{\partial g^{r}_{\mathrm{Im}}}{\partial \varphi^{r}_{p,s_p}} \right) \tag{5.32}$$

$$\frac{\partial f}{\partial \sigma^{v_r}_{p,s_p}} = \left(\frac{\partial f}{\partial h_{\mathrm{Re}}} \frac{\partial h_{\mathrm{Re}}}{\partial g^{r}_{\mathrm{Re}}} \frac{\partial g^{r}_{\mathrm{Re}}}{\partial v^{r}_{p,s_p}} \frac{\partial v^{r}_{p,s_p}}{\partial \sigma^{v_r}_{p,s_p}} + \frac{\partial f}{\partial h_{\mathrm{Im}}} \frac{\partial h_{\mathrm{Im}}}{\partial g^{r}_{\mathrm{Im}}} \frac{\partial g^{r}_{\mathrm{Im}}}{\partial v^{r}_{p,s_p}} \frac{\partial v^{r}_{p,s_p}}{\partial \sigma^{v_r}_{p,s_p}} \right) \tag{5.33}$$

$$\frac{\partial f}{\partial c^{v_r}_{p,s_p}} = \left(\frac{\partial f}{\partial h_{\mathrm{Re}}} \frac{\partial h_{\mathrm{Re}}}{\partial g^{r}_{\mathrm{Re}}} \frac{\partial g^{r}_{\mathrm{Re}}}{\partial v^{r}_{p,s_p}} \frac{\partial v^{r}_{p,s_p}}{\partial c^{v_r}_{p,s_p}} + \frac{\partial f}{\partial h_{\mathrm{Im}}} \frac{\partial h_{\mathrm{Im}}}{\partial g^{r}_{\mathrm{Im}}} \frac{\partial g^{r}_{\mathrm{Im}}}{\partial v^{r}_{p,s_p}} \frac{\partial v^{r}_{p,s_p}}{\partial c^{v_r}_{p,s_p}} \right) \tag{5.34}$$

5.3 Model Evaluation

The objective of the ANFIS-SICFIS model is to create a partition of the dataset in order to obtain a global model composed of local interpretable non-linear models. To maintain interpretability each rule should model accurately a local region of the dataset. Therefore, to assess the performance and interpretability of the system, a local performance will be taken into consideration, this measurement will not be included in the objective function, rather it will be used to assess the final performance of the system.

The local performance is assessed as follows: Each one of the instances in a dataset is evaluated utilizing the trained real and complex ANFIS-SICFIS models. Instead of utilizing the prediction of the global model, a local SICFIS model will be selected to assess its performance, the local SICFIS model is selected according to the normalized rule strength values. The rule with the highest normalized rule strength corresponds to the local SICFIS utilized in the evaluation of that record. This is repeated for each data point, the results are collected and the RMSE is calculated for the training, checking and testing partitions. Both ANFIS-SICFIS models utilize the same evaluation method shown in Algorithm 5.2.

During the training of the ANFIS-SICFIS models, the rule-base may be altered, this may affect the local performance of a model. To asses these alterations three different optimization strategies are to be implemented. The first one will optimize all the parameters at the same time, we would define this as the complete parameter optimization process. The second one would optimize the premise parameters and the SICFIS parameters separately, each one at different epochs, this method is defined as the alternate parameter optimization process. The third method would optimize solely the SICFIS parameters, we would define this method as the consequent parameter optimization process.

Algorithm 5.2: Local Performance Evaluation

Inputs: Parameter vector: normalized rule firing strength $\bar{\mathbf{w}}(\mathbf{x}_j)=[\bar{w}_1(\mathbf{x}_j)\cdots\bar{w}_R(\mathbf{x}_j)]$ obtained from (5.3). Vector containing the local SICFIS magnitude output for the rules $\mathbf{h}(\mathbf{x}_j)=[h^1(\mathbf{x}_j)\cdots h^R(\mathbf{x}_j)]$ obtained from (5.8). Number of records N. outputs Y

Outputs: RMSE pertaining to the local performance

$j \leftarrow 1$

while $j < N$

$\bar{\mathbf{w}}(\mathbf{x}_j) \leftarrow [\bar{w}_1(\mathbf{x}_j)\cdots\bar{w}_R(\mathbf{x}_j)]$

$k_j \leftarrow \arg\max(\bar{\mathbf{w}}(\mathbf{x}_j))$

$\hat{y}_{Local,j} \leftarrow h^k(\mathbf{x}_j)$

$$RMSE_{Local}=\sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(\hat{y}_{Local,j}-y_j\right)^2}$$

It is expected that the consequent optimization algorithm would yield the best local performance given that the premises of the rule-base would remain unaltered. Additionally, it is expected that an initial rule-base created with a fuzzy partition coefficient closer to one would improve upon the local performance. To assess these hypotheses a parameter grid search will be performed with the parameters observed in Table 5.2. This exhaustive grid search is implemented to the Charpy impact dataset, resulting in the training of 1,440 models.

A summary of the results from the exhaustive grid search can be observed in the four graphs shown in Figure 5.4 and Figure 5.5. The four graphs correspond to the Global and Local performances of the real and complex -ANFIS-SICFIS, showing the mean RMSE of the models with 2, 3 and 4 rules utilizing each of the three different optimization strategies, Complete, Consequents and Alternate. The results include the training, checking and testing partition performance displaying the corresponding proportion of influence to the final error, the total length of these bars represent the complete RMSE. Any performance registering a deviance of more than two standard deviations is treated as an outlier and removed.

Table 5.2: Parameter grid search.

Parameter	Values
Models	{Real-ANFIS-SICFIS, Complex-ANFIS-SICFIS}
Optimization Method	{Complete, Consequents, Alternate}
Number of rules	{2,3,4}
Number of membership functions per feature (SICFIS)	{2,3,4,5}
Fuzzy partition coefficient values	{1.1,1.35,1.85,2.10}
Number of k-fold cross validation per model	5
Maximum number of epochs	50
Training-Checking-Testing partition	[65-18-17]

No correlation between the fuzzy partition coefficients, this may be very well explained from the graphs in Figure 5.1 as there is not a major change in the partition coefficient of the rules base for values of *m* between 1 and 2.

It is shown in the graphs below that the worse performing optimization strategy is the alternate parameter optimization process. With just a minor difference between the complete and the consequent optimization results. The complex ANFIS-SICFIS models yielded better results for the local models. In Figure 5.6 the training times for each of the optimization strategies is shown. It can be observed that the slowest algorithm is the complete parameter optimization process as it's training time grows exponentially with the addition of rules and membership functions.

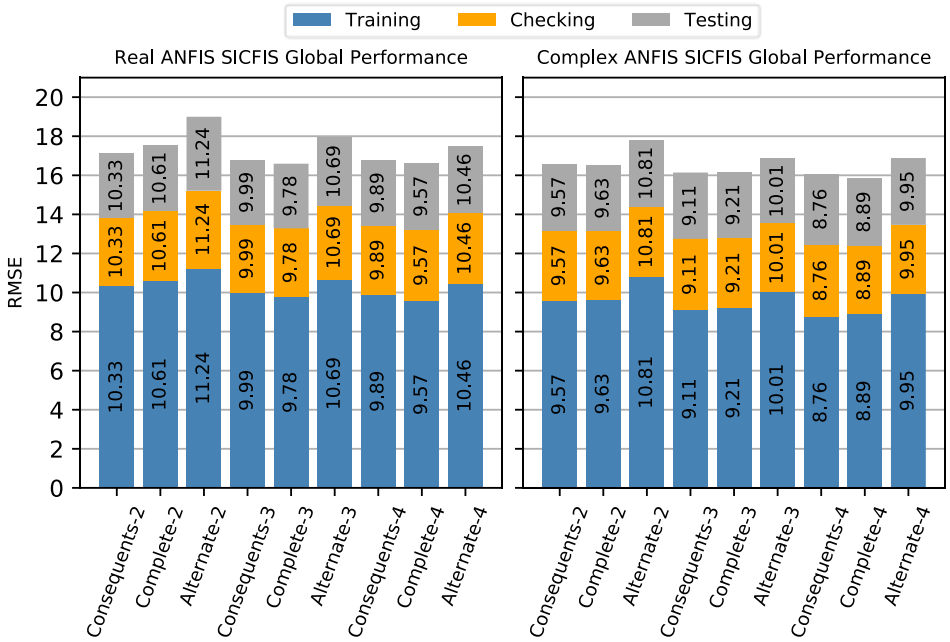


Figure 5.4: Real and Complex ANFIS-SICFIS global performance for the three optimization process given 2,3 and 4 rules. Stacked bar chart.

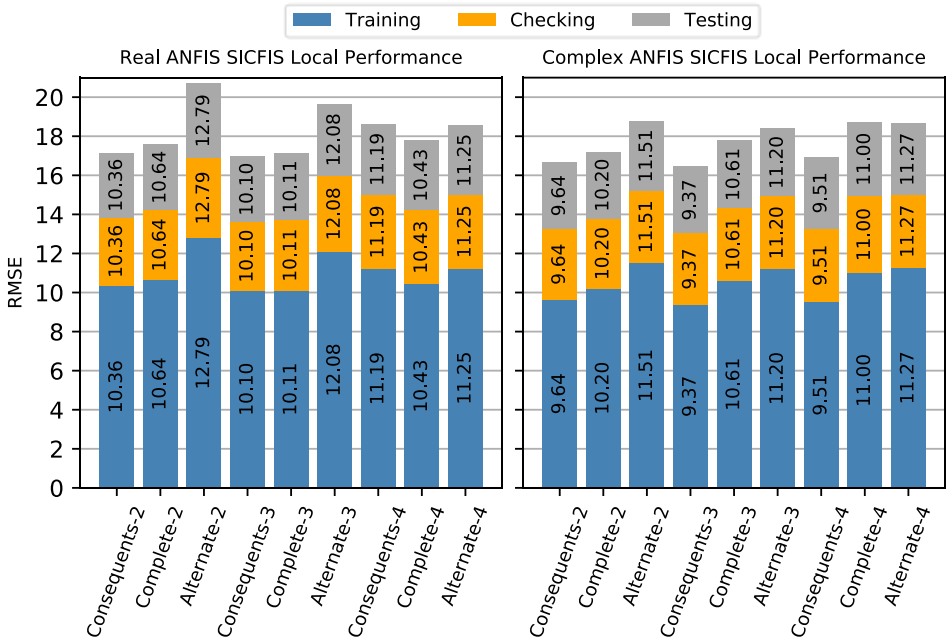


Figure 5.5: Real and Complex ANFIS-SICFIS local performance for the three optimization process given 2,3 and 4 rules. Stacked bar chart.

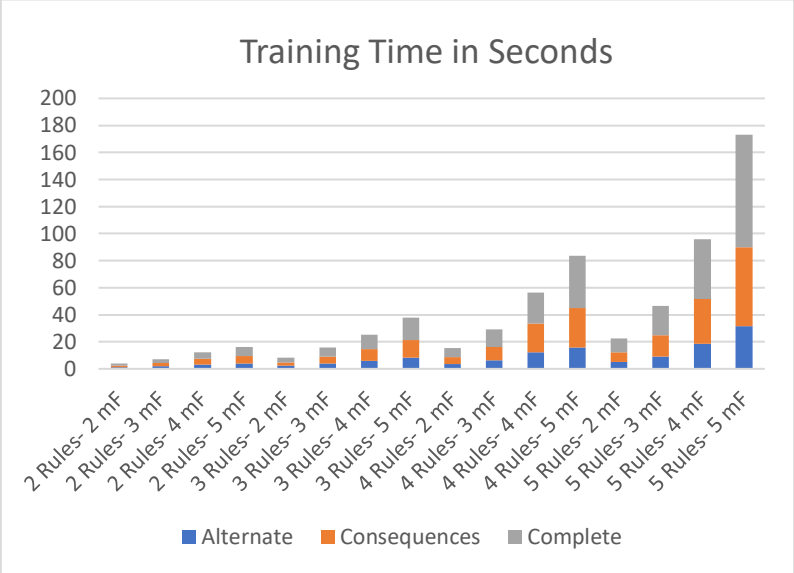


Figure 5.6 Training times for the complex-ANFIS-SICFIS model utilizing the alternate, consequent and complete parameter optimization method with a varying number of rules and membership functions (mF). Overlapping bar chart.

The fastest and worse performing optimization process is the alternate optimization algorithm. Therefore, the consequences optimization algorithm offers the best trade-off between Local-Global performance and training times.

It is concluded that the best results are obtained utilizing the complex-ANFIS-SICFIS model, and the consequent optimization process. Therefore in the following section and simulations it is the model selected to obtain the results.

5.3.1 Charpy Impact Test Results

A parameter grid search was performed on the Charpy impact test in the previous section to determine the performance of the two different ANFIS-SICFIS models and various optimization methods. A more detailed analysis based on the previous results

obtained is performed, the details of the new grid search are shown in Table 5.3, the training, checking and testing partition remains 65-18-17 respectively.

Table 5.3: Parameter grid search for the Charpy impact test.

Parameter	Values
Models	{Complex-ANFIS-SICFIS}
Optimization Method	{Consequents}
Number of rules	{2,3,4,5,6}
Number of membership functions per feature (SICFIS)	{2,3,4}
Fuzzy partition coefficient values	{1.2,1.8}
Number of k-fold cross validation per model	5
Maximum number of epochs	70

The mean results and the corresponding standard deviation given a number of rules are shown in Table 5.4 and Table 5.5 respectively. The mean RMSE for the training decreases with the addition of rules, while the checking and testing mean RMSE increases. The effect is greater for the local performance. The sharp increase in the standard deviation and mean RMSE given 6 rules indicates overfitting.

Table 5.4: Charpy Mean RMSE results given different number of rules.

	Training		Checking		Testing		All	
No. Rules	Global	Local	Global	Local	Global	Local	Global	Local
2	15.78	15.81	19.21	19.32	19.93	19.94	17.22	17.27
3	15.19	15.37	18.83	19.21	19.19	19.46	16.67	16.90
4	14.68	16.19	20.13	21.44	20.32	21.39	16.87	18.24
5	14.53	15.43	19.69	20.62	19.98	20.97	16.62	17.54
6	14.58	20.02	19.29	24.15	19.27	24.41	16.42	21.68

Table 5.5: Charpy Standard deviation results given different number of rules.

	Training		Checking		Testing		All	
No. Rules	Global	Local	Global	Local	Global	Local	Global	Local
2	0.947	0.976	1.421	1.377	1.303	1.317	0.650	0.674
3	1.268	1.100	1.385	1.386	1.419	1.530	0.604	0.479
4	0.946	1.645	1.755	2.191	1.771	2.260	0.478	1.378
5	1.341	1.396	1.799	1.890	1.584	1.925	0.893	1.071
6	1.100	10.738	1.398	9.815	1.430	9.907	0.658	10.288

Similar results are observed in Figure 5.7, where the addition of membership function results in a decreasing RMSE for the training partition and an increasing RMSE for the testing partition.

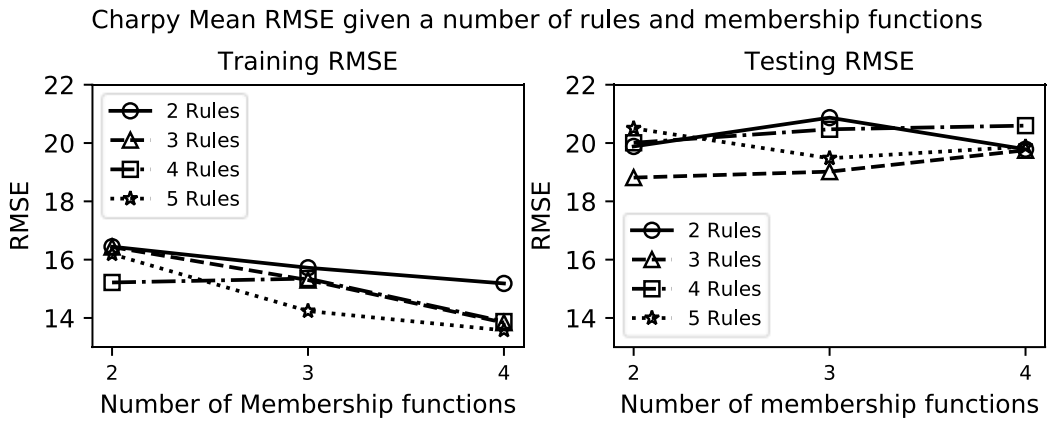


Figure 5.7: effect of membership functions to performance.

The best results given different number of rules are shown in Table 5.6, with the corresponding number of membership function per feature. For comparison purposes the results obtained from different studies utilizing ANN are shown in Table 5.7. The first one is an Ensemble-NN [129], the second one is an ANN model whose hyperparameters are selected with a GA [129], the third one is an GA-NN Ensemble [129], which optimize the hyperparameters as well as the ensemble structure. The best out-of-sample RMSE was obtained with a 2-rule complex-ANFIS-SICFIS model with 4 membership functions per feature. The regression plots of the global and local models are shown in Figure 5.8 and Figure 5.9 respectively.

Table 5.6: Charpy Best results given different number of rules.

		Training		Checking		Testing		All	
No. Rules	No. mF	Global	Local	Global	Local	Global	Local	Global	Local
2	4	14.58	14.59	17.44	17.47	18.01	17.92	15.73	15.75
3	4	12.76	13.39	19.75	20.37	18.36	19.08	15.28	15.91
4	4	13.12	13.67	18.07	18.27	21.67	22.07	15.83	16.27
5	4	11.94	12.30	21.07	22.00	18.06	18.20	15.10	15.55
6	3	11.49	12.89	19.77	19.74	19.50	20.82	14.86	15.87

*mF: Membership function

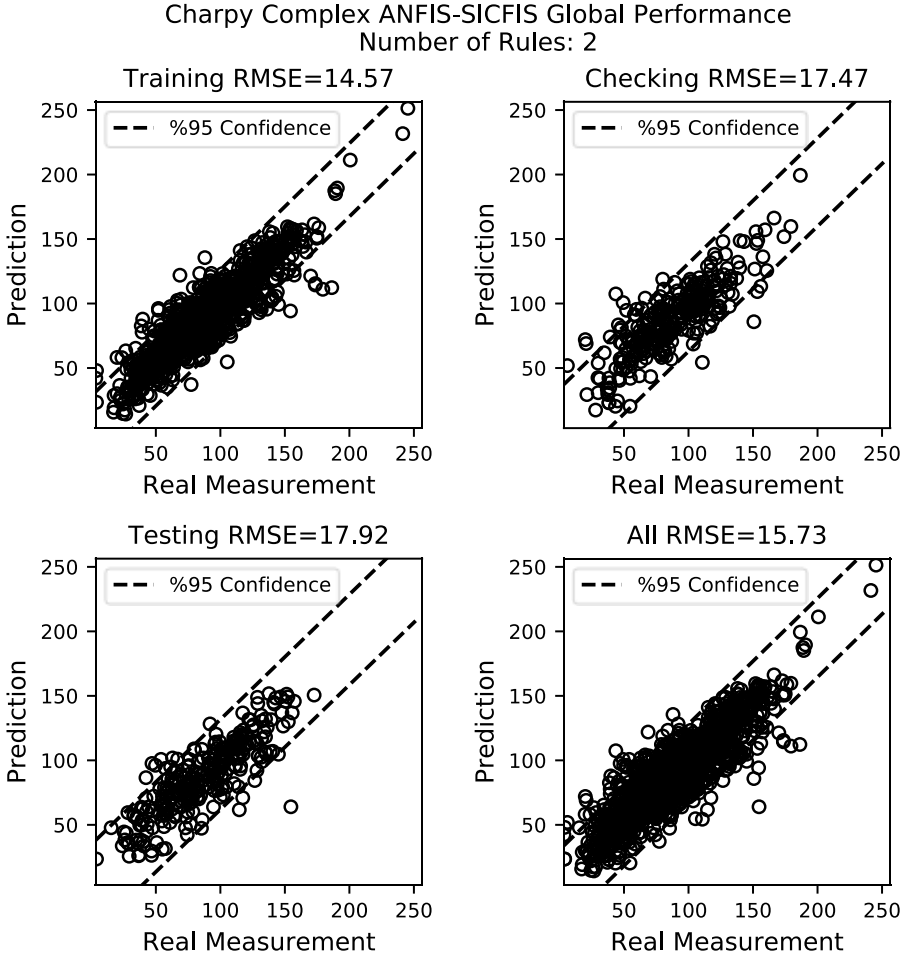


Figure 5.8: Charpy Impact complex ANFIS-SICFIS global performance 2 rules.

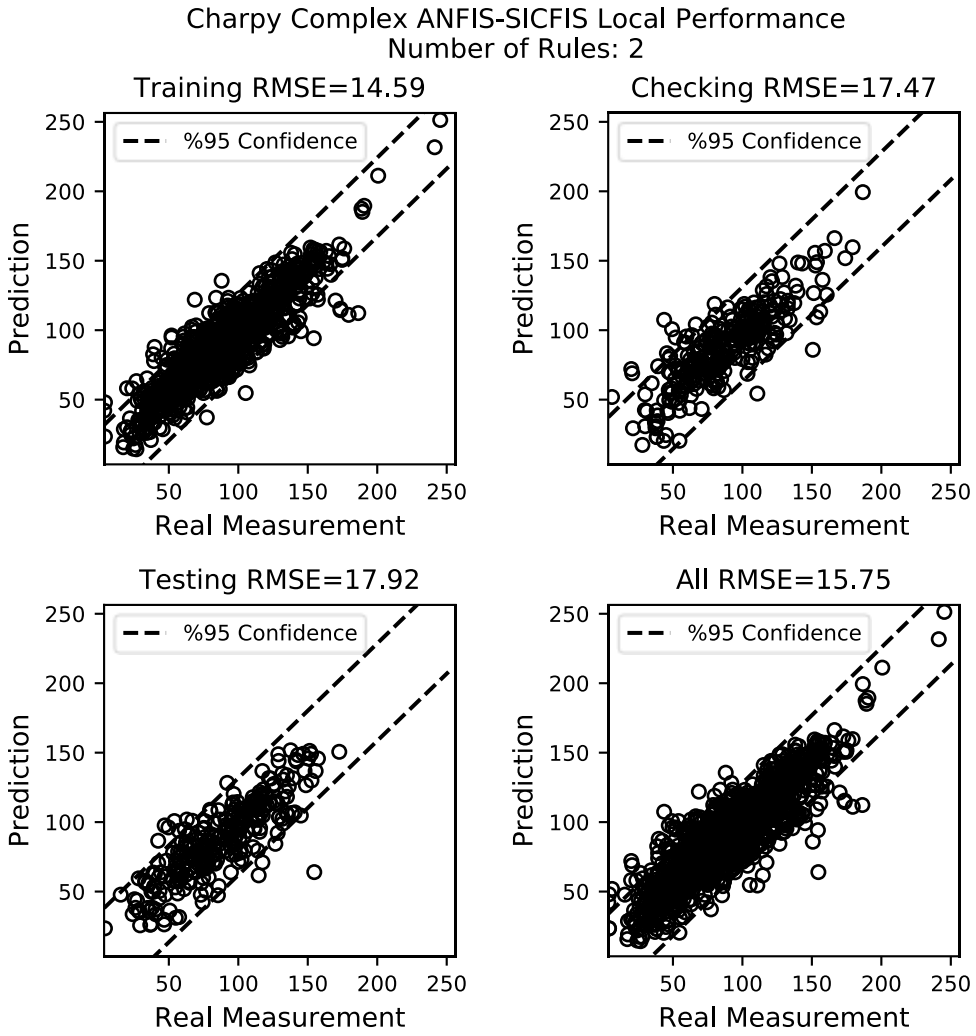


Figure 5.9: Charpy Impact complex ANFIS-SICFIS local performance 2 rules.

Table 5.7: Charpy results comparison.

Model	Training	Checking	Testing	All
Ensemble-NN[129],	12.60	17.30	19.4	14.79
GA-NN Optimized [129]	14.32	17.94	18.96	15.92
GA-NN Ensemble[129]	13.12	17.25	18.13	14.90
Normalized-SICFIS 6mF	15.41	19.45	17.98	16.66
Fast-SICFIS 5mF	15.38	19.63	18.52	16.77

*mF: Membership function

5.3.2 Tensile Strength Results

The same parameter grid search shown in Table 5.3 is implemented to the UTS dataset, with a training-checking partition of 70-30 respectively and 12 data points for the validation. The mean results and the corresponding standard deviation given a number of rules are shown in Table 5.8 and Table 5.9 respectively. The mean RMSE for the training decreases with the addition of rules, while the checking and testing mean RMSE increases slightly. However, no major differences are observed between the global and local performances for the training and checking partitions with the addition of rules. An increase in the mean RMSE is observed for the validation partition.

Just as in the case with the Charpy Impact test, in Figure 5.10, is observed that the addition of membership function results in a decreasing RMSE for the training partition and an increasing RMSE for the testing partition.

Table 5.8: UTS mean of results given different number of rules.

	Training		Checking		Testing		All	
No. Rules	Global	Local	Global	Local	Global	Local	Global	Local
2	32.99	34.00	40.32	41.38	52.90	53.72	35.45	36.47
3	32.20	34.38	42.71	44.73	57.14	57.90	35.81	37.91
4	31.37	33.80	41.98	44.82	70.80	74.29	35.10	37.67
5	29.25	31.95	41.29	43.24	60.46	60.40	33.49	35.85
6	29.56	32.81	41.08	44.09	62.07	63.50	33.61	36.72

Table 5.9: UTS standard deviation of results given different number of rules.

	Training		Checking		Testing		All	
No. Rules	Global	Local	Global	Local	Global	Local	Global	Local
2	1.66	1.83	3.01	3.19	10.15	11.35	1.67	1.95
3	2.22	2.18	4.00	4.61	12.14	11.78	2.41	2.58
4	1.52	2.12	3.62	3.95	9.67	11.43	1.71	2.13
5	1.67	1.56	3.50	3.78	10.03	9.64	1.80	2.02
6	1.97	2.56	2.61	3.33	12.48	12.11	1.37	2.28

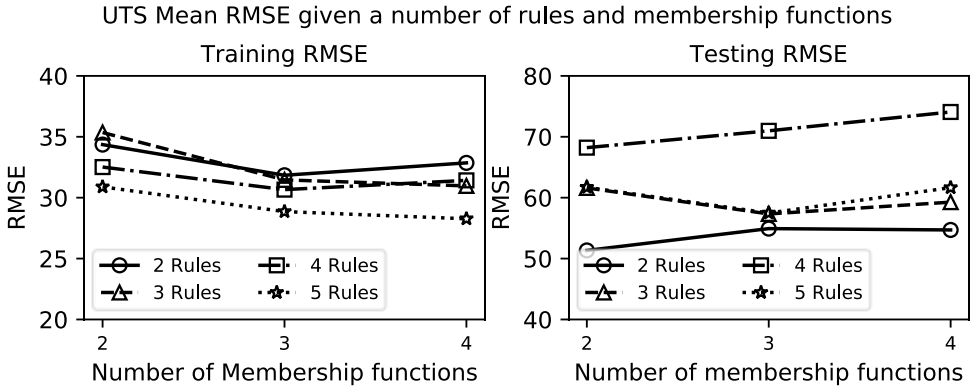


Figure 5.10: Effect of membership functions to performance

The best results given a number of rules are shown in Table 5.10. For comparison purposes the results obtained from different studies are shown in Table 5.11 as well as the results obtained in Chapter 4. The best out-of-sample RMSE was obtained with a 5-rule complex-ANFIS-SICFIS model with 3 membership functions per feature. The regression plots of the global and local models are shown in Figure 5.11 and Figure 5.12 respectively.

Table 5.10: UTS Best results given different number of rules.

		Training		Checking		Testing		All	
No. Rules	mF*	Global	Local	Global	Local	Global	Local	Global	Local
2	2	32.36	33.35	35.48	37.23	33.25	32.25	33.32	34.55
3	4	30.59	33.24	44.24	45.76	34.28	42.79	35.24	37.45
4	4	31.49	35.01	37.12	43.22	45.71	48.82	33.33	37.70
5	3	27.84	29.91	33.20	35.66	43.94	44.33	29.61	31.79
6	4	27.25	33.35	38.10	45.89	59.38	55.61	31.04	37.63

*mF: Membership function

Table 5.11: UTS result comparisons

Model	**Training**	**Testing**	**Validation**
IT2-Squared [40]	34.45	38.76	37.34
MOIT2FM [118]	36.33	40.52	34.77
IMOFM-M [118]	46.47	45.52	49.87
Normalized-SICFIS 4mF	35.64	36.25	55.05
Fast-SICFIS 4mF	33.69	39.26	52.3

*mF: Membership function

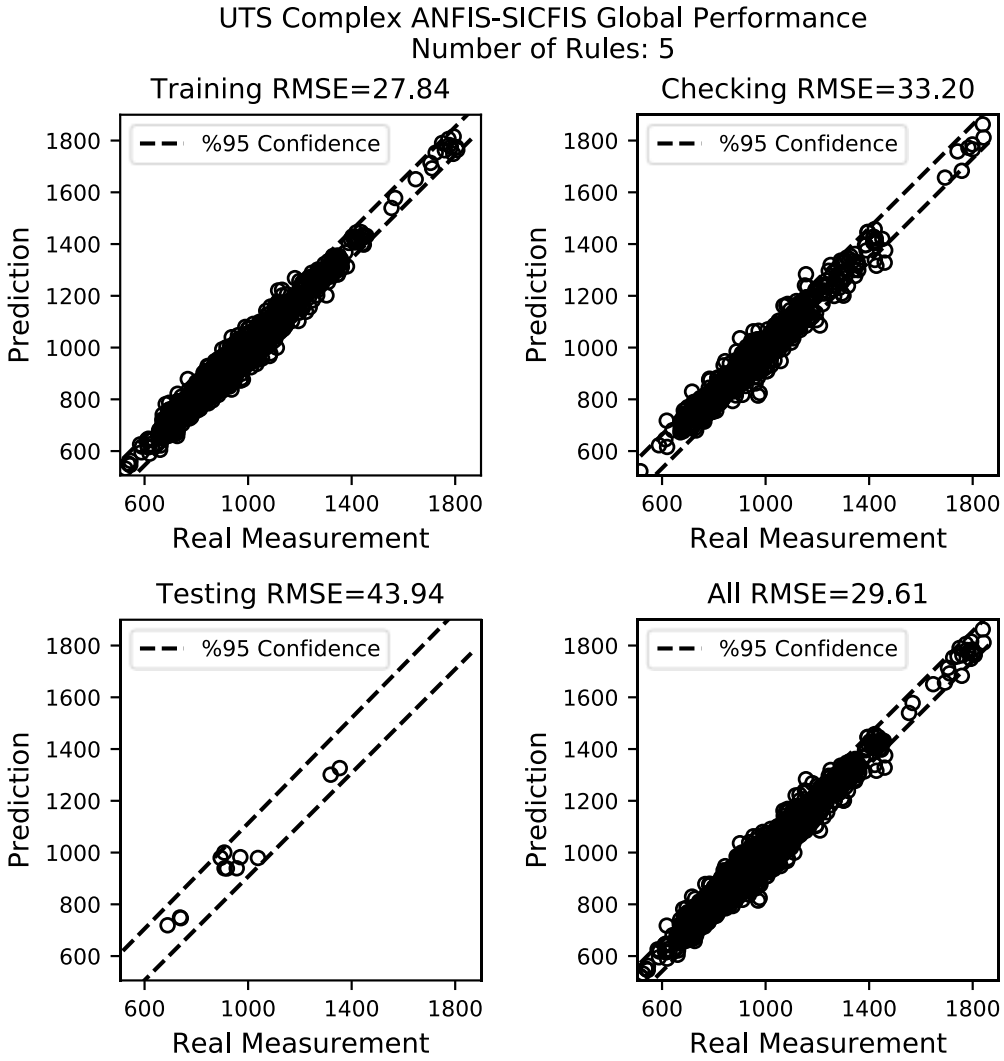


Figure 5.11: UTS complex ANFIS-SICFIS global performance 5 rules.

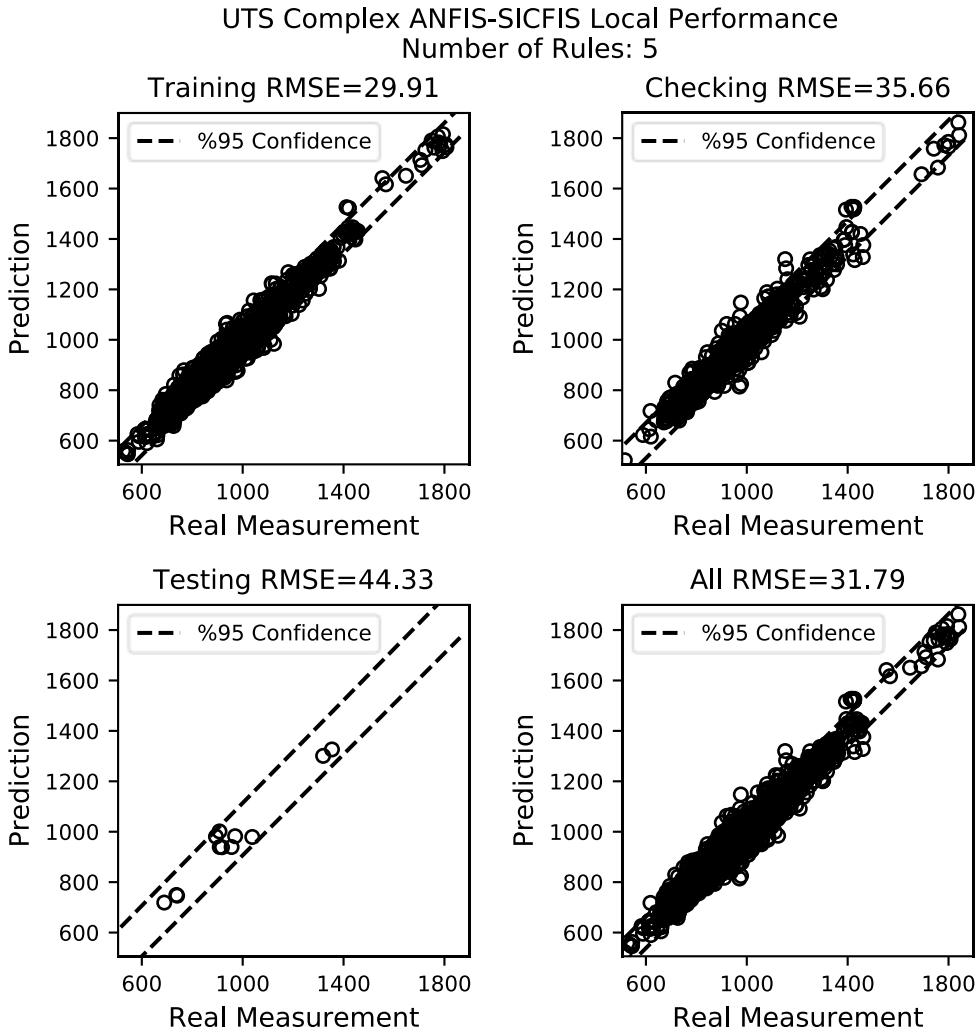


Figure 5.12: UTS complex ANFIS-SICFIS local performance 5 rules.

5.3.3 Bladder Cancer Results

A smaller parameter grid search is performed on the Bladder Cancer dataset, shown in Table 5.12 with a 70-30 partition for training and testing respectively. The mean and standard deviation RMSE results obtained from the parameter grid search given a number of rules is shown in Table 5.13 and Table 5.14 respectively. A decrease in

performance for the testing partition is observed with the addition of rules with just a slight increase in the training partition performance.

The best results obtained given a number of rules are shown in Table 5.15. The corresponding ROC curves and score scatter plots obtained from the best performing models are shown in Figure 5.13, Figure 5.14 and Figure 5.15. It is evident from the Table 5.13 and Table 5.15 that the ANFIS-SICFIS model overfits the bladder cancer dataset. For comparison purposes Table 5.16 shows the results obtained in previous studies as well as the results obtained in Chapter 4.

Table 5.12: Parameter grid search for the Bladder Cancer dataset.

Parameter	Values
Models	{Complex-ANFIS-SICFIS}
Optimization Method	{Consequents}
Number of rules	{2,3,4,5}
Number of membership functions per feature (SICFIS)	{2,3,4}
Fuzzy partition coefficient values	{1.2,1.8}
Number of k-fold cross validation per model	5
Maximum number of epochs	70

Table 5.13: Bladder Cancer Mean results.

	Training		Testing		All	
No. Rules	Global	Local	Global	Local	Global	Local
2	0.9086	0.8980	0.8699	0.8600	0.8954	0.8850
3	0.9168	0.8889	0.8703	0.8529	0.9009	0.8765
4	0.9182	0.8440	0.8723	0.7977	0.9025	0.8281
5	0.9184	0.8896	0.8693	0.8436	0.9016	0.8739

Table 5.14: Bladder Cancer standard deviation results.

	Training		Testing		All	
No. Rules	Global	Local	Global	Local	Global	Local
2	0.0063	0.0105	0.0102	0.0104	0.0034	0.0073
3	0.0064	0.0142	0.0121	0.0140	0.0034	0.0098
4	0.0073	0.0777	0.0115	0.0773	0.0046	0.0770
5	0.0076	0.0234	0.0134	0.0187	0.0029	0.0195

Table 5.15: Bladder Cancer best results given a number of rules and membership functions.

		Training		Testing		All	
No. Rules	No. mF*	Global	Local	Global	Local	Global	Local
2	2	0.9122	0.9034	0.8886	0.8734	0.9040	0.8928
3	3	0.9138	0.8658	0.8935	0.8667	0.9069	0.8665
4	3	0.9086	0.8481	0.8994	0.8397	0.9055	0.8453
5	3	0.9144	0.8441	0.8915	0.8041	0.9065	0.8292

*mF: membership function

Table 5.16 Bladder Cancer Results Comparison.

Model	Training	Testing
Cox [39]	0.83	0.82
LoR [39]	0.76	0.74
ANN [39]	0.88	0.84
T1 FIS [39]	0.88	0.83
T2 FIS [39]	0.92	0.91
Norm-SICFIS 4mF*	0.906	0.8852
Fast-SICFIS 4mF*	0.9015	0.8998

*mF: membership function

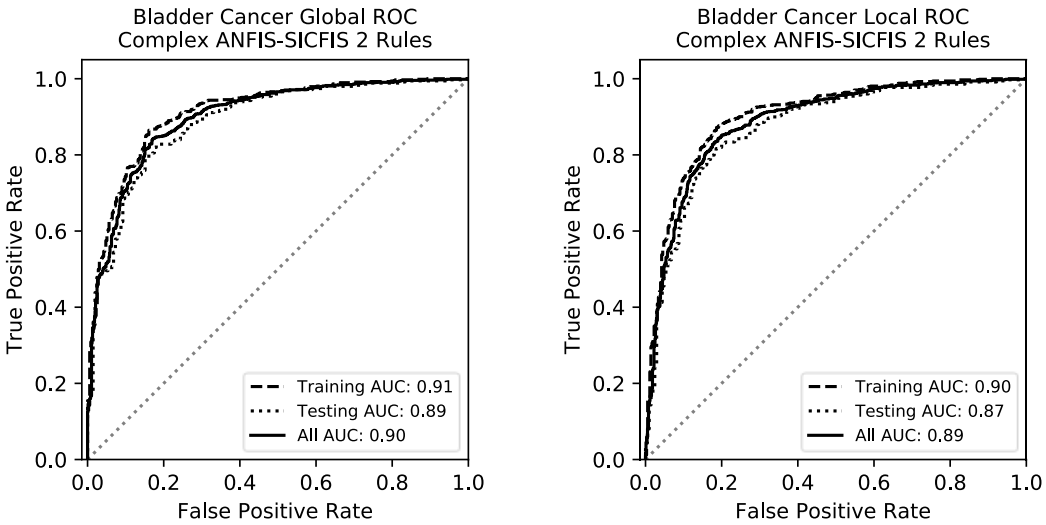


Figure 5.13: Bladder cancer ROC curves for the global (a) and local performance (b).

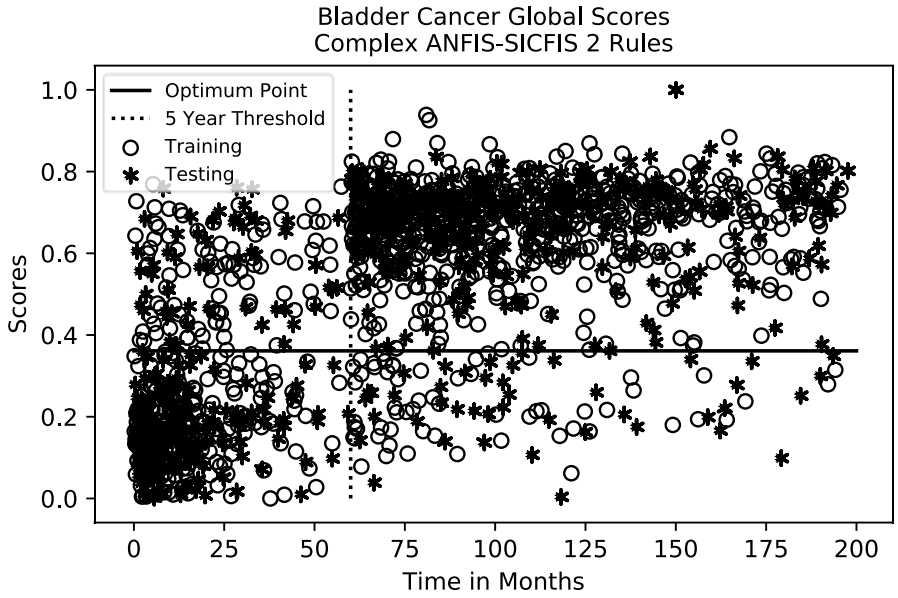


Figure 5.14: Bladder Cancer Global Scores.

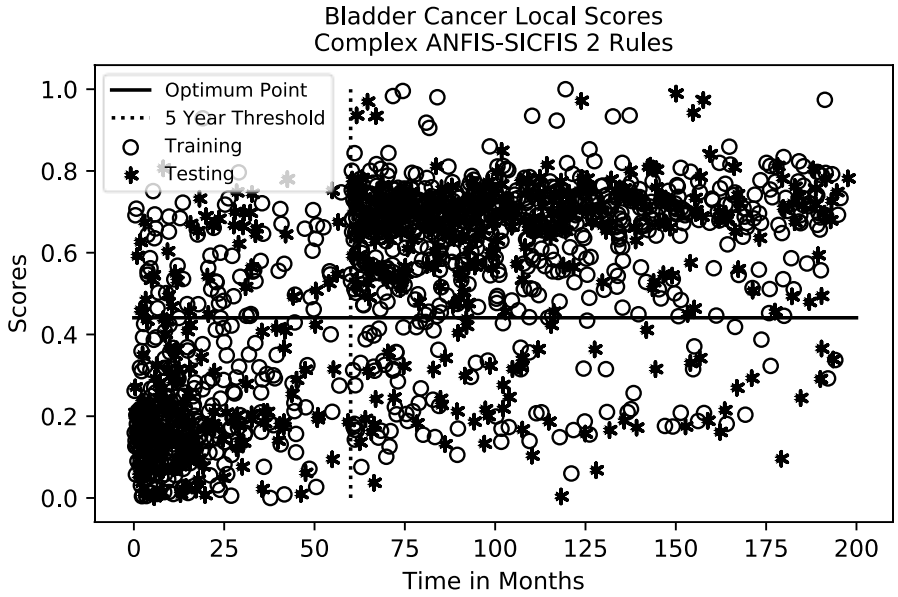


Figure 5.15: Bladder Cancer Local Scores.

5.3.4 Superconductivity Results

A summary of the results obtained from the superconductivity data set are shown in Table 5.17 Table 5.18 and. The best results obtained given a number of rules and membership functions is shown in Table 5.19. A result comparison is shown in Table 5.20.

Table 5.17: Superconductivity mean of results given different number of rules.

	Training		Checking		Testing		All	
No. Rules	Global	Local	Global	Local	Global	Local	Global	Local
2 – 2mF	12.60	12.66	13.82	13.87	13.39	13.42	12.97	13.02
2 – 3mF	12.23	12.28	13.44	13.50	13.09	13.11	12.61	12.65
3 – 2mF	13.00	13.18	13.91	14.04	13.61	13.67	13.27	13.42

Table 5.18: Superconductivity standard deviation of results given different number of rules.

	Training		Checking		Testing		All	
No. Rules	Global	Local	Global	Local	Global	Local	Global	Local
2 – 2mF	0.74	0.76	0.15	0.16	0.47	0.49	0.56	0.58
2 – 3mF	0.45	0.46	0.30	0.32	0.27	0.29	0.36	0.38
3 – 2mF	0.33	0.18	0.17	0.10	0.14	0.09	0.26	0.13

Table 5.19: Superconductivity best results given different number of rules.

	Training		Checking		Testing		All	
No. Rules	Global	Local	Global	Local	Global	Local	Global	Local
2 – 2mF	11.97	12.00	13.68	13.71	12.97	12.97	12.46	12.49
2 – 3mF	11.95	11.98	13.13	13.16	12.70	12.70	12.30	12.32
3 – 2mF	12.65	12.93	13.72	13.90	13.54	13.64	13.00	13.23

Table 5.20: Superconductivity Results Comparison.

Model	Training	Checking	Testing	All
Linear Regression [109]	NA	NA	17.6	NA
XG-Boost [109]	NA	NA	9.4	NA
ANFIS 8 Rules	13.37	16.27	16.08	14.42
ANN 10 hidden layers	13.42	13.50	14.23	13.58
ANN 20 hidden layers	12.54	13.39	12.93	12.76
Normalized-SICFIS 4mF	13.45	14.25	14.46	13.77
Fast-SICFIS 4mF	13.19	14.05	14.65	13.61

5.4 Summary

This work presented an improvement of the traditional ANFIS model, whose linear consequences are replaced with the SICFIS model, a non-linear and highly interpretable model. The compactness, interpretability properties and low computation required to train local SICFIS allows to create accurate rule-base system with a considerable low number of rules.

Two different modelling strategies where presented as well as three different optimization processes. From the exhaustive grid search it was determined that the best performances where obtained when the complex information from the local SICFIS was transmitted through the rules. Together with better performances, the ability to obtain an additional degree of information made the complex-ANFIS-SICFIS the clear choice for future work.

From the optimization strategies presented it was determined that optimizing solely the consequents would return the best performance, especially for local performance evaluation, additionally this strategy would reduce considerably the training times for larger datasets. The design of optimization algorithms that modify the premises of the rule-base while maintaining in consideration it's interpretability would require modifications to the objective function, it has been proposed the application of evolutionary algorithms in the optimization process in order to maintain interpretability of the rule-base premises [93], [95], [130], [131]. It is important to consider that evolutionary algorithms and other global optimization require to evaluate a large number of models which increasing exponentially computation times. It is therefore concluded that the premises of the rule-base should remain unchanged during the optimization processes.

The ANFIS-SICFIS was tested in four different datasets. The results obtained from the Charpy impact test are comparable with large and complex ANN models, this performance was obtained with just two rules. The results from the UTS dataset are the best obtained so far in the literature. The results from the Cancer dataset underperformed and overfitted the data, this may be caused by the large number of categorical features in the dataset and the application of a least square optimization algorithm instead of performing a survival analysis which is out of the scope of this work. Results obtained from the superconductivity dataset are superior to most modelling strategies.

Chapter 6

Mamdani Single Input Complex Fuzzy Inference System

6.1 Introduction

The consequence of the SICFIS proposed in Chapter 4 was defined as a complex singleton membership function. The disadvantage of utilizing a singleton membership function is the loss of vagueness and linguistic meaning a typical gaussian membership function may provide, as it is the case in Mamdani-FIS. Therefore, a Mamdani-SICFIS model, where the fuzzy complex singleton membership function is replaced with a complex Gaussian in order to better model uncertainties and increase interpretability is proposed.

According to Ramot et al [8], [55]. the magnitude of a CFS represents a traditional type-1 fuzzy set while the phase is a non-fuzzy quantity that defines the “context”. The CF membership function represents a trajectory in 3 dimensions, in contrast with the type 2 fuzzy set which represent a surface [132], the reason for this representation is that while a type-2 fuzzy set includes an additional degree of uncertainty, the CFS includes an additional non-fuzzy degree of information defined as the “context”.

Complex membership functions have been proposed previously [57]. The complex Gaussian membership functions proposed to date [74], [75] do not represent a trajectory in 3 dimensions whit a coupled phase and magnitude. The sinusoidal membership function proposed by Dick [73] does represents a 3 dimensional trajectory, where both the magnitude and the phase are coupled.

Sinusoidal and Gaussian membership functions are utilized for different purposes, while the sinusoidal membership function is utilized to model semi-periodic behaviour, Gaussian represent a region of space at a particular time [57]. The proposed complex Gaussian membership function is therefore the first linguistic membership function based on the CFS and CFL developed by Ramot et al. [8], [55].

6.2 Development of a Complex Gaussian Membership Function

The complex membership function should respect the following:

1) The magnitude represents a type-1 fuzzy membership function, the phase is a non-fuzzy quantity that represents the "context".

2) A complex membership function in 3 dimensions should represent a trajectory, not a surface.

3) A complex membership function should be equivalent to a traditional type-1 membership function when all the phases are equal to zero.

4) The defuzzification results in crisp complex number, with a magnitude and a phase.

5) Given points (3) and (4); when all the phases in a system are equal, that is when no interference occurs, the resultant magnitude should be equivalent to a traditional type-1 system. Given that an ordering does not exists in complex numbers, the phase should be taken into consideration together with a frame of reference.

In order to differentiate between real valued membership functions and complex membership function the symbol μ will be used for the former and ζ for the latter. The complex membership function will be defined as follows:

$$\zeta_S(x) = r_S e^{j\omega_S(x)} \tag{6.1}$$

where r represents the magnitude and ω the phase. The complex membership functions described in the following section maps real-valued inputs to the complex domain, $\mathbb{R} \to \mathbb{C}$.

6.2.1 Type-1 Membership Function Equations: Singleton and Gaussian Membership Functions

For the development of a complex fuzzy gaussian membership function it is necessary to first present the equations for a traditional type-1 singleton and Gaussian membership functions. The singleton membership function has a membership value of 1 when a variable x is equal to the singleton position b, and 0 otherwise. The Gaussian membership function is a normal convex function which has a value of 1 when a variable k is at the center of the gaussian membership function, defined as b in this example, the value decreases as the distance between k and b increases. In two dimensions the x-axis represents the value of the variable k. The position of the singleton membership function and the center of the Gaussian membership function are represented by the same variable b.

Singleton membership function:

$$y^{\text{Singleton}} = \mu^{\text{Singleton}}(k) = \begin{cases} 1, & if\ k = b \\ 0, & if\ k \neq b \end{cases} \tag{6.2}$$

Gaussian membership function:

$$y^{\text{Gaussian}} = \mu^{\text{Gaussian}}(k) = \exp\left(-\frac{1}{2}\left(\frac{(k-b)}{\sigma} \right)^2 \right) \tag{6.3}$$

Figure 6.1 shows the two-dimensional view of a gaussian membership function and singleton membership function, both functions centres are equal to 0.5 and the spread σ of the Gaussian membership function is equal to 0.2.

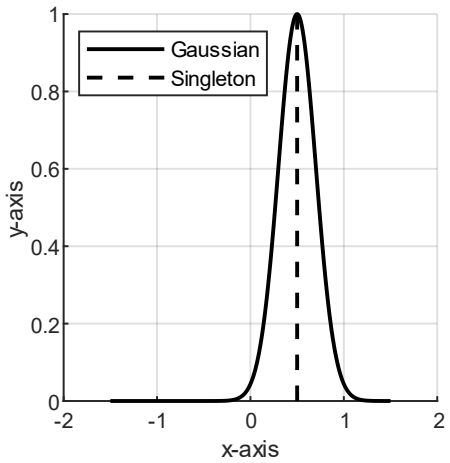


Figure 6.1: Two-dimension view of a Gaussian and singleton membership function, center *b=0.5* and σ*=0.2*.

6.2.2 Complex Singleton Membership Function

In the case of a complex fuzzy singleton membership function in 3-dimensions the *x*, and *y* axis represents the real and the imaginary plane respectively, while the *z* axis represents the membership value, which would be equal to the magnitude of the complex membership function. Therefore, taking a real-valued input *k* the magnitude and the phase of the complex singleton membership function is as follows:

$$z = \left|\zeta(k)\right| = \begin{cases} 1 & if \;\; k \cdot \cos(\varphi) = \beta_{\mathrm{Re}} \text{ and } k \cdot \sin(\varphi) = \beta_{\mathrm{Im}} \\ 0 & \text{otherwise} \end{cases} \tag{6.4}$$

$$\arg(\zeta(k)) = \varphi \tag{6.5}$$

where *z* represents the magnitude of the fuzzified variable. φ represents the phase of the membership function, β_{Re} and β_{Im} represent the real and the imaginary coordinates of the singleton location and are calculated as follows:

$$\beta_{\text{Re}} = b \cdot \cos(\varphi) \tag{6.6}$$

$$\beta_{\text{Im}} = b \cdot \sin(\varphi)\, j \tag{6.7}$$

It should be noted that the magnitude of the complex singleton membership function is equivalent to a type-1 singleton membership function with the addition of the context represented by the phase variable φ. This is in accordance with points 1 and 3, and the complex singleton membership function can be tough as a traditional type-1 singleton membership function whose centers rotates according to the value of the context variable φ. An example of a singleton membership function located at $|\beta| = 0.5$ and $\varphi = 45°$ is shown in Figure 6.2. The dotted lines represent the slope where the trajectory of *k* travels as well as the location of β_{Re} and β_{Im} for visual reference.

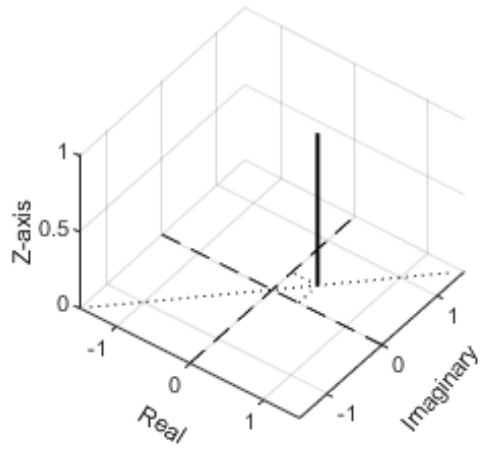


Figure 6.2: Three-dimension view of a singleton membership function, center $|\beta|$ =0.5 and a pahse $\varphi = 45°$

6.2.3 Complex Gaussian Membership Function

Just as in the case of the complex singleton membership function, in 3-dimensions the complex Gaussian membership function should represent a traditional type-1 Gaussian membership function whose trajectory is rotated by φ degrees. The parametric equations in 3-dimensions of the complex Gaussian membership function for the *x*, *y* and *z* axis are as follows:

$$x = k * \cos(\varphi) \tag{6.8}$$

$$y = k * \sin(\varphi) \tag{6.9}$$

$$z = \left|\zeta(k)\right| = \exp\left(-\frac{1}{2}\left(\frac{\left|(x,y)-\beta\right|}{\sigma}\right)^2\right) \tag{6.10}$$

$$\arg(x,y) = \varphi \tag{6.11}$$

where x and y represent the real and the imaginary values, because the phase is constant, the values should move in a straight line, and the slope represent the phase φ. The *z* axis represents the magnitude of the CFS, which represents a traditional type-1 fuzzy set and its shape should then be of a type-1 gaussian membership function. An example of a complex-Gaussian membership function is shown in Figure 6.3.

During the rule interference process, it is necessary to aggregate the real and imaginary parts respectively. In the case of the complex Gaussian membership function, it is necessary to separate the real and imaginary components of the complex Gaussian membership function to assign the proportional degree of membership. This can be accomplished by multiplying the Gaussian membership function by the absolute value of a sine and cosine function. The absolute value is utilized given that the membership value needs to remain positive. The real and imaginary components of a complex

Gaussian membership function are shown in Figure 6.4. The complex Gaussian membership function is as follows:

$$\zeta(k) = \exp\left(-\frac{1}{2}\left(\frac{\|(x,y)-\beta\|}{\sigma}\right)^2\right)\left|\cos(\varphi)\right| + \exp\left(-\frac{1}{2}\left(\frac{\|(x,y)-\beta\|}{\sigma}\right)^2\right)\left|\sin(\varphi)\right| j \qquad (6.12)$$

where $\|(x,y)-\beta\|$ represents the distance from the origin to the centre of the membership function, σ is the spread and φ is the angle of the complex Gaussian membership function.

The proposed complex Gaussian membership increases the interpretability of the system given its proximity to human natural languages. While the complex Singleton membership function allows to represent linguistic variables with context, the complex Gaussian membership function adds the vagueness characteristic of human speech. The oven example shown in Section 2.1 demonstrates how a normal membership function is better suited for representing information in an intuitive manner, something that cannot be fully achieved with a complex Singleton membership function.

By adding context to a complex Gaussian membership function, it is possible to increase the information representation in the system. As an example, an Oven might be considered “Hot” at a certain temperature, given other circumstances such temperature might not be considered “Hot” at all. The context is expressed as interference, and depends on the angle φ of the complex membership function.

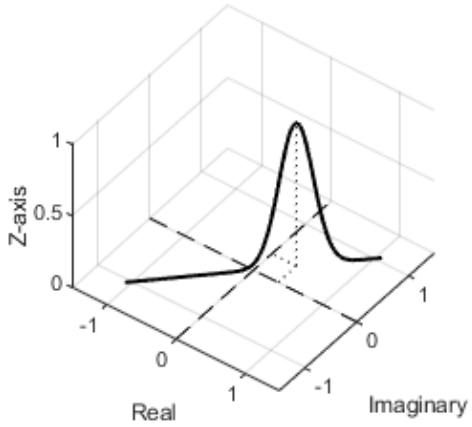


Figure 6.3: Three-dimension view of a complex Gaussian membership function, center $|\beta|$ =0.5, spread σ =0.2 and phase $\varphi = 45°$.

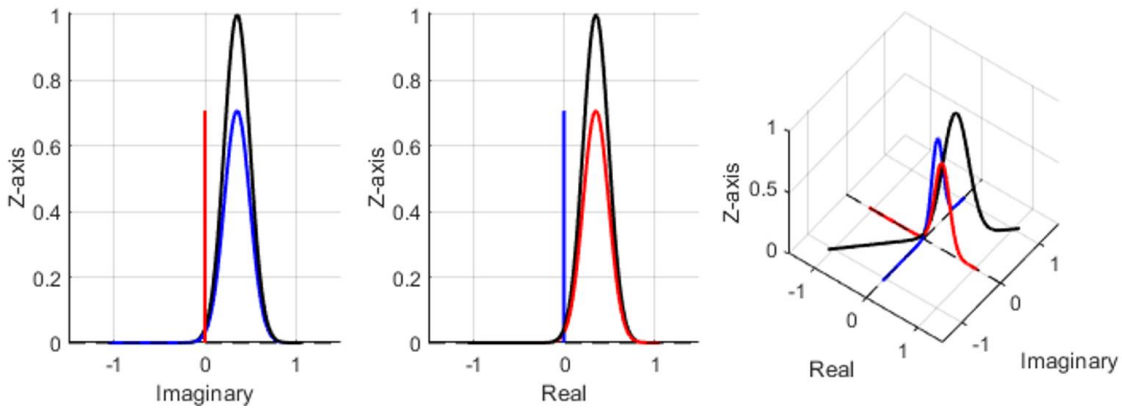


Figure 6.4: Three-dimension view of a complex Gaussian membership function and the corresponding real and imaginary projection. Center $|\beta|$ =0.5, spread σ =0.2 and phase $\varphi = 45°$.

6.2.4 Interference and Defuzzification

The interference and defuzzification operation are relatively straight forward. The complex gaussian membership function is represented by its real and imaginary part, each is aggregated respectably, creating an interference. The obtained crisp value is a

complex quantity, and the measured output is the magnitude, while the phase is used for additional information. The COG de-fuzzification is as follows:

$$h^{\mathrm{Re}} = \frac{\sum_{d=1}^{D}\sum_{r=1}^{R}\zeta_r^{\mathrm{Re}}(k_d)\cdot x_d}{\sum_{d=1}^{D}\sum_{r=1}^{R}\zeta_r^{\mathrm{Re}}(k_d)} \tag{6.13}$$

$$h^{\mathrm{Im}} = \frac{\sum_{d=1}^{D}\sum_{r=1}^{R}\zeta_r^{\mathrm{Im}}(k_d)\cdot y_d}{\sum_{d=1}^{D}\sum_{r=1}^{R}\zeta_r^{\mathrm{Re}}(k_d)} \tag{6.14}$$

$$f(x_i) = \sqrt{\left(h^{\mathrm{Re}}\right)^2 + \left(h^{\mathrm{Im}}\right)^2} \tag{6.15}$$

As explained in the previous section, the real and the imaginary parts of the complex gaussian membership function corresponds to the projections to their respective axis. The particle moves at *k* intervals in the space, at a rate of $k\cdot\cos(\varphi)$ in the *x*-axis and $k\cdot\sin(\varphi)$ in the y-axis as shown in Figure 6.5. The equations (6.12)-(6.15) comply with the objectives formulated at the beginning of this Chapter.

6.2.4.1 Defuzzification and Equivalence to Type-1 System

One of the essential requirements for the proposed complex gaussian membership function is the equivalence to a type-1 membership function when all the phases are equal to zero. Additionally, the proposed membership function is equivalent to a type-1 system when all the phases in a system are equal, the magnitude of the defuzzied value should remain constant as there is no interference, below is an example of the defuzzification of two complex gaussian membership function and the defuzzification of two type-1 gaussian membership functions. Table 6.1 shows the parameters of both the complex and the type-1 Gaussian membership functions. The graphical

representation of the defuzzification are shown in Figure 6.6 and Figure 6.7 for the type-1 and the complex membership function respectively.

Both the magnitude of the complex defuzzified value and the absolute value of the type-1 defuzzification are the same. Complex numbers are not ordered; therefore the resultant number has an phase of 240° and the type-1 quantity has a negative sign.

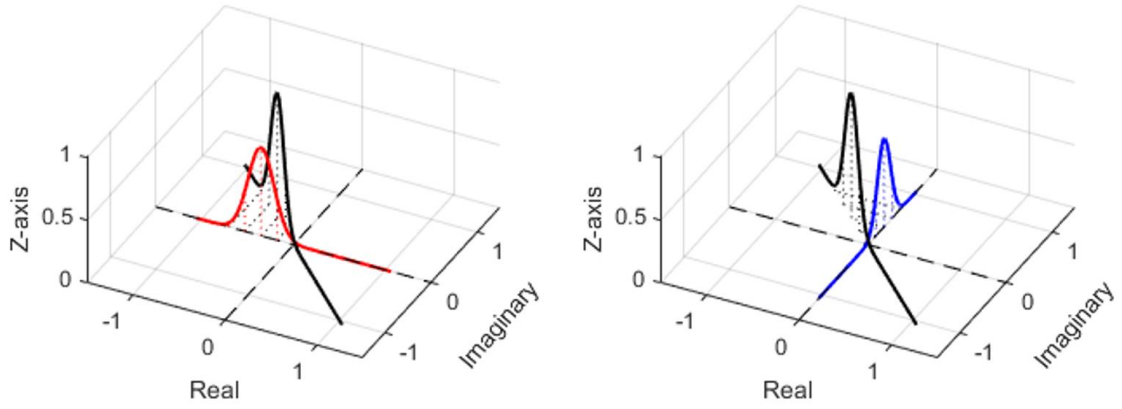


Figure 6.5: Three-dimension view of a Gaussian membership function. Center $|\beta|$ =0.5, spread σ =0.2 and phase $\varphi = 135°$.

Table 6.1 Complex and type-1 defuzzification

	Complex Gaussian	Type -1 Gaussian
Sigma	[0.2],[0.3]	[0.2],[0.3]
Centre	[0.7],[0.1]	[-0.7],[0.1]
Angle(degrees)	[240],[60]	NA
Defuzzified value	0.220, 240°	-0.220

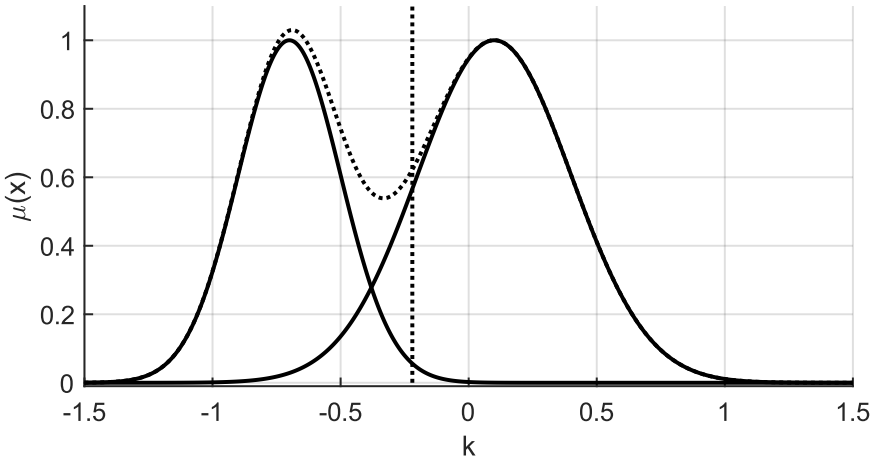


Figure 6.6: Type-1 COG defuzzification Sigma=[0.2,0.3], centres =[-0.7,0.1].

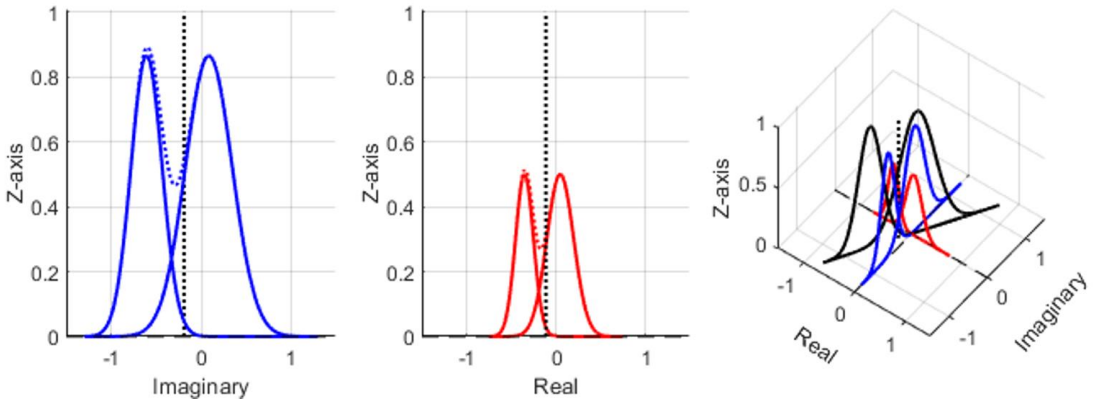


Figure 6.7: complex defuzzification Sigma:[0.2,0.3], centres:[0.7,0.1], angles = [240,60].

6.3 The Mamdani-Single Input Complex Fuzzy Inference System Model

The Mamdani-SICFIS just as the SICFIS is a single rule per feature partition rule-base FIS, each rule has one premise and one consequent, the premises are composed of

type-1 Gaussian membership functions and the consequents are composed of complex Gaussian membership function defined in (6.12).

Table 6.2: Mamdani SICFIS rule-base

Premise	Consequence
$IF\ x_1\ is\ A_1^1\ THEN\ y = is\ \zeta_1^1$	
$IF\ x_1\ is\ A_1^2\ THEN\ y = is\ \zeta_1^2$	
$IF\ x_2\ is\ A_2^1\ THEN\ y = is\ \zeta_2^1$	
$IF\ x_2\ is\ A_2^2\ THEN\ y = is\ \zeta_2^2$	
⋮	
$IF\ x_P\ is\ A_P^{S_P}\ THEN\ y = is\ \zeta_P^{S_P}$	

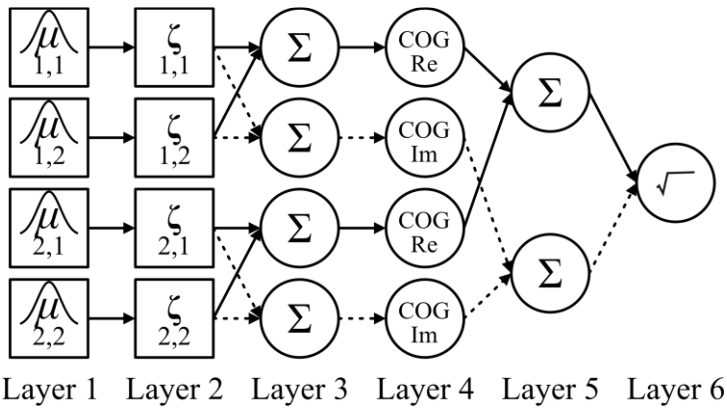


Figure 6.8 Mamdani-SICFIS architecture.

The Mamdani SICFIS can be described as a 6 layered FIS, the first layer fuzzifies the input utilizing a type-1 Gaussian membership function as follows:

$$O_{p,s_p}^1 = \mu_{p,s_p}(x_p) = \exp\left(-\frac{1}{2}\left(\frac{x_p - c_{p,s_p}^{\mu}}{\sigma_{p,s_p}^{\mu}}\right)^2\right) \tag{6.16}$$

The second layer calculates the consequents of the rules utilizing the complex gaussian membership function, the real and imaginary components of the complex gaussian membership function are as follows:

$$O_{p,s_p}^{2,\mathrm{Re}} = \zeta_{p,s_p}^{\mathrm{Re}} = \exp\left(-\frac{1}{2}\left(\frac{\left(k_j - c_{p,s_p}^{\zeta}\right)}{\sigma_{p,s_p}^{\zeta}}\right)^2\right)\left|\cos(\varphi_{p,s_p})\right| \tag{6.17}$$

$$O_{p,s_p}^{2,\mathrm{Im}} = \zeta_{p,s_p}^{\mathrm{Im}} = \exp\left(-\frac{1}{2}\left(\frac{\left(k_j - c_{p,s_p}^{\zeta}\right)}{\sigma_{p,s_p}^{\zeta}}\right)^2\right)\left|\sin(\varphi_{p,s_p})\right| \tag{6.18}$$

The third layer aggregates the real and imaginary components of the complex Gaussian membership function respectively.

$$O_p^{3,\mathrm{Re}} = \sum_{s_p=1}^{S_p} \mu_{p,s_p}(x_p) \cdot \zeta_{p,s_p}^{\mathrm{Re}} \tag{6.19}$$

$$O_p^{3,\mathrm{Im}} = \sum_{s_p=1}^{S_p} \mu_{p,s_p}(x_p) \cdot \zeta_{p,s_p}^{\mathrm{Im}} \tag{6.20}$$

The fourth layer performs the defuzzification operation:

$$O_p^{4,\mathrm{Re}} = \frac{\sum_{s_p=1}^{S_p}\sum_{d=1}^{D} \zeta_{p,s_p}^{\mathrm{Re}}(k_d) \cdot \mu_{p,s_p}(x_p) \cdot k_d \cdot \cos(\varphi_{p,s_p})}{\sum_{s_p=1}^{S_p}\sum_{d=1}^{D} \zeta_{p,s_p}^{\mathrm{Re}}(k_d) \cdot \mu_{p,s_p}(x_p)} \tag{6.21}$$

$$O_p^{4,\mathrm{Im}} = \frac{\sum_{s_p=1}^{S_p}\sum_{d=1}^{D} \zeta_{p,s_p}^{\mathrm{Im}}(k_d) \cdot \mu_{p,s_p}(x_p) \cdot k_d \cdot \sin(\varphi_{p,s_p})}{\sum_{s_p=1}^{S_p}\sum_{d=1}^{D} \zeta_{p,s_p}^{\mathrm{Im}}(k_d) \cdot \mu_{p,s_p}(x_p)} \tag{6.22}$$

The fifth layer performs the rule interference layer is the output of the system as follows:

$$O^{5,\mathrm{Re}} = h^{\mathrm{Re}} = \sum_{p=1}^{P} O_p^{4,\mathrm{Re}} \tag{6.23}$$

$$O^{5,\mathrm{Im}} = h^{\mathrm{Im}} = \sum_{p=1}^{P} O_p^{4,\mathrm{Im}} \tag{6.24}$$

The sixth layer calculates the final output as follows:

$$f(\mathbf{x}) = \sqrt{\left(h^{\mathrm{Re}}\right)^2 + \left(h^{\mathrm{Im}}\right)^2} \tag{6.25}$$

6.3.1 Optimization

The optimization algorithm is the LM and the derivative equations are as follows:

$$\frac{\partial f}{\partial \varphi_{p,s_p}} = \left(\frac{\partial f}{\partial h_{\mathrm{Re}}} \frac{\partial h^{\mathrm{Re}}}{\partial \varphi_{p,s_p}} + \frac{\partial f}{\partial h_{\mathrm{Im}}} \frac{\partial h^{\mathrm{Im}}}{\partial \varphi_{p,s_p}} \right) \tag{6.26}$$

$$\frac{\partial f}{\partial \sigma^{\mu}_{p,s_p}} = \left(\frac{\partial f}{\partial h_{\mathrm{Re}}} \frac{\partial h_{\mathrm{Re}}}{\partial \mu_{p,s_p}} \frac{\partial \mu_{p,s_p}}{\partial \sigma_{p,s_p}} + \frac{\partial f}{\partial h_{\mathrm{Im}}} \frac{\partial h_{\mathrm{Im}}}{\partial \mu_{p,s_p}} \frac{\partial \mu_{p,s_p}}{\partial \sigma_{p,s_p}} \right) \tag{6.27}$$

$$\frac{\partial f}{\partial c^{\mu}_{p,s_p}} = \left(\frac{\partial f}{\partial h_{\mathrm{Re}}} \frac{\partial h_{\mathrm{Re}}}{\partial \mu_{p,s_p}} \frac{\partial \mu_{p,s_p}}{\partial c_{p,s_p}} + \frac{\partial f}{\partial h_{\mathrm{Im}}} \frac{\partial h_{\mathrm{Im}}}{\partial \mu_{p,s_p}} \frac{\partial \mu_{p,s_p}}{\partial c_{p,s_p}} \right) \tag{6.28}$$

$$\frac{\partial f}{\partial \sigma^{\zeta}_{p,s_p}} = \left(\frac{\partial f}{\partial h^{\mathrm{Re}}} \frac{\partial h^{\mathrm{Re}}}{\partial \sigma^{\zeta}_{p,s_p}} + \frac{\partial f}{\partial h^{\mathrm{Im}}} \frac{\partial h^{\mathrm{Im}}}{\partial \sigma^{\zeta}_{p,s_p}} \right) \tag{6.29}$$

$$\frac{\partial f}{\partial c^{\zeta}_{p,s_p}} = \left(\frac{\partial f}{\partial h^{\mathrm{Re}}} \frac{\partial h^{\mathrm{Re}}}{\partial c^{\zeta}_{p,s_p}} + \frac{\partial f}{\partial h^{\mathrm{Im}}} \frac{\partial h^{\mathrm{Im}}}{\partial c^{\zeta}_{p,s_p}} \right) \tag{6.30}$$

The application of such optimization algorithm presents several challenges, given the computation of the defuzzification operation. The LM optimization requires the calculation of a pseudoinverse matrix. Additionally, if parallel computation is implemented, the size of the matrix increases exponentially, therefore increasing the computational complexity of the pseudoinverse matrix calculation.

6.4 Results

6.4.1 Charpy Impact Test

For the Charpy impact dataset the parameter grid is shown in Table 6.3. The RMSE index is used to measure the performance of the models. A summary of the results of models is shown in Table 6.4. The best results given a number of membership functions are shown in Table 6.5. The regression plot of the best performing model is shown in Figure 6.9

Table 6.3: Charpy impact Mamdani-SICFIS parameter grid.

Parameter	Values
Model	Mamdani-SICFIS
Optimization Method	LM
Number of membership functions per feature	{2,3,4,5,6}
Initial LM coefficient	20
Number of k-fold cross validation per model	5
Maximum number of epochs	90
Training-Checking-Testing partitions	[65-18-17]

Table 6.4: Charpy Impact Mamdani-SICFIS Results Summary.

No. mF	Training		Checking		Testing		All	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
2mF	18.45	0.77	19.33	0.53	21.71	0.78	19.20	0.67
3mF	17.57	1.53	21.22	2.12	21.61	1.68	19.02	1.49
4mF	17.61	1.65	21.84	1.73	22.66	1.87	19.40	0.95
5mF	16.17	0.31	19.43	1.81	19.23	1.32	17.36	0.19
6mF	15.78	0.75	21.18	0.83	22.40	1.70	18.11	0.76

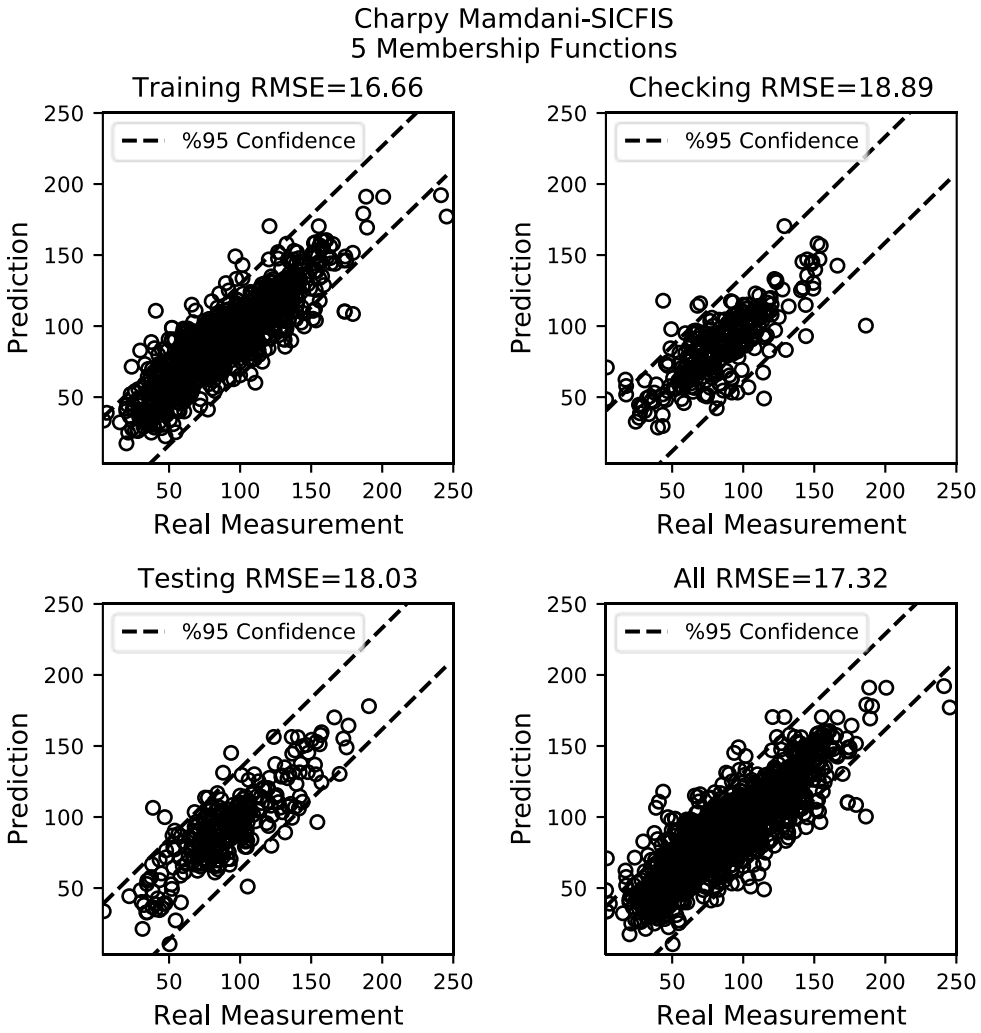


Figure 6.9: Charpy Mamdani-SICFIS 5 membership Functions (mF) regression plots.

Table 6.5: Charpy Impact Mamdani-SICFIS Best Results.

No. mF	Training	Checking	Testing	All
2mF	17.75	19.42	20.91	18.63
3mF	17.00	21.65	19.33	18.32
4mF	18.83	22.63	20.54	19.86
5mF	16.66	18.89	18.03	17.32
6mF	15.68	20.10	21.14	17.56

6.4.2 Tensile Strength

For the UTS dataset the parameter grid is shown in Table 6.6 the RMSE is used to measure the performance of the model. A summary of the results of models are shown in Table 6.7. The best results given a number of membership functions are shown in Table 6.8. The regression plots of the best performing model are shown in Figure 6.10.

Table 6.6: UTS Mamdani-SICFIS parameter grid.

Parameter	Values
Model	Mamdani-SICFIS
Optimization Method	LM
Number of membership functions per feature	{2,3,4,5,6}
Initial LM coefficient	20
Number of k-fold cross validation per model	5
Maximum number of epochs	90
Training-Checking-Testing partitions	[65-18-17]

Table 6.7: UTS Mamdani-SICFIS results summary.

No. mF	Training		Checking		Validation		All	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
2mF	45.07	4.47	46.44	4.05	78.51	5.06	45.63	4.33
3mF	42.45	3.89	46.76	5.33	68.97	12.24	43.90	4.35
4mF	36.90	0.68	40.79	1.51	63.36	3.01	38.22	0.46
5mF	39.97	1.95	47.58	6.01	63.32	4.89	42.55	2.45
6mF	45.92	7.41	52.33	6.28	76.46	18.10	48.06	7.06

Table 6.8: UTS Mamdani-SICFIS best results.

No. mF	Training	Checking	Testing	All
2mF	40.70	42.54	73.35	41.40
3mF	37.97	40.40	58.04	38.79
4mF	37.06	38.89	62.78	37.72
5mF	38.32	41.44	66.87	39.40
6mF	37.54	44.96	69.62	40.04

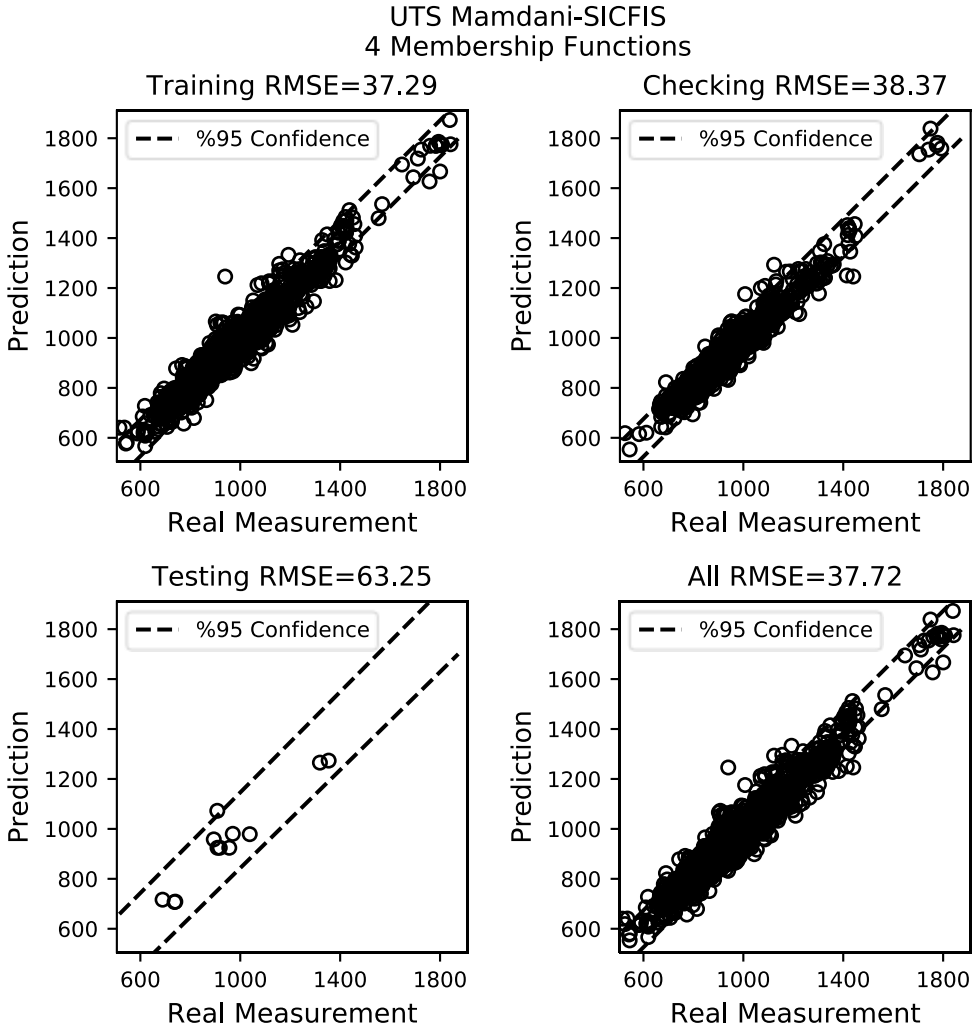


Figure 6.10: UTS Mamdani-SICFIS 4 membership Functions regression plots.

6.4.3 Bladder Cancer

For the Bladder Cancer dataset, the parameter grid is shown in Table 6.9 the RMSE is used to measure the performance of the models during training. A summary of the results of models measured utilizing the AUC are shown in Table 6.10. The best results given a number of membership functions are shown in Table 6.11. The ROC curves of the best performing model is shown in Figure 6.11 and the scatter plot of the scores is shown in Figure 6.12

Table 6.9: Bladder Cancer Mamdani-SICFIS parameter grid.

Parameter	Values
Model	Mamdani-SICFIS
Optimization Method	LM
Number of membership functions per feature	{2,3,4}
Initial LM coefficient	20
Number of k-fold cross validation per model	5
Maximum number of epochs	90
Training-Checking-Testing partitions	[65-18-17]

Table 6.10: Bladder Cancer Mamdani-SICFIS results summary.

No. mF	Training		Testing		All	
	Mean	SD	Mean	SD	Mean	SD
2mF	0.9022	0.0046	0.8753	0.0069	0.8941	0.0026
3mF	0.8772	0.0186	0.8483	0.0171	0.8684	0.0176
4mF	0.8914	0.0106	0.8815	0.0168	0.8886	0.0077

Table 6.11: Bladder Cancer Mamdani-SICFIS best results.

No. mF	Training	Testing	All
2mF	0.9075	0.8704	0.8957
3mF	0.8910	0.8726	0.8855
4mF	0.9083	0.8604	0.8952

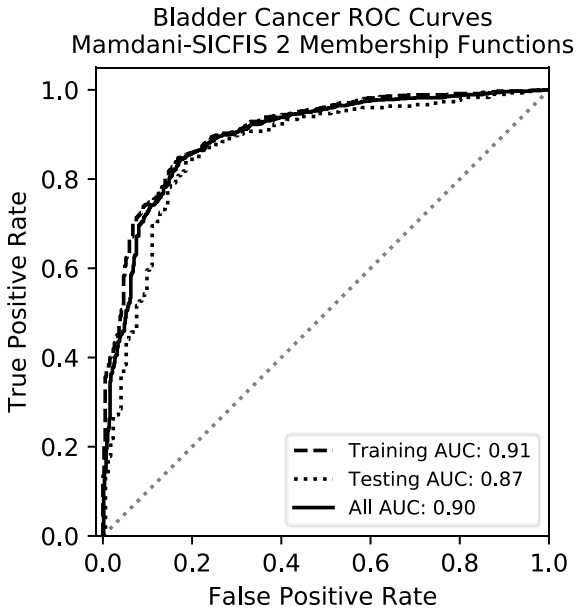


Figure 6.11: Bladder Cancer Mamdani-SICFIS 2 membership Functions ROC curves.

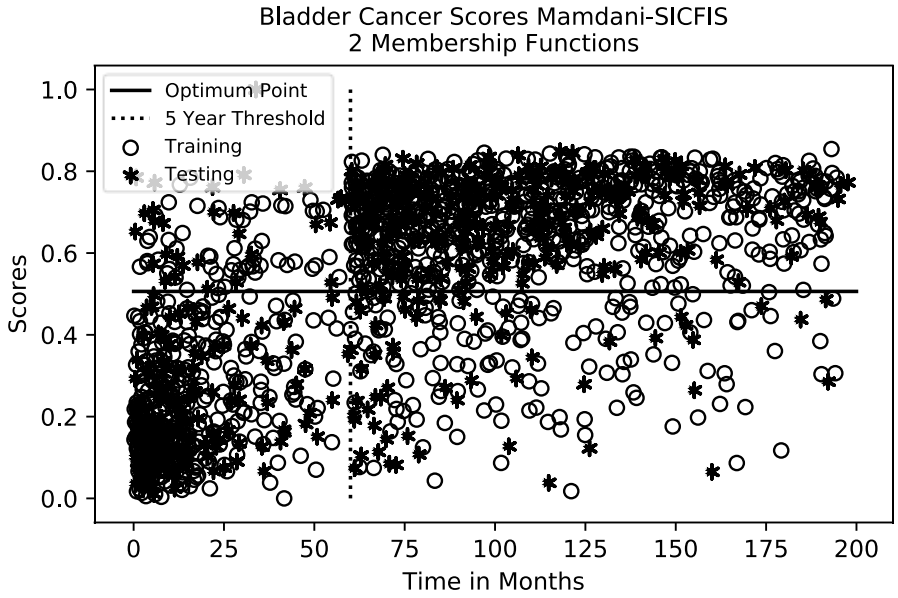


Figure 6.12: Bladder Cancer Mamdani-SICFIS 2 membership Functions Scores.

6.4.4 Superconductivity Results

The superconductivity results are shown in Table 6.12. The data partition is 65-18-17 for training, checking and testing respectively.

Table 6.12: Superconductivity results.

	Training	Checking	Testing	All
Mamdani 2mF	16.93	17.33	16.97	17.01
Mamdani 3mF	16.74	17.04	16.55	16.76

6.5 Charpy Impact Magnitude-Phase Plots Comparison Between SICFIS Models

In order to perform some comparison between the Mamdani, normalized and fast SICFIS models beyond the prediction accuracy, the magnitude and phase plots of the

output of three features, carbon, tempering temperature and impact temperature are shown in Figure 6.13, Figure 6.14 and Figure 6.15 respectively, each feature is partitioned by 5 membership functions.

Table 6.13: Charpy impact normalized, fast and Mamdani-SICFIS best results given 5 membership functions (mF).

	Training	Checking	Testing	All
Normalized 5mF	15.23	21.12	19.75	17.25
Fast 5mF	15.38	19.63	18.52	16.77
Mamdani 5mF	16.66	18.89	18.03	17.32

On the one hand the sharp changes shown in Figure 6.15 may result in overfitting, on the other hand the small changes shown in Figure 6.13 may result in an underperforming model. From Table 6.13 it can be observed that the Mamdani SICFIS model obtained the best out-of-sample RMSE in comparison with the normalized and fast SICFIS models. Therefore, it may be concluded that the Mamdani-SICFIS model may model uncertainties more appropriately in the Charpy impact test dataset than the normalized and fast SICFIS models.

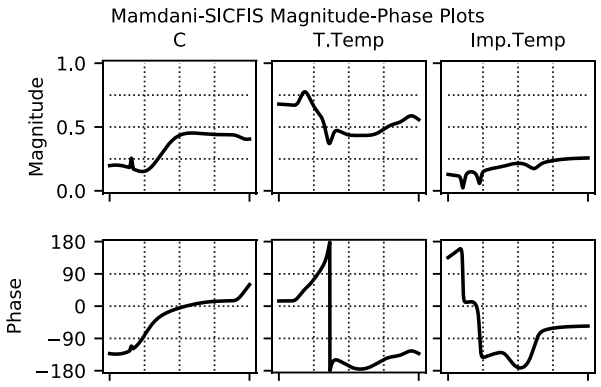


Figure 6.13: Magnitude-Phase plots for the Mamdani-SICFIS model for Carbon, Tempering Temperature (T.Temp) and Impact Temperature (Imp. Temp).

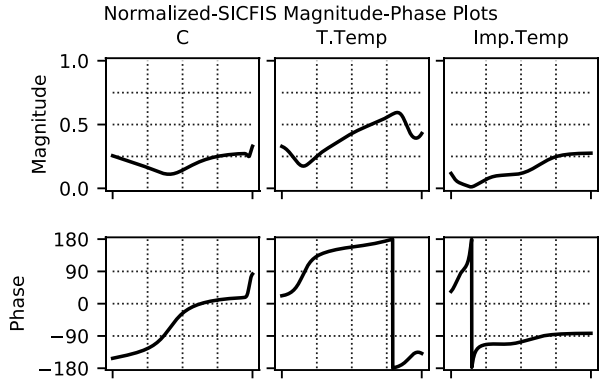


Figure 6.14: Magnitude-Phase plots for the Normalized-SICFIS model for Carbon, Tempering Temperature (T.Temp) and Impact Temperature (Imp. Temp).

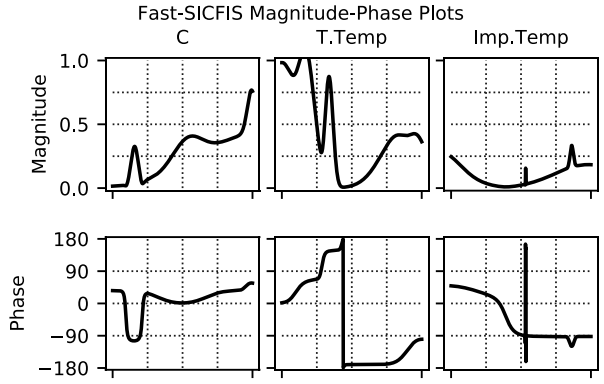


Figure 6.15: Magnitude-Phase plots for the Fast-SICFIS model for Carbon, Tempering Temperature (T.Temp) and Impact Temperature (Imp. Temp).

6.6 Summary

This chapter presented the development of a linguistic and interpretable complex Gaussian membership function following the indications presented in [8], [55], [132]. The complex fuzzy Gaussian membership function was implemented to develop linguistic complex single input FIS. The system is equivalent to a type-1 single input Mamdani FIS when all the phases are equal to zero, furthermore this equivalence extent

to cases in which all the phases in the system are aligned, that is, when no interference occurs in the system.

The results obtained from the SICFIS -Mamdani model are comparable with other FIS systems such as the RBFN and the ANFIS models. The results did not outperform the singleton-SICFIS model. These results are consistent with type-1 Mamdani FIS, which are known to be less accurate than RBFN and TSK FISs. The reduced accuracy can be compensated with an increase in the interpretability of the model.

Chapter 7

Feature Selection Algorithm with Fuzzy Rough Sets and the Single Input Complex Fuzzy Inference System

7.1 Introduction

Feature selection algorithms have become increasingly important in machine learning and AI given the ever-expanding size of databases created for industrial and commercial applications [133]. Feature selection algorithms can be used to create smaller datasets composed of the features that have the most impact in the prediction accuracy, producing better and more compact models by removing unimportant and uncorrelated information. Other advantages of feature selection algorithm are the assessment of the impact a feature has in increasing the prediction accuracy of a model.

Assessing the importance of a feature is of importance in fields such as medicine and engineering. In medicine for example, it is crucial to identify symptoms for the proper diagnosis of diseases [134], [135]. In material engineering it is important to identify process and alloys that have the most impact in the material properties in order to allocate properly the resources to ensure right first-time production.

Feature selection algorithm can be classified in three major categories. Filter, wrappers and embedded methods [136]. Filter methods are used during data pre-processing. Wrapper methods select a subset of a features based on their impact in the prediction accuracy of a model. Embedded methods realize the feature selection process within the algorithm and training.

The SICFIS model introduced in Chapter 4 presents novel methods for interpreting and extracting knowledge. The SICFIS model maps real-valued inputs into the complex domain, representing the relationship between input and output variables as interferences. The magnitude-phase plots introduced in section 4.4.3 display the behaviour of the system given any combination of inputs within a range of operation. A filter method utilizing complex-valued statistics and the information extracted from the magnitude-phase plots is devised and implemented in four real-world datasets utilized in this work.

For comparison purposes a wrapper method utilizing the SICFIS model and a filter/wrapper method utilizing fuzzy rough sets are to be implemented in Charpy, TS and Bladder cancer datasets previously studied, in order to compare the performance of the SICFIS filter. For the superconductivity dataset, a result comparison is presented from the results presented in [109].

7.2 Wrapper Method Utilizing the SICFIS Model

Wrapper methods select a subset of features based on the impact these features have on the prediction accuracy. Wrapper methods are "model agnostic" meaning that any model can be selected, including simple linear models or more complex machine learning models such as ANN. Wrapper methods can be considered "brute force" as it requires to compute a large number of models to derive a proper subset of features. These methods become intractable as the dimension of the dataset increases, given that the number of models needed to evaluate grows exponentially. To reduce the size of the grid search, it is possible to implement Greedy search strategies. Greedy search strategies can be either forward selection or backward elimination [136].

In a forward selection algorithm, the algorithm begins by selecting each of the feature of the sets of available features. The performance of each feature is assessed

and compared, the best performing feature is added to a subset of features, once added to this subset, it will be part of the remaining iterations of the algorithm. This process is repeated until an end condition is met, such as an optimal number of features are selected, or no features are left to be tested, the forward selection algorithm is shown in Algorithm 7.2. The backward elimination algorithm works opposite, eliminating the worst performing feature at each iteration, the backward elimination algorithm is shown in Algorithm 7.1. The order in which the features are eliminated or added to the algorithm can serve as a measurement of their impact on the prediction [136].

Algorithm 7.1: Backward elimination algorithm.

Inputs: Set of all possible features $Features = \{p_1, p_2, ..., p_{1-P}, p_P\}$
Output: Set *A* composed of subsets of best features at each iteration

$A_1 = Features$
For $j = 2:|P|-1$
    For $k = 1:|A_{j-1}|$
        $B_k = A_{j-1} \setminus \{a_k\}$
        Calculate performance $f(B_k)$
    End
    $A_j = B_k$ *Best Performance*
End

Algorithm 7.2: Forward selection algorithm.

Inputs: Set of all possible features $P = \{p_1, p_2, ..., p_{1-P}, p_P\}$
Output: Set *A* composed of subsets of best features at each iteration

$A_1 = \varnothing$
For $j = 2:|P|-1$
    For $k = 1:|A_{j-1}|$
        $B_k = A_{j-1} \cup \{a_k\}$
        Calculate performance $f(B_k)$
    End
    $A_j = B_k$ *Best Performance*
End

In this section, a backward elimination algorithm is to be developed utilizing the fast-SICFIS model, given the low computation required to train the fast-SICFIS algorithm it is ideal for such brute-force algorithms.

7.2.1 Results Wrapper Method Utilizing Fast-SICFIS Model

The results of the first three real-world datasets are summarized in Table 7.1. The order in which features are eliminated is shown in descending order, showing at the last row of each column the last remaining feature, which can be considered as the most important feature for prediction accuracy. To assess the performance of the feature selection algorithm, *P-1* models are to be trained and evaluated (*P* being the number of features in a dataset), each with a decreasing number of features according to the results obtained and shown in Table 7.1. Ideally, a slight decrease in performance should be observed, a sharp decrease in performance would indicate an improper elimination of a feature. Results for the Charpy, UTS and Bladder Cancer datasets are shown in Figure 7.1, Figure 7.2 and Figure 7.3 respectively

Table 7.1: SICFIS Wrapper method for feature selection results.

	Charpy		UTS		Cancer
1	Test Depth	1	S	1	CIS Present
2	Cooling Medium	2	Al	2	Squamous
3	Al	3	Hardening Temperature	3	Muscle
4	Mo	4	Mn	4	Cystectomy
5	V	5	Si	5	Grade
6	Ni	6	V	6	Sex
7	S	7	Test Depth	7	Urothelium
8	Si	8	Site	8	Radiotherapy
9	Cr	9	Ni	9	Nodes Details
10	Mn	10	Cooling Medium	10	Vascular
11	Hardening Temperature	11	Cr	11	SPB
12	Site	12	Size	12	Age
13	C	13	C	Final	Stage
14	Impact Temperature	14	Mo		
15	Size	Final	Tempering Temperature		
Final	Tempering Temperature				

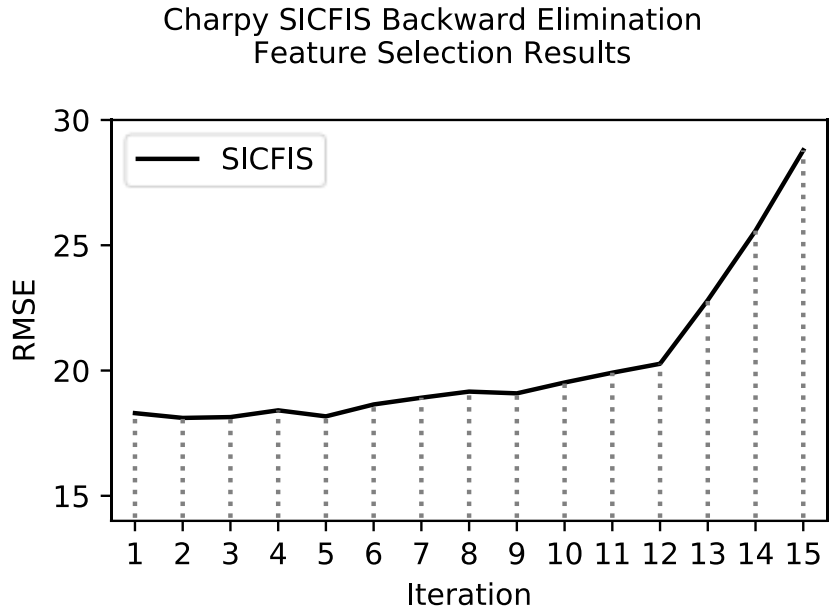


Figure 7.1: Charpy Impact Test SICFIS Backward elimination feature selection results.

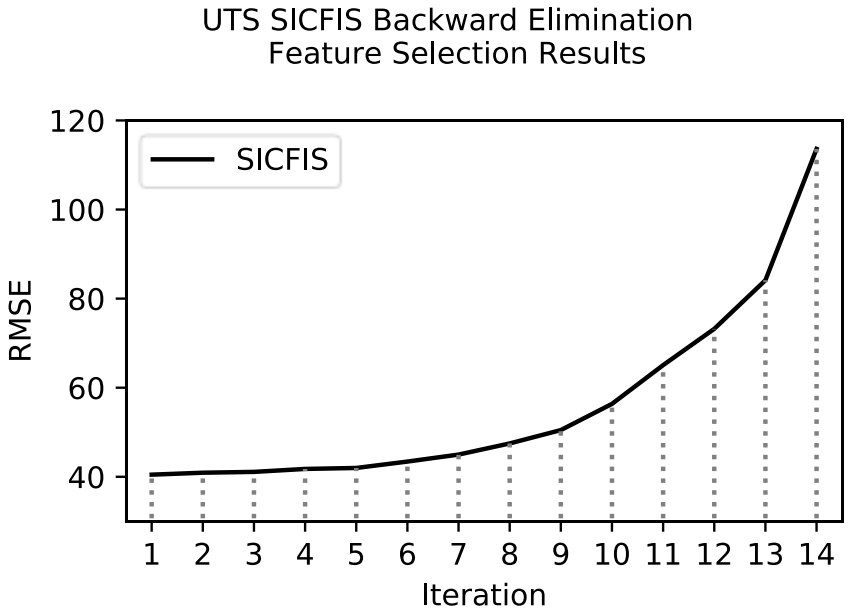


Figure 7.2: UTS SICFIS Backward elimination feature selection results.

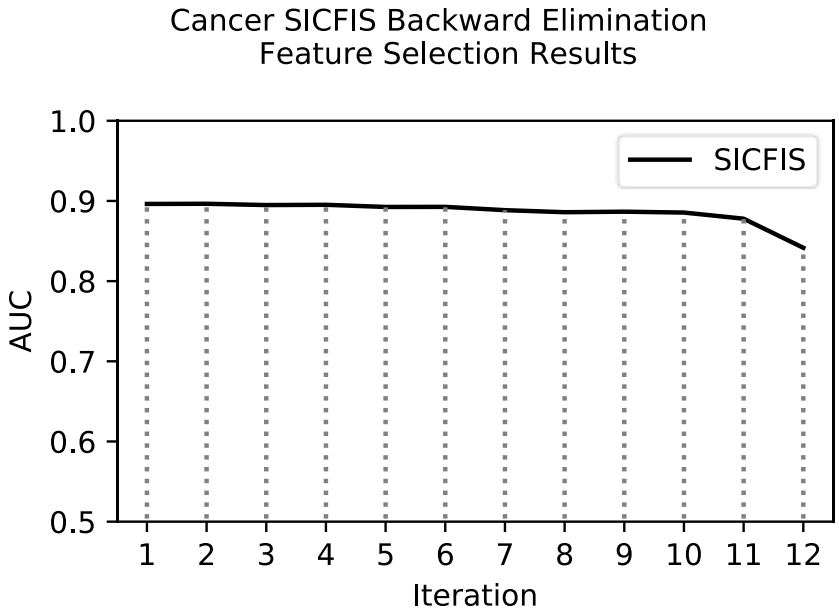


Figure 7.3: Bladder Cancer SICFIS Backward elimination feature selection results.

The backward elimination algorithm is designed specifically to obtain the best results given such evaluation, therefore the wrapper-SICFIS method will serve as a benchmark score for the remining models.

7.3 Filter Method Utilizing Fuzzy Rough Sets

Rough sets and fuzzy rough sets can be utilized to measure the dependency between features and output variables. The rough set feature dependency is a measure of how accurately a set of features can describe the output. An information table filled with irrelevant and/or random features would score a low dependency value. The method described in this section for feature selection may be classified as a filter/wrapper method, given that it is necessary to implement "brute-force" algorithms to measure the feature dependency of different combination of features. Methods such as particle swarm optimization [46] and a forward selection algorithm [49], [50] have been

implemented successfully. As is the case in the previous section, a backward elimination greedy search algorithm is to be implemented, utilizing the fuzzy-rough feature dependency as a criterion for eliminating poor performing features.

A major disadvantage of utilizing fuzzy rough sets methods for feature selection is the exponential growth of computational time with the addition of features and the number of instances in the dataset [52], the implementation of parallel computing operations reduces considerably the computation time for larger data-sets, nonetheless memory problems may arise for "big data" applications.

In section 2.5 rough sets and fuzzy rough sets were introduced. The method for calculating the fuzzy roughs sets, positive region and feature dependency utilized in this work is the same as the one introduced by Etienne and Kerre in [48] and further developed by Jensen and Shen in [49]. Three different fuzzy similarity relationship equations utilized to calculate fuzzy-rough sets were presented, the three equations are presented again for clarification below:

$$\mu_{R_p}(x,y) = 1 - \frac{|p(x) - p(y)|}{|p_{\max} - p_{\min}|} \tag{7.1}$$

$$\mu_{R_p}(x,y) = \exp\left(-\frac{(p(x) - p(y))^2}{2\sigma_p^2}\right) \tag{7.2}$$

$$\mu_{R_p}(x,y) = \max\left(\min\left(\frac{p(y) - \left(p(x) - \sigma_p\right)}{p(x) - \left(p(x) - \sigma_p\right)}, \frac{\left(p(x) + \sigma_p\right) + p(y)}{\left(p(x) + \sigma_p\right) + p(x)}\right)\right) \tag{7.3}$$

The positive region and feature dependency of a fuzzy-rough sets are calculated as follows:

$$\mu_{POS_{R_p}(Q)}(X) = \sup_{X \in U/Q} \mu_{\underline{R_p} X}(x) \tag{7.4}$$

$$\gamma_p^{\mu}(Q) = \frac{\sum_{x \in U} \mu_{POS_{R_p}(Q)}(x)}{|U|} \tag{7.5}$$

Each of the three equations will be implemented and the performance is to be compared in order to identify the best fuzzy similarity relationship equation for measuring feature dependency. The equations (7.2), (7.1) and (7.3) will be referred to as *fuzzy similarity -1, fuzzy similarity -2* and *fuzzy similarity -3* respectively.

7.3.1 Results

The order in which features are eliminated at each iteration for the Charpy, UTS, and Bladder Cancer datasets are shown in Table 7.2, Table 7.3 and Table 7.4 respectively. The performance evaluation method utilized with the wrapper-SICFIS method is implemented and the results for the Charpy, UTS and Bladder Cancer datasets are shown in Figure 7.4, Figure 7.5 and Figure 7.6 respectively. It is seen that the performance of Fuzzy similarity -1 is superior as compared with Fuzzy similarity -2 and Fuzzy similarity.

Table 7.2: Fuzzy Rough set feature selection Charpy dataset variables eliminated at each iteration

	Feature Eliminated		
Iteration	Fuzzy Similarity - 1	Fuzzy Similarity - 2	Fuzzy Similarity - 3
1	V	V	Mo
2	Ni	Ni	V
3	Cr	Cooling Medium	Ni
4	Mo	Mn	Cooling Medium
5	Mn	Cr	C
6	Hardening Temperature	Mo	Site
7	Cooling Medium	Hardening Temperature	Mn
8	Test Depth	Test Depth	Cr
9	S	S	Test Depth
10	Al	Site	Hardening Temperature
11	Site	Impact Temperature	Impact Temperature
12	Si	Si	Si
13	Impact Temperature	Al	S
14	Size	Size	Al
15	Tempering Temperature	Tempering Temperature	Size
Final	C	C	Tempering Temperature

Table 7.3: Fuzzy Rough set feature selection UTS dataset variables eliminated at each iteration

	Feature Eliminated		
Iteration	Fuzzy Similarity - 1	Fuzzy Similarity - 2	Fuzzy Similarity - 3
1	V	V	V
2	Al	Al	Cr
3	Test Depth	Cr	Al
4	Ni	Test Depth	Ni
5	Mn	Mn	Cooling Medium
6	Cooling Medium	Cooling Medium	Test Depth
7	Site	Site	Mn
8	S	Ni	Site
9	Cr	Hardening Temperature	Hardening Temperature
10	Hardening Temperature	S	S
11	Si	Si	Si
12	Size	Size	Mo
13	C	C	C
14	Mo	Mo	Size
Final	Tempering Temperature	Tempering Temperature	Tempering Temperature

Table 7.4: Fuzzy Rough Sets feature selection Cancer dataset features eliminated at each iteration

	Variable Eliminated		
Iteration	Fuzzy Similarity - 1	Fuzzy Similarity - 2	Fuzzy Similarity - 3
1	Cystectomy	Cystectomy	Cystectomy
2	Radiotherapy	Radiotherapy	Radiotherapy
3	Nodes Detail	Nodes Detail	Nodes Detail
4	Squamous	Squamous	Squamous
5	CIS Present	CIS Present	CIS Present
6	Vascular	Vascular	Vascular
7	SPB	SPB	SPB
8	Urothelium	Urothelium	Urothelium
9	Grade	Grade	Grade
10	Muscle	Muscle	Muscle
11	Sex	Sex	Sex
12	Age	Age	Stage
Final	Stage	Stage	Age

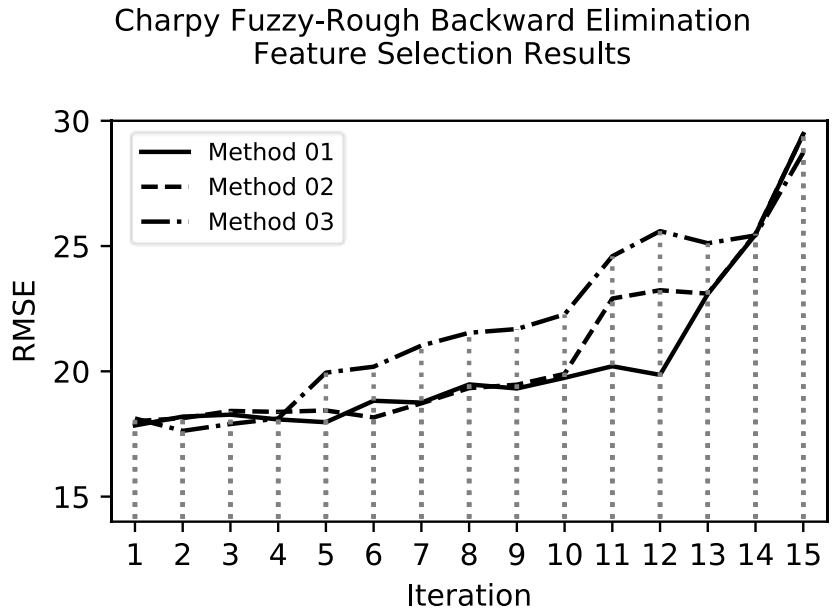


Figure 7.4: Charpy Fuzzy-rough sets Backward elimination feature selection results.

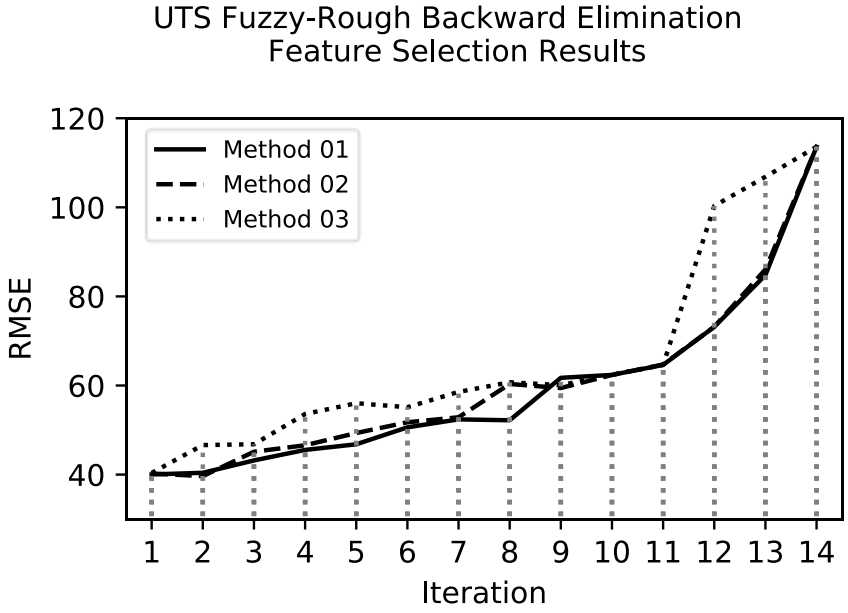


Figure 7.5: UTS Fuzzy-rough sets Backward elimination feature selection results.

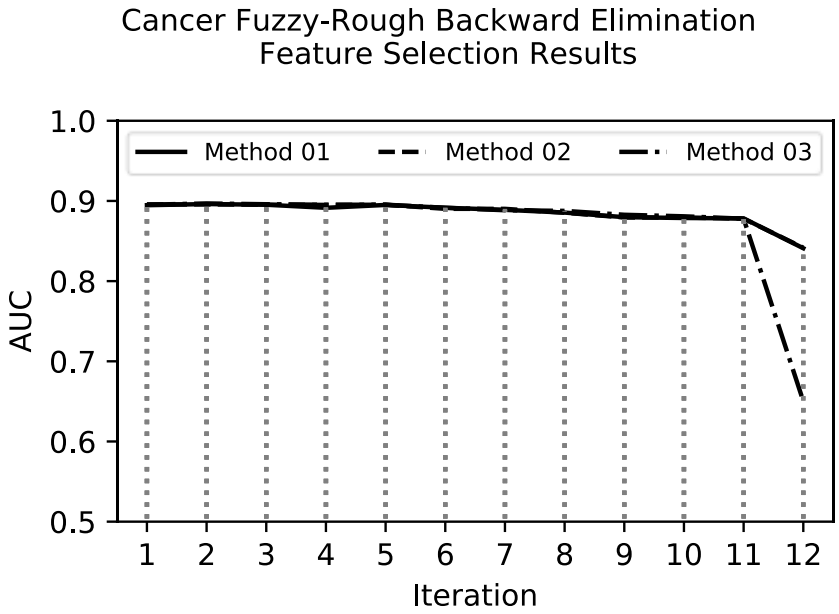


Figure 7.6:Bladder Cancer Fuzzy-rough sets Backward elimination feature selection results.

7.4 SICFIS Filter Feature Selection Algorithm.

The SICFIS model introduced in section Chapter 4 maps real-valued inputs to the complex domain, this allows to model the interaction between features as interferences. This process can be represented utilizing the magnitude-phase information of each feature (section 4.4.2.1) to model the behaviour of the system given any input within the range of operation. The magnitude and phase information for a feature *p* given an input k are as follows:

$$Mag_p = \left| \sum_{s_p=1}^{S_p} \left(\bar{\mu}_{p,s_p}(k_p) + \cos(\varphi_{p,s_p}) \cdot \beta_{p,s_p} \right) + \sum_{s_p=1}^{S_p} \left(\bar{\mu}_{p,s_p}(k_p) + \sin(\varphi_{p,s_p}) \cdot \beta_{p,s_p} \right) i \right| \quad (7.6)$$

$$Ph_p = \arg\left(\sum_{s_p=1}^{S_p} \left(\bar{\mu}_{p,s_p}(k_p) + \cos(\varphi_{p,s_p}) \cdot \beta_{p,s_p} \right) + \sum_{s_p=1}^{S_p} \left(\bar{\mu}_{p,s_p}(k_p) + \sin(\varphi_{p,s_p}) \cdot \beta_{p,s_p} \right) i \right) \quad (7.7)$$

where *k* is a continuous variable with strictly increasing values within the range of operation of a feature *p*.

Given that the entire behaviour of the system is represented with the magnitude-phase plots, it is possible to estimate which are the most important features in the system. For example, below in Figure 7.7 are shown the magnitude-phase plots for the Charpy impact test features, utilizing 3 membership function per feature. In Figure 7.8 the complex-valued output prediction when fixing all the features to a specific value and varying each one of the following features Carbon, Sulphur, Nickel and tempering temperature is shown.

From the results shown in Figure 7.8 the feature "tempering temperature" produces the highest complex-valued variance, followed by Carbon, while Nickel and Sulphur hardly produce any variance in the complex valued output.

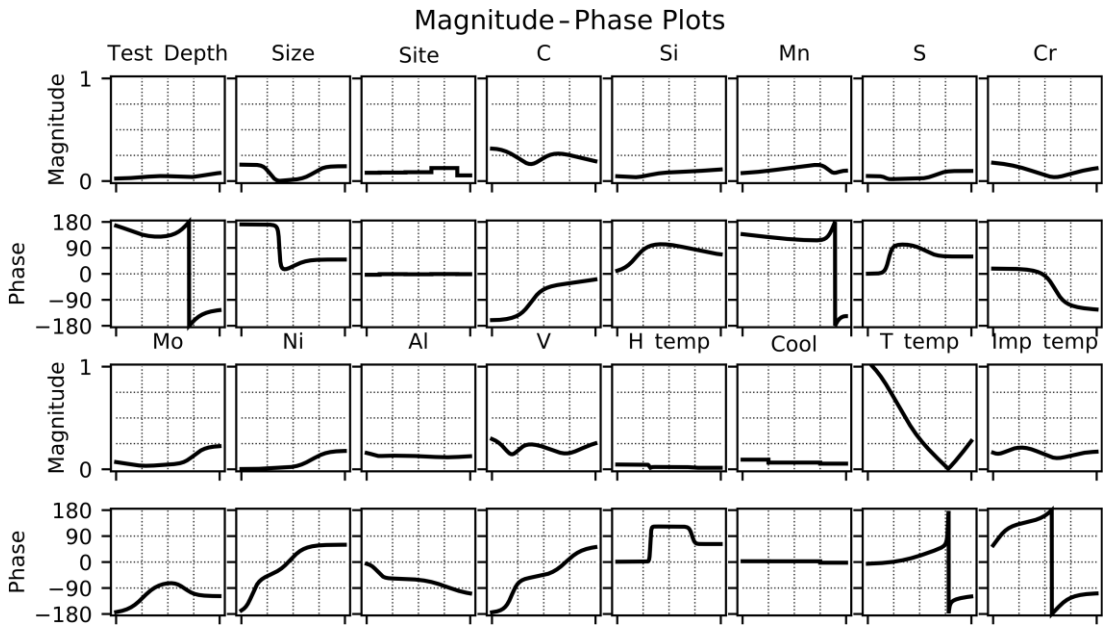


Figure 7.7: Charpy Impact Magnitude Phase Plots.

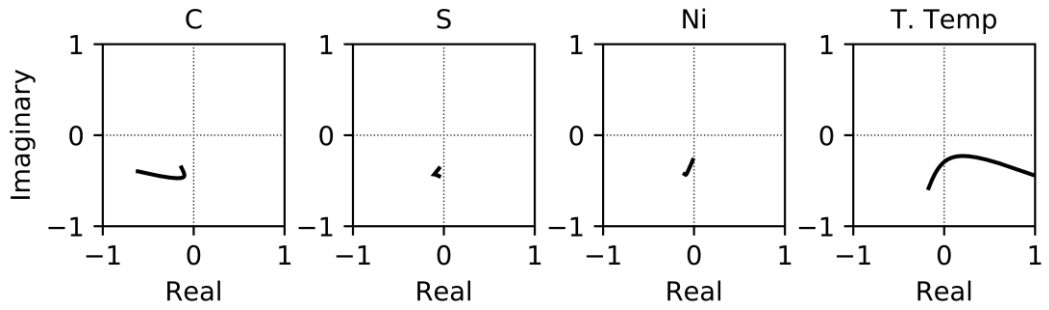


Figure 7.8: Charpy Impact normalized complex-valued output prediction varying: Carbon (C), Sulphur (S), Nickel (Ni) and tempering temperature (T. Temp).

Given the example, two different feature importance measurement methods may be implemented. The first method takes into consideration the variables that produce the greater variance in the output, these variables are: the magnitude of the resultant vector of a feature and the rate of change of its magnitude and phase. The second method measures the complex-valued covariance between a complex-value feature and the predicted output.

7.4.1 Feature Importance Score Based on a Features Magnitude and Rate of Change.

A feature importance score based on a features magnitude and rate of change may be calculated utilizing the magnitude-phase plots. One may calculate the area under the curve of the magnitude and the area under the curves of the magnitude and phase rate of change. This method presents several challenges: The first challenge arises from the datasets itself. Such method would be appropriate only for datasets containing continuous features with a uniform distribution. For example, the Charpy impact test is known for its scattered measurements, (the histogram plots of each of the features is shown in Figure 7.9, additionally the Bladder cancer dataset contains mostly categorical features.

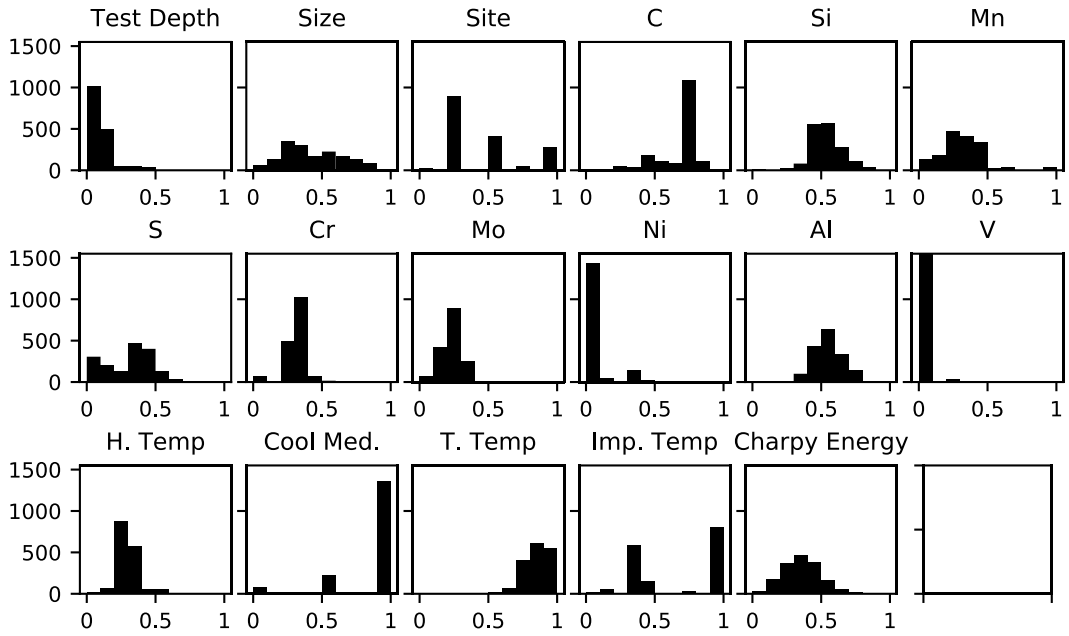


Figure 7.9: Charpy impact test feature histogram.

Therefore, in order to develop an appropriate feature importance score formula, it is necessary to utilize the magnitude and the phase information of the instances x_p^n in the dataset. The magnitude, phase are as follows:

$$Mag_p^n = \left| \sum_{s_p=1}^{S_p} \left(\bar{\mu}_{p,s_p}(x_p^n) + \cos(\varphi_{p,s_p}) \cdot \beta_{p,s_p} \right) + \sum_{s_p=1}^{S_p} \left(\bar{\mu}_{p,s_p}(x_p^n) + \sin(\varphi_{p,s_p}) \cdot \beta_{p,s_p} \right) i \right| \quad (7.8)$$

$$Ph_p^n = \arg\left(\sum_{s_p=1}^{S_p} \left(\bar{\mu}_{p,s_p}(x_p^n) + \cos(\varphi_{p,s_p}) \cdot \beta_{p,s_p} \right) + \sum_{s_p=1}^{S_p} \left(\bar{\mu}_{p,s_p}(x_p^n) + \sin(\varphi_{p,s_p}) \cdot \beta_{p,s_p} \right) i \right) \quad (7.9)$$

The resulting complex-valued variable for each feature instance is as follows:

$$z_p^n = \sum_{s_p=1}^{S_p} \left(\bar{\mu}_{p,s_p}(x_p^n) + \cos(\varphi_{p,s_p}) \cdot \beta_{p,s_p} \right) + \sum_{s_p=1}^{S_p} \left(\bar{\mu}_{p,s_p}(x_p^n) + \sin(\varphi_{p,s_p}) \cdot \beta_{p,s_p} \right) i \quad (7.10)$$

where $n:[1\cdots N]$ and N represent the number of instances in the dataset.

The following formula replaces the area under the curve of the magnitude with the expected value of the magnitude, and the area under the curves of the rate of change of the magnitude and the phase with the variance function.

$$FeatureScore_p^{Mag-Ph} = E[\mathbf{Mag}_p]\cdot\left(\frac{\text{var}(\mathbf{Mag}_p)}{\sum_{p=1}^{P}\text{var}(\mathbf{Mag}_p)}+\frac{\text{var}(Ph_p)}{\sum_{p=1}^{P}\text{var}(Ph_p)}\right) \tag{7.11}$$

The calculation of the expected value and variance of the magnitude is straight forward. Calculating the variance of the phase requires some modifications to the variance equation. The variance is calculated as the expected value of the squared distances between the mean and the samples. Given that the angular values are circular, it is more appropriate to calculate the angular distance between the mean value of the complex random variable *z* as follows:

$$\text{var}(Ph_p) = E\left[\left(\cos^{-1}\left(\frac{E[\mathbf{z}_p]\cdot\mathbf{z}_p}{\|E[\mathbf{z}_p]\|\|\mathbf{z}_p\|}\right)\Big/\pi\right)^2\right] \tag{7.12}$$

The expected value of a complex random variable is calculated as follows [137]:

$$E[\mathbf{z}] = E(\mathbf{z}_{\text{Re}}+\mathbf{z}_{\text{Im}}i) = E(\mathbf{z}_{\text{Re}})+E(\mathbf{z}_{\text{Im}}i) \tag{7.13}$$

Given that the magnitude an the angular distance utilize different measurements, each of the variables in (7.11) are normalized to give a proportional weight to each of the variables.

7.4.2 Covariance of Complex-Valued Random Variables

In the previous section t was shown that the SICFIS model maps real-valued features inputs into the complex domain, the variance and covariance of two complex-valued random variables is as follows [137]:

$$\sigma_x^2 = Var(\mathbf{z}) \triangleq E\left[(\mathbf{z} - E[\mathbf{z}])(\mathbf{z} - E[\mathbf{z}])^*\right] \tag{7.14}$$

$$cov(\mathbf{z}_1\mathbf{z}_2) = E[\mathbf{z}_1\mathbf{z}_2^*] - E[\mathbf{z}_1]E[\mathbf{z}_2]^* \tag{7.15}$$

where * represents the complex conjugate of the complex quantity.

In order to utilize the complex covariance as a feature importance measure, it is necessary to calculate its magnitude, the larger the magnitude, the larger the covariance between a feature and the output.

$$FeatureScore_p^{Cov} = \left|cov(\mathbf{z}_p\mathbf{z}_{output})\right| \tag{7.16}$$

7.4.3 Combined Feature Importance Equation

Finally, it is possible to combine both equations into a single feature importance equation as follows:

$$FeatureScore_p^{Combined} = \frac{FeatureScore_p^{Mag-Ph}}{\sum_{p=1}^{P} FeatureScore_p^{Mag-Ph}} + \frac{FeatureScore_p^{Cov}}{\sum_{p=1}^{P} FeatureScore_p^{Cov}} \tag{7.17}$$

Equation (7.17) utilizes both measurements, in order to provide more robust predictions, each one normalized to provide an adequate weight to each of the feature score equations. The three equations, (7.11), (7.16) and (7.17) will be evaluated in the following section to provide more insight to which feature selection equation provides best results.

7.4.4 Results

The feature importance measurements relies entirely on producing a properly trained SICFIS model, therefore in order to increase robustness in the feature score measurement, *K* models are to be trained and evaluated, utilizing the same training data partition, the initial values will be randomly modified to present different results. The final feature importance equation is as follows:

$$FeatureScore_p = \frac{\sum_{k=1}^{K} FeatureScore_{p,k}}{K} \tag{7.18}$$

Each of the datasets will be evaluated utilizing the three feature score equations (7.11), (7.16) and (7.17). Both the normalized and fast SICFIS models will be evaluated utilizing the same method explained in the previous sections.

Results of the Charpy impact test for the normalized and fast SICFIS models are shown in Table 7.5 and Table 7.6. The evaluation of each of the equations and both the normalized and fast models is shown in Figure 7.13. Results obtained by the normalized-SICFIS model are superior to that of the fast-SICFIS model, given the obvious elimination of tempering feature. For both models the combined equation performed slightly better than the Mag-Phase equation. The worse performing equation was the covariance equation for both models.

Table 7.5: Charpy Normalized-SICFIS filter method for feature selection results.

	Combined	Score	Mag-Phase	Score	Covariance	Score
1	Si	0.0220	Ni	0.0172	Al	0.0172
2	Al	0.0224	Si	0.0210	Si	0.0251
3	H.Temp	0.0401	Al	0.0258	H.Temp	0.0399
4	Ni	0.0632	H.Temp	0.0397	S	0.0411
5	Depth	0.0665	Depth	0.0630	Depth	0.0719
6	Site	0.0733	Site	0.0672	Site	0.0761
7	S	0.0798	Cool. Med.	0.0875	Ni	0.1032
8	Cool. Med.	0.1156	S	0.1061	Mn	0.1246
9	Cr	0.1344	Cr	0.1235	Cr	0.1334
10	Mn	0.1352	Mn	0.1324	Cool. Med.	0.1355
11	V	0.2375	V	0.1531	V	0.2975
12	Mo	0.3619	Mo	0.2907	Imp. Temp.	0.3017
13	C	0.3941	C	0.3049	Mo	0.4168
14	Imp. Temp.	0.5503	Size	0.6329	C	0.4500
15	Size	0.5698	Imp. Temp.	0.7339	Size	0.4648
Final	T. Temp.	1.0000	T. Temp	0.9147	T. Temp	1.0000

Table 7.6: Charpy Fast-SICFIS filter method for feature selection results.

	Combined	Score	Mag-Phase	Score	Covariance	Score
1	Al	0.0075	Al	0.0173	Al	0.0097
2	Ni	0.0177	Ni	0.0184	Si	0.0218
3	Si	0.0199	Si	0.0284	Ni	0.0269
4	Site	0.0854	Site	0.0933	Site	0.0834
5	Mo	0.1129	Cool. Med.	0.0937	Mo	0.1014
6	Cool. Med.	0.1320	H.Temp	0.1181	S	0.1372
7	S	0.1361	Mo	0.1293	V	0.1444
8	Depth	0.2006	S	0.1399	Depth	0.1577
9	V	0.2141	Cr	0.1958	Cool. Med.	0.1739
10	H_temp	0.2170	Mn	0.2412	Mn	0.2881
11	Mn	0.2674	Depth	0.2440	H.Temp	0.3148
12	Cr	0.3156	V	0.2835	Size	0.4043
13	Size	0.3700	Size	0.3282	Cr	0.4297
14	T. Temp.	0.5523	T. Temp	0.5264	T. Temp	0.5677
15	C	0.7719	C	0.8556	C	0.6603
Final	Imp. Temp.	0.9713	Imp. Temp.	0.9303	Imp. Temp.	0.9799

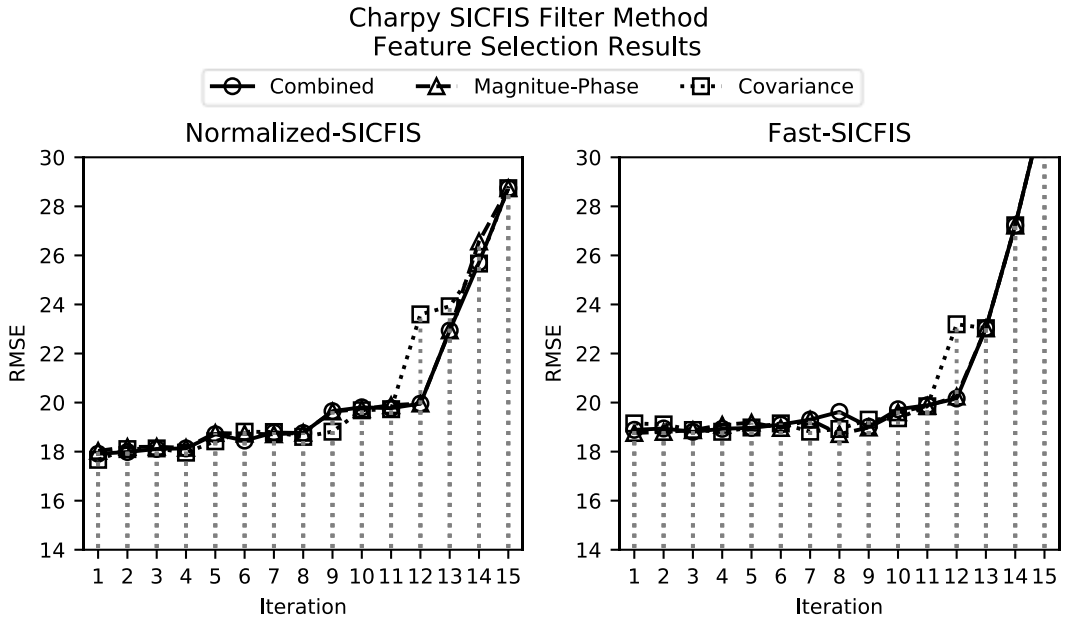


Figure 7.10: Charpy SICFIS-Filter feature selection results.

Results of the UTS for the normalized and fast SICFIS models are shown in Table 7.7 and Table 7.8. The evaluation of each of the equations for both the normalized and fast models is shown in Figure 7.14. The best results are obtained by the combined equation for the fast-SICFIS model. The results from the Mag-Phase and the Covariance equation seem to vary between different points, showing a clear advantage of utilizing both equation for obtaining better and more robust results.

Results of the Cancer for the normalized and fast SICFIS models are shown in Table 7.9 and Table 7.10. The evaluation of each of the equations for both the normalized and fast models is shown in Figure 7.12 Figure 7.15. In section 4.7.3, the results for the Cancer dataset utilizing the normalized and fast-SICFIS models showed a clear difference between both methods, being the fast-SICFIS model better suited for modelling the Cancer dataset, therefore a poor performance of the normalized-SICFIS model for feature selection is the results of its poor performance in prediction. The

combined equation provided the best results as shown in Figure 7.12. From the results observed it is concluded that after Stage, Age is the most important feature for prediction.

Table 7.7: UTS Normalized-SICFIS filter method for feature selection results.

	Combined	Score	Mag-Phase	Score	Covariance	Score
1	Si	0.00172	Si	0.00128	Si	0.00320
2	Al	0.00426	Al	0.00394	Al	0.00571
3	Depth	0.00777	Depth	0.00776	Depth	0.00860
4	H. Temp	0.01227	H. Temp	0.01289	H. Temp	0.01216
5	V	0.02945	Site	0.02332	V	0.02991
6	S	0.04061	V	0.02854	S	0.04336
7	Site	0.04424	S	0.03700	Site	0.06466
8	Mn	0.06885	Mn	0.04977	Mn	0.08434
9	Size	0.12031	Cool. Med.	0.05208	C	0.12399
10	Cool. Med.	0.12953	Size	0.09009	Size	0.14659
11	C	0.20376	Mo	0.21006	Cr	0.18584
12	Mo	0.29607	C	0.27420	Cool. Med.	0.20420
13	Cr	0.35418	Ni	0.27670	Mo	0.36804
14	Ni	0.42908	Cr	0.49190	Ni	0.55409
Final	T. Temp	1.00000	T. Temp	0.95626	T. Temp	1.00000

Table 7.8: UTS Fast-SICFIS filter method for feature selection results.

	Combined	Score	Mag-Phase	Score	Covariance	Score
1	Al	0.00000	Al	0.00000	Al	0.00000
2	Si	0.00640	Si	0.00218	Si	0.01061
3	V	0.01621	Depth	0.00345	V	0.02894
4	Depth	0.02358	V	0.00348	Depth	0.04371
5	H. Temp	0.03047	S	0.00603	H. Temp	0.05384
6	S	0.03049	H. Temp	0.00710	S	0.05495
7	Cool. Med.	0.06226	Cool. Med.	0.01945	Mn	0.07376
8	Mn	0.07073	Size	0.02420	Cool. Med.	0.10508
9	Site	0.08729	Site	0.02525	Site	0.14934
10	Cr	0.11057	Cr	0.05424	Cr	0.16691
11	Size	0.13020	Mo	0.06677	Size	0.23621
12	C	0.16364	Mn	0.06769	C	0.25593
13	Mo	0.18410	C	0.07135	Mo	0.30143
14	Ni	0.22954	Ni	0.07570	Ni	0.38337
Final	T. Temp	1.00000	T. Temp	1.00000	T. Temp	1.00000

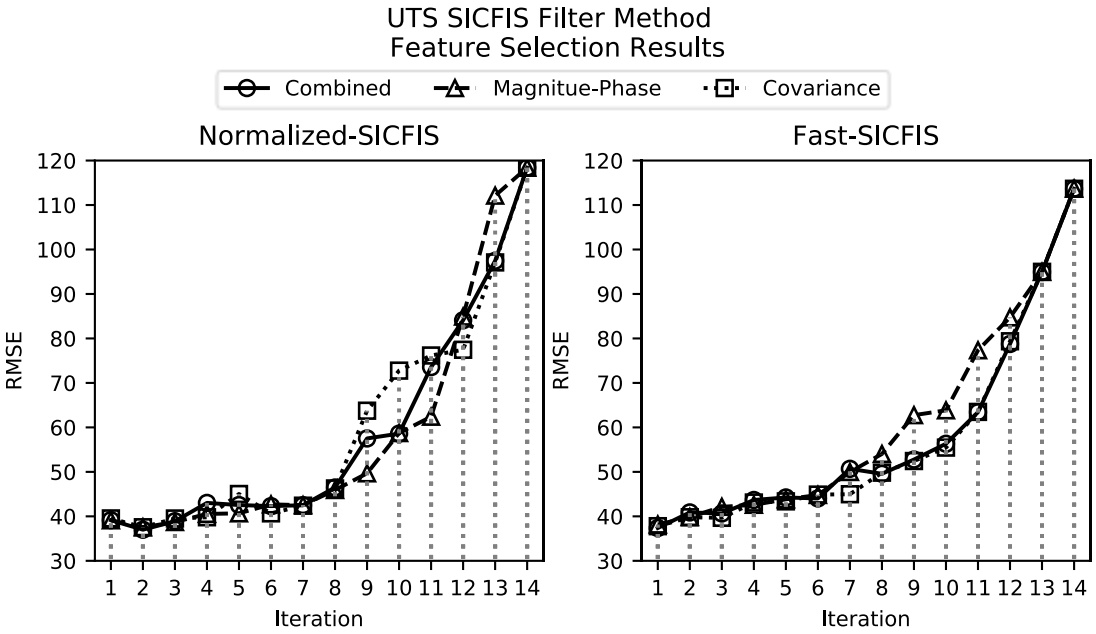


Figure 7.11: UTS SICFIS-Filter feature selection results.

Table 7.9: Bladder Cancer Normalized-SICFIS filter method for feature selection results.

	Combined	Score	Mag-Phase	Score	Covariance	Score
1	Cystectomy	0.0033	Cystectomy	0.0053	Cystectomy	0.0035
2	Vascular	0.0485	Vascular	0.0303	Radiotherapy	0.0288
3	Radiotherapy	0.0548	Radiotherapy	0.0676	Squamous	0.0305
4	Grade	0.0908	Grade	0.0962	Urothelium	0.0611
5	Urothelium	0.1078	Urothelium	0.1386	Vascular	0.0778
6	Nodes Detail	0.1166	Nodes Detail	0.1403	Nodes Detail	0.0862
7	Squamous	0.1686	Squamous	0.1438	CIS Present	0.1459
8	CIS Present	0.1974	CIS Present	0.1694	Muscle	0.1460
9	Muscle	0.2139	Muscle	0.2226	Sex	0.1567
10	Age	0.2584	Age	0.2601	Age	0.2416
11	Sex	0.2613	Sex	0.3349	Grade	0.3126
12	SPB	0.7550	SPB	0.8052	SPB	0.6131
Final	Stage	0.9581	Stage	0.8308	Stage	1.0000

Table 7.10: Bladder Cancer Fast-SICFIS filter method for feature selection results.

	Combined	Score	Mag-Phase	Score	Covariance	Score
1	Squamous	0.0017	Squamous	0.0021	Sex	0.0026
2	Vascular	0.0088	Vascular	0.0070	Squamous	0.0032
3	Radiotherapy	0.0089	Radiotherapy	0.0104	Radiotherapy	0.0095
4	Cystectomy	0.0104	Cystectomy	0.0125	Cystectomy	0.0104
5	Sex	0.0178	Sex	0.0171	Nodes Detail	0.0196
6	Nodes Detail	0.0243	Nodes Detail	0.0180	Vascular	0.0437
7	Grade	0.0718	Grade	0.0823	Muscle	0.0612
8	Muscle	0.1453	Muscle	0.0841	Urothelium	0.1417
9	SPB	0.1752	SPB	0.1326	CIS Present	0.1506
10	Urothelium	0.1948	Urothelium	0.1503	Grade	0.3089
11	CIS Present	0.2578	CIS Present	0.2014	Age	0.3267
12	Age	0.3222	Age	0.3193	SPB	0.3846
Final	Stage	1.0000	Stage	1.0000	Stage	1.0000

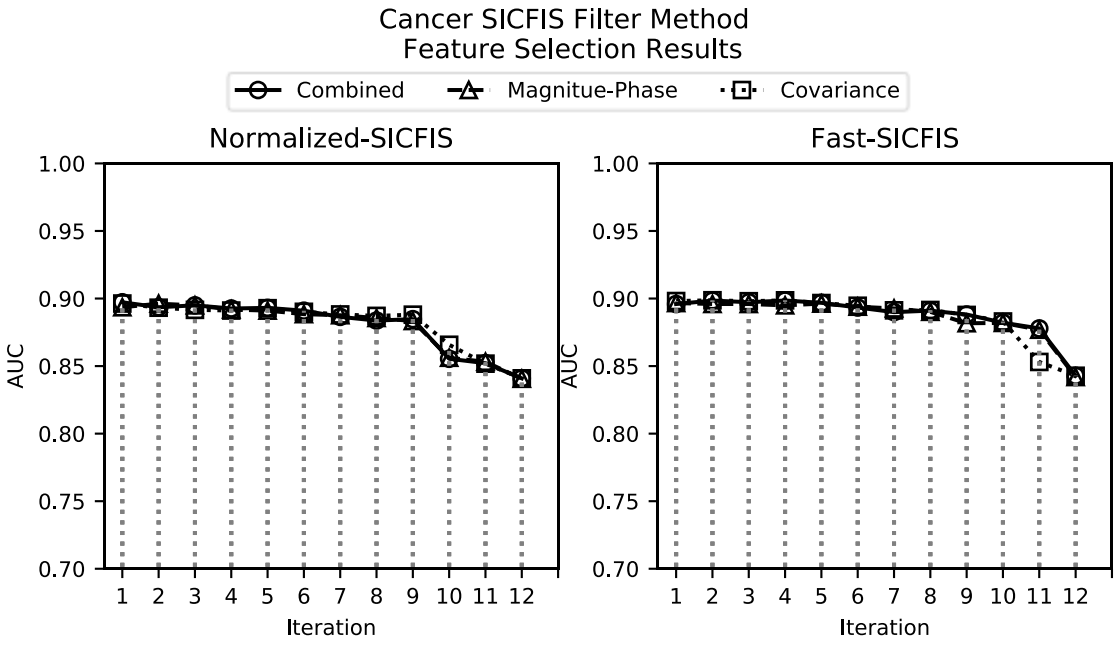


Figure 7.12: Bladder Cancer SICFIS-Filter feature selection results.

7.5 Results Comparisons

The results of the combined filter-SICFIS method for the fast and normalized SICFIS model, the wrapper-SICFIS method and the best performing Fuzzy rough set method are plotted for comparison purposes. Some variation is expected given random effects during training.

Results for the Charpy impact test are shown in Figure 7.13. The worse performing method is the filter fast-SICFIS method. While the remaining methods seem to perform equivalent and most of the difference in performance can be attributed to random errors.

The UTS results are shown in Figure 7.14. The wrapper method provided the best results, while the rest of the methods performance deviate from the wrapper method slightly at different points.

The Bladder Cancer results are shown in Figure 7.15. The worse performing model is the filter normalized-SICFIS method. This is expected, given the results observed in section 4.7.3. The rest of the results difference are attributed to random errors.

The computation time of each algorithm are shown in Table 7.11. From the computation times an exponential increase in computational times for the UTS dataset utilizing any of the fuzzy-rough set methods is observed. This exponential increase is due to the UTS dataset containing twice the number of instances in comparison with the Charpy impact dataset. For the wrapper method the number of features in the dataset has more of an impact than the number of instances. The filter SICFIS method proposed in this work produced the lowest computational time as expected, with a considerable reduction in computational times.

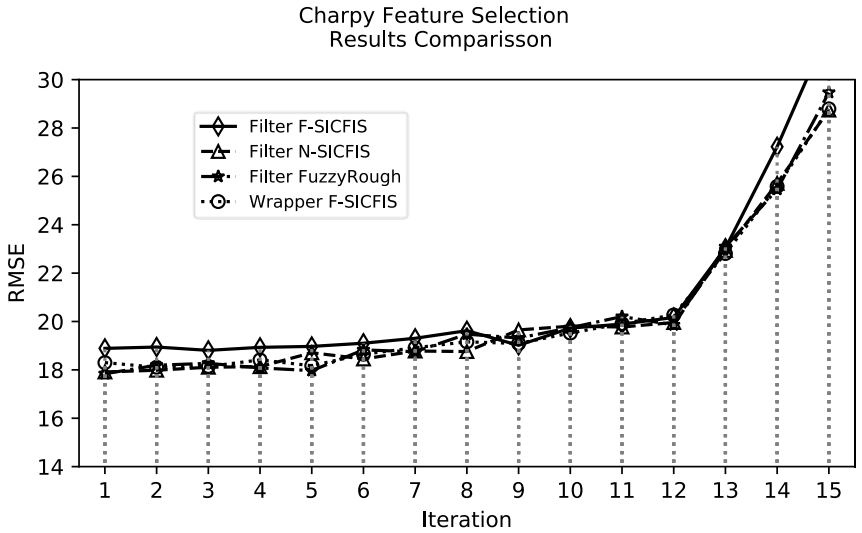


Figure 7.13: Charpy Results Comparisons between Filter-SICFIS methods, Wrapper-SICFIS and Fuzzy Rough sets

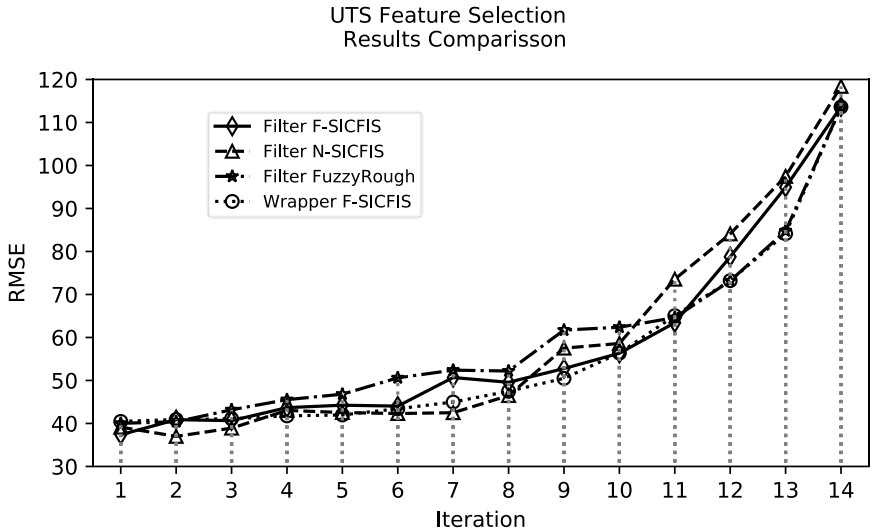


Figure 7.14: UTS Results Comparisons between Filter-SICFIS methods, Wrapper-SICFIS and Fuzzy Rough sets

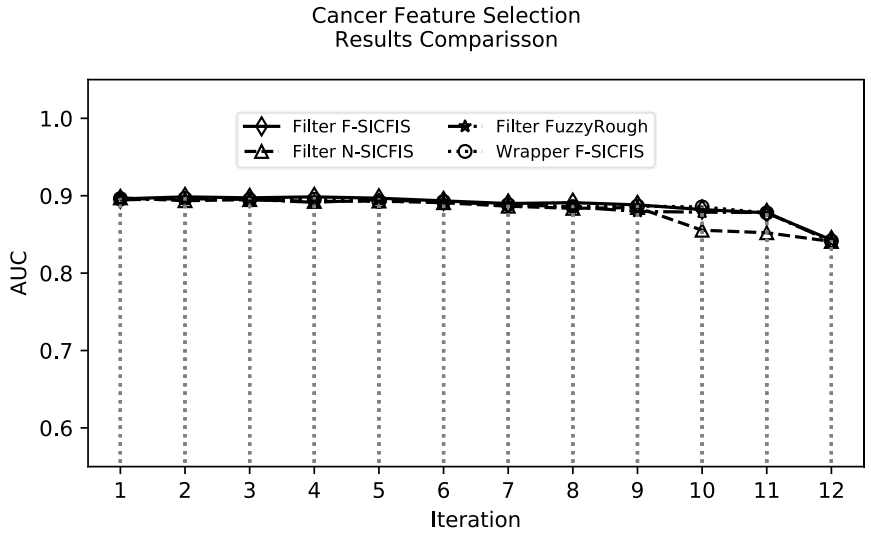


Figure 7.15: Cancer Results Comparisons between Filter-SICFIS methods, Wrapper-SICFIS and Fuzzy Rough sets

Table 7.11: Computation time comparison between the different datasets and methods measured in seconds (s).

	Charpy	UTS	Cancer
Wrapper-SICFIS	289.25 s	234.43 s	121.95 s
FRS-01	101.49 s	1012.2 s	57.34 s
FRS-02	100.78 s	975.03 s	56.07 s
FRS-03	101.35 s	978.05 s	56.34 s
Filter N-SICFIS	31.26 s	32.05 s	22.05 s
Filter F-SICFIS	17.67 s	22.81 s	13.83 s

FRS: Fuzzy Rough Set, N: Normalized, F: Fast.

7.6 Superconductivity Results

Given the large size of the superconductivity dataset, it is not possible to implement the rough -sets and wrapper feature selection methods. In [109], the authors present the 20 most significant features obtained from an XG-Boost analysis results. The results obtained from the three feature selection algorithms as well as the XG-Boost analysis are shown in Table 7.13. In order to compare the efficacy of the feature selection algorithms a reduced data set consisting of the 20 most significant features is utilized

for training a 5 membership function normalized and fast SICFIS models. The results of the evaluation are shown in Table 7.12.

Table 7.12: Superconductivity results obtained from reduced model, utilizing 5 membership functions (mF) per feature normalized(N) and fast (F) SICFIS models.

		Training	Checking	Testing	All
XG-Boost	N-SICFIS 5mF	15.33	15.65	15.40	15.40
	F-SICFIS 5mF	15.64	16.07	15.94	15.77
Combined	N-SICFIS 5mF	17.95	18.18	18.25	18.04
	F-SICFIS 5mF	17.34	17.70	17.69	17.46
Mag-Phase	N-SICFIS 5mF	17.45	16.99	17.49	17.38
	F-SICFIS 5mF	17.46	17.33	18.03	17.53
Covariance	N-SICFIS 5mF	18.73	18.49	18.21	18.60
	F-SICFIS 5mF	18.68	18.50	18.27	18.58

Table 7.13: Superconductivity feature selection results summary.

Combined		Mag-Phase		Covariance		XG-Boost [109]	
Feature	Score	Feature	Score	Feature	Score	Feature	Score
24	1.000	24	1.000	15	1.000	67	0.295
15	0.668	22	0.562	74	0.726	70	0.084
22	0.562	15	0.668	25	0.719	27	0.072
25	0.480	25	0.480	24	0.713	64	0.047
74	0.462	74	0.462	72	0.691	69	0.042
72	0.423	75	0.366	75	0.568	76	0.038
75	0.366	73	0.344	73	0.548	50	0.036
73	0.344	12	0.269	71	0.546	6	0.025
71	0.332	72	0.423	12	0.421	72	0.022
12	0.269	4	0.153	22	0.418	44	0.021
54	0.214	69	0.127	54	0.352	48	0.016
14	0.186	71	0.332	14	0.297	62	0.015
52	0.174	14	0.186	52	0.284	74	0.014
76	0.153	27	0.092	76	0.251	9	0.013
4	0.153	2	0.124	19	0.246	39	0.01
19	0.150	54	0.214	4	0.233	68	0.01
17	0.138	52	0.174	17	0.228	66	0.01
51	0.130	67	0.074	51	0.213	2	0.009
69	0.127	19	0.150	2	0.195	33	0.009
2	0.124	76	0.153	53	0.194	10	0.009

7.7 Summary

From the results obtained, the best performing algorithm in the first three datasets was the wrapper method utilizing the fast-SICFIS model. The feature selection method utilizing fuzzy rough sets with the first formula also produced comparable results, with the UTS dataset outperformed by the wrapper method. The filter-SICFIS performed comparable with the other methods, slightly decrease in performance in the UTS and cancer dataset was observed.

The advantages of the filter-SICFIS method is the possibility of assigning a score to each of the features, and the fast computation times. The fuzzy rough sets relative feature dependency can also be utilized to rank each of the features, but as observed, the computational times for both the fuzzy rough sets and the wrapper method grow exponentially with the size of the dataset.

Given that the demand for computational efficient code to deal with big-data, both the fuzzy rough set and the wrapper methods are not well equipped for large dataset such as the superconductivity dataset. The filter SICFIS-model has shown promising results for the smaller datasets but requires additional modifications for larger datasets.

Chapter 8

Fuzzy Rough Sets for Data-mining: Inconsistency Identification and Modelling

8.1 Introduction

The Charpy impact dataset is known to be difficult to model due to the scatter in the dataset and inconsistencies in the measurement values [129]. Objects in an information table are considered inconsistent when two or more objects contain the same or similar feature values but different outputs. Inconsistencies arise either by errors in measurement or by features not included in the information table. Rough sets can be utilized to identify inconsistent records and to measure the degree of inconsistency in a dataset.

This Chapter proposes an application of fuzzy rough sets for modelling under inconsistent datasets. The modelling paradigm proposes to 1. Identify and classify consistent and inconsistent instances present in the dataset utilizing fuzzy rough sets. 2. Propose a method for identifying inconsistencies in a testing partition. 3. Improve upon the results by crating different models to predict the previously identified consistent and inconsistent partitions. 4. Generate a multiple point prediction instead of single point to model inconsistencies and aid in the development of material design.

8.2 Data Inconsistency Identification

The consistency of an object can be measured by utilizing the positive region of the lower approximation of a fuzzy-rough set (8.2). The feature dependency (8.1) utilized

in the previous section can be considered as the mean measurement of consistency. In a classic rough set, a consistent object is added to its lower approximation, assigning a membership value of 1. For continuous datasets, it is necessary to implement fuzzy-rough sets. The values of the lower approximation range from 0-1, an object considered totally consistent would be assigned a membership value of 1, and a totally inconsistent object would be assigned a membership value of 0.

$$\gamma'_P(Q) = \frac{\sum_{x \in U} \mu_{POS_{RP}(Q)}(x)}{|U|} \tag{8.1}$$

$$\mu_{POS_{R_P}(Q)}(X) = \sup_{X \in U/Q} \mu_{\underline{R_P X}}(x) \tag{8.2}$$

Table 8.1 shows an example of an inconsistent information granule. The features are normalized, rounded and randomly selected for confidentiality reasons. The positive region score (8.2) shown is the last column allows to identify such information granule as inconsistent. Given that the membership value of the positive region ranges from 0 to 1 it is necessary to select a threshold value to classify objects as either consistent or inconsistent.

Table 8.1: Inconsistencies in the Charpy Impact Dataset

	Ftr 0	Ftr 1	Ftr 2	Ftr 3	Ftr 4	Ftr 5	Ftr 6	Ftr 7	Ftr 9	Output	$\mu_{POS_{R_P}(Q)}$
1	0.05	0.25	0.50	0.44	0.02	0.31	0.23	0.03	0.35	106.204	0.29
2	0.05	0.25	0.50	0.44	0.02	0.31	0.23	0.03	0.35	173.543	0.29
3	0.05	0.25	0.50	0.44	0.02	0.31	0.23	0.03	0.35	173.543	0.30
4	0.05	0.25	0.50	0.44	0.02	0.31	0.23	0.03	0.35	61.011	0.33
5	0.05	0.25	0.49	0.44	0.02	0.31	0.23	0.03	0.35	89.9347	0.33
6	0.05	0.25	0.49	0.44	0.02	0.31	0.23	0.03	0.35	86.319	0.33
7	0.05	0.25	0.49	0.44	0.02	0.31	0.23	0.03	0.35	121.118	0.40
8	0.05	0.25	0.49	0.44	0.02	0.31	0.23	0.03	0.35	101.233	0.40

Ftr: Feature.

A simple method for identifying inconsistent instances is to select the a threshold value equal to the feature dependency (8.1). Table 8.2 shows the feature dependency

value of the first three real world datasets explored in this work. It can be observed that the Cancer dataset contains by far the lowest feature dependency, followed by the Charpy impact test, while the UTS can be considered mostly consistent.

Table 8.2: Dataset Feature Dependency

	Fuzzy similarity 1	Fuzzy similarity 2	Fuzzy similarity 3
Charpy Impact	0.9612	0.9310	0.9196
UTS	0.9960	0.9883	0.9769
Cancer	0.4211	0.5299	0.6786

The low feature dependency observed in the bladder cancer dataset is related to the complex relationship and difference between different persons genetics and lifestyle, making a prediction based on a few parameters highly difficult and random [39].

8.2.1 Effects of Feature Selection in the Number of Inconsistencies and Feature Dependency

In Chapter 7 fuzzy-rough sets were implemented to develop a feature selection algorithm. This feature selection algorithm removed features based on the value of its feature dependency score. Within the context of rough sets, removing features reduces the capability of discerning between instances, increasing the number of inconsistencies in the dataset and reducing its feature dependency. Figure 8.1 shows the effect of the number of features in a dataset and the feature dependency value. The features selected for such plots where taken from the results obtained and shown in in the previous chapter in Table 7.2.

Figure 8.2 shows the effect of the number of features and a selected threshold in the number of inconsistencies. The number of inconsistencies grows significantly with the elimination of features, even when these features have a small impact in the prediction accuracy as observed in the previous chapter in Figure 7.1.

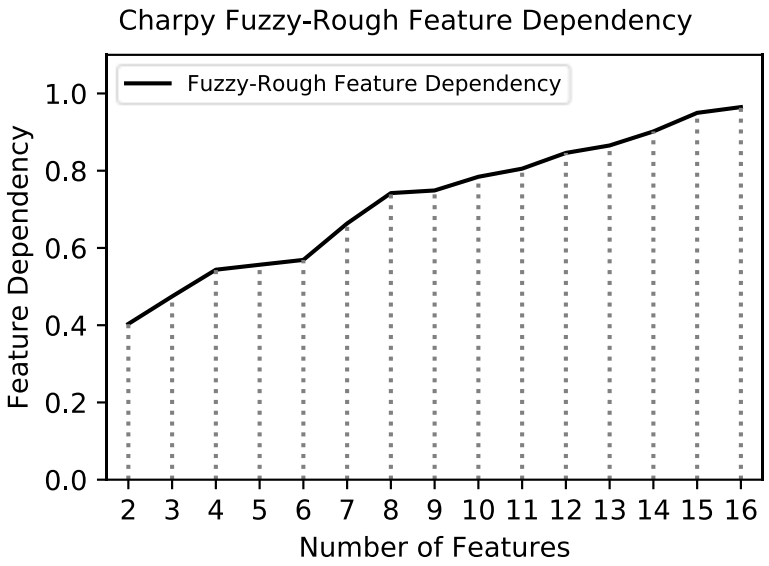


Figure 8.1: Effect of the number of features in Feature Dependency.

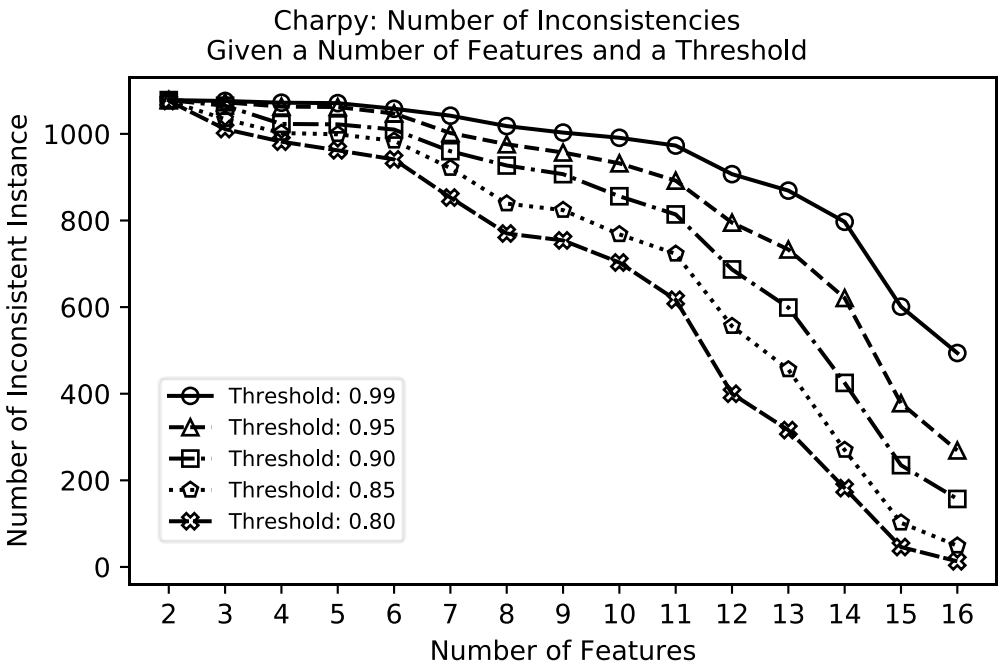


Figure 8.2: Effects on the number of inconsistencies given different number of features and different threshold values.

8.2.2 Inconsistency Identification in Testing Partition of Dataset Utilizing k-Nearest Neighbour

The identification of inconsistent instances in the dataset is performed in the training partition utilizing fuzzy-rough sets, in order to identify inconsistencies located in the testing partition it is proposed the utilization of a k-nearest neighbour (KNN) algorithm.

The KNN algorithm can be utilized for classification tasks, it classifies testing sample based on the known class values of the *k* nearest samples [138]. An example of the KNN classification is shown in Figure 8.3. Different metrics can be implemented for finding the nearest neighbour. In this work a Euclidean distance metric is implemented, a weighted method is implemented, in which nearest neighbours have more impact in the decision than further neighbours, ties are resolved by the nearest neighbour.

In order to identify am optimal number of *k* neighbours, a 10 k-fold cross validation is performed in the Charpy impact dataset, varying the number of features from 16 to 9 and selecting the number of *k* neighbours from 1 to 10. The feature similarity equation (7.1) and a threshold value of 0.9 are selected. The mean results are shown in Table 8.3. It is observed that the number of features have an impact in the prediction accuracy of the testing dataset, overall the prediction is above 85%, the number of *k* neighbours seems to have a random effect on the prediction accuracy, therefore it can be concluded that any number of *k* neighbours may be selected without affecting significantly the overall results, as a rule of thumb a *k* value below 5 seems to be sufficient.

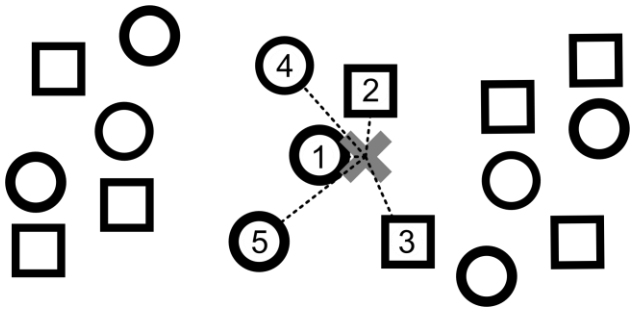


Figure 8.3: Example of a KNN classification utilizing Euclidean distances. If *k=1,5* then test sample will be classified as a circle, if *k=3* test sample is classified as square, tie resolution is problem dependent.

Table 8.3: Accuracy varying the number of features and the number of *k* neighbours

	16 Features	14 Features	12 Features	10 Features	8 Features
k=1	89.47	89.08	86.18	87.06	86.72
k=2	89.47	89.08	86.22	87.06	86.72
k=3	89.45	89.10	86.28	86.97	86.95
k=4	89.62	89.18	86.07	86.85	86.81
k=5	90.15	88.87	86.24	86.79	86.56
k=6	89.94	88.80	86.43	86.79	86.47
k=7	90.10	88.87	86.39	86.55	86.34
k=8	89.98	88.78	86.34	86.43	86.18
k=9	89.92	88.93	86.36	86.15	85.82
k=10	89.83	88.86	86.51	86.34	85.73

8.3 Effect of Inconsistencies in Performance

The effect of inconsistencies in performance is considerable. In Figure 8.4 a) and Figure 8.4 b) it is observed the regression plots of the results of the Charpy impact test and the UTS datasets respectively, utilizing all the features in the dataset and selecting a threshold value equal to the feature dependency measure. the consistent and inconsistent instances are shown in blue and red respectively. An increase in the RMSE of 54% and 99% for the Charpy and the UTS datasets respectively is measured. Such results show that a significant portion of the error in prediction can be attributed to inconsistencies present in the dataset.

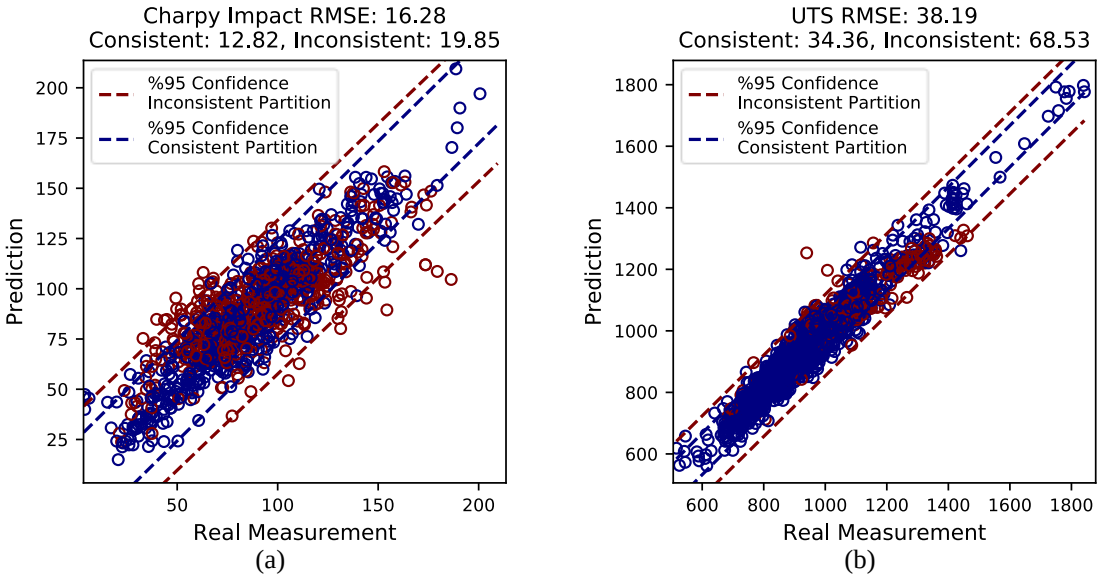


Figure 8.4: Effect of inconsistence in Charpy impact prediction (a) and UTS prediction (b).

8.4 Multiple Point Prediction for Datasets Containing Inconsistencies

On the one hand removing inconsistent objects from the dataset may cause the loss of valuable information, limiting the prediction capabilities of a model. On the other hand, inconsistencies may result in unreliable models and a considerable increase in the prediction error, as is observed in Figure 8.4. Therefore, in the presence of inconsistencies it is proposed to implement a modelling strategy, which considers the inconsistencies present in the dataset and perform predictions accordingly. Instead of providing a single point prediction, a set of predictions are to be presented in regions estimated to contain inconsistencies.

Two or more instances are considered inconsistent when contain the same or very similar feature values and different outputs. These inconsistencies result in a large portion in the error prediction, nonetheless, contain valuable information, given that the

inconsistencies do not arise due to errors in measurements, but for the lack of information. This was confirmed with the observed increase in inconsistencies with the removal of features.

In the case of the Charpy impact test, it is well known the considerable amount of inconsistencies in measurements. Some of these inconsistencies may be attributed to inhomogeneities in the microstructure [139], or other features difficult or non-cost efficient to measure.

A modelling paradigm is proposed to perform prediction in inconsistent datasets utilizing a multiple point prediction. The multiple point prediction is formed by a set of *M*-models, each trained with a different dataset containing a consistent partition of the training dataset and a number of inconsistent instances.

Initially, the inconsistencies are identifying utilizing the positive region of the fuzzy-rough sets, calculating utilizing the fuzzy similarity equation (7.1). The consistent instances are added to a set *C*, the inconsistent instances are divided into *N* different sets $I=\{I_1,...,I_N\}$ clusters utilizing a FCM algorithm. A SICFIS model is trained utilizing the set containing only the consistent instances, further *N* SICFIS models are trained utilizing the consistent partition and each one of the inconsistent partitions I_n. The process is summarized in Algorithm 8.1.

8.4.1 Results

In order to increase the number of inconsistencies a reduced dataset consisting on the 8 most important features obtained from the results obtained in Chapter 7 and shown in Table 7.2. The normalized-SICFIS model with 3 membership function per feature is

used for performing the predictions. A 1 K-NN algorithm is performed in the testing partition to identify inconsistencies.

Algorithm 8.1: Data selection for training M SICFIS models to perform the multiple point prediction.

Inputs: Charpy impact dataset ***H***, Threshold *Thr*
Output: Set containing consistent elements ***C***, set containing inconsistent elements ***I***, set of ***M*** trained SICFIS models

$C, I = \varnothing$

Calculate $\mu_{POS_{B_p}(Q)}$ for all the elements in *H*

For $j = 1:|H|$

If $\mu_{POS_{B_p}(Q)}(h_j) > Thr$: $C = C \cup \{h_j\}$

Else: $I = I \cup \{h_j\}$

Create a KNN model with *C* and *I*
Train $SICFIS_1$ with *C*
Create *N* clusters from inconsistent set *I;* $I = \{I_{c1}, I_{c2}, ..., I_{cn}\}$

For $j = 1:N$

Train $SICFIS_{j+1}$ with $C \cup I_{cj}$

End

The results for the consistent and inconsistent partitions are shown in Figure 8.5 an Figure 8.6 respectively. It can be observed a greater gap between the benchmark and the prediction intervals for the inconsistent testing partition. Table 8.4 shows the mean gap in prediction measured using the RMSE index. Furthermore, it is observed from Figure 8.6 both the benchmark model and the intervals seem to be unable to perform proper predictions to the inconsistent testing partition.

Table 8.4: Mean absolute prediction difference between the prediction interval for the consistent and inconsistent partitions

	Mean prediction interval absolute difference
Inconsistent Testing partition	29.31 RMSE
Consistent Testing Partition	13.95 RMSE

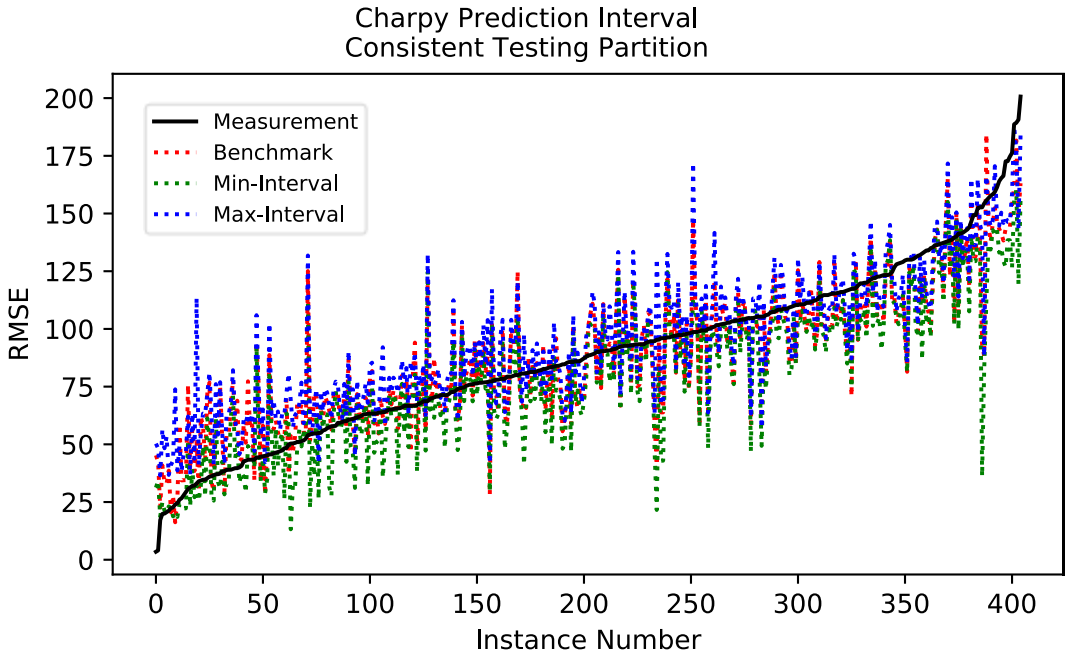


Figure 8.5: Charpy Impact test prediction interval for consistent testing partition.

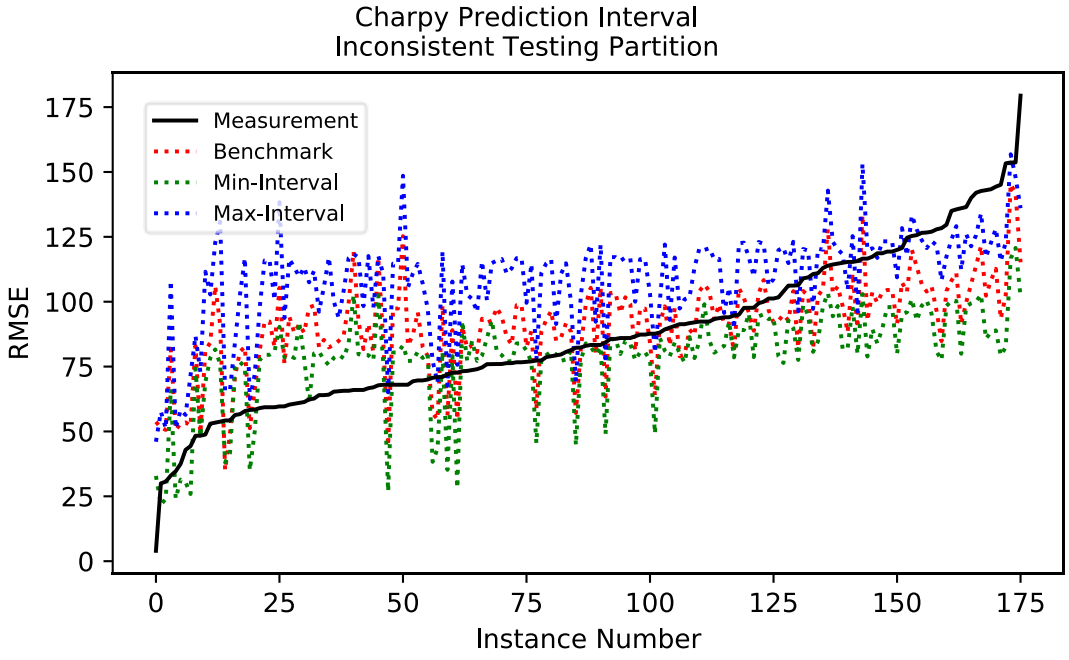


Figure 8.6: Charpy Impact test prediction interval for inconsistent testing partition.

8.5 Data-Mining Utilizing Fuzzy Rough Sets- Application to The Bladder Cancer Dataset

It was shown in Table 8.2 that the Cancer dataset contained the worse score in feature dependency, meaning that most of the records are inconsistent. This is well known in medicine, given that the different effects of lifestyle and genetics make it almost impossible to obtain consistent results. Utilizing a Threshold, it was selected the most consistent data points.

A summary of the results is shown in Table 8.5, the consistent partition consists of 97 patient records. Most of such records contain patients whose time of death was within the first five years. As observed by the mean observed time, being 10 months. The age, and grade means are also superior to the average, while the stage seems to be below the average.

Table 8.5: Cancer dataset comparison Consistent dataset

	Mean		Mode		Standard deviation	
Feature	All	Consistent*	All	Consistent*	All	Consistent*
Time	52.35	10.04	**	**	48.66	9.91
Age	71.59	76.24	**	**	11.02	9.42
Sex	0.73	0.68	1	1	0.44	0.47
Grade	2.17	2.48	3	3	0.80	0.75
Stage	4.03	3.02	6	2	2.25	2.02
Nodes	3.97	3.94	4	4	0.30	0.43
Squamous	0.04	0.10	0	0	0.21	0.30
CIS Present	0.13	0.20	0	0	0.33	0.40
SPB	2.03	2.00	2	2	0.60	0.79
Vascular	0.07	0.14	0	0	0.26	0.35
Urothelium	3.42	3.40	2	2	1.80	1.73
Muscle	0.72	0.73	1	1	0.45	0.44
Cystectomy	0.01	0.03	0	0	0.10	0.17
Radiotherapy	0.02	0.02	0	0	0.15	0.14

*Consistent: Consistent Partition.

8.6 Summary

In this work a method for evaluating the consistency of a dataset utilizing fuzzy rough set was implemented for data-mining. The feature dependency was shown to measure the average consistency of a dataset. Inconsistencies are the result of instances that contain the same or similar input values and exhibit different outputs.

It was demonstrated that a significant proportion of the errors in prediction can be attributed to the presence of inconsistencies in the dataset. Inconsistencies present in a testing partition can be identified by applying a k-NN algorithm. The gap in prediction accuracy between the benchmark model and the prediction interval increases considerably between the consistent and inconsistent testing partition data. It can be further concluded that fuzzy-rough set can be used to measure the limitations in prediction accuracy of a model given a dataset.

Additionally, fuzzy rough sets can be used to identify consistencies in the dataset as it was the case in the Cancer dataset, where it is possible to determine which parameter values produces more consistent results. Such information can be used by a medical professional for evaluating the life expectancy of a patient.

Chapter 9

Conclusions and Future Work

9.1 Conclusions

Among the research realized in the topic of CFS worldwide only three research groups have focused on the development of CFISs, resulting in the development of the ANCFIS, CNFIS and ACNFIS. Neither the ACNFIS nor the CNFIS model exploit the property of interference, which according to Ramot, is the main property of CFS. Furthermore, both models (CNFIS and ACNFIS) ignore, for the most part, the effect and meaning of the imaginary component of the output. It can be concluded that neither one of these two models are adequate CFISs and should be considered instead as modifications to the real-valued ANFIS. The ANCFIS model, however, utilizes the complex component of the CFS to model interferences by using a dot product operation. ANCFIS was developed for time series applications showing promising results. Regardless, none of the research groups have adequately addressed the problem of interpretability, *the raison d'etre* of fuzzy logic.

The SICFIS model introduced in Chapter 4 is therefore the first interpretable CFIS hitherto proposed. The SICFIS exploits the property of interference to model the complex interaction between features and outputs, resulting in a parsimonious model framework. The expansion to the complex domain presents several advantages over traditional FIS, including a higher prediction accuracy, faster computation times and greater interpretability given the number of tools capable of extracting and representing knowledge. The magnitude-phase plots demonstrate the full transparency; the interpretability analysis performed for the Charpy impact test demonstrated its interpretability. Both the normalized and fast SICFIS models outperformed most of the

FIS for different applications, and the choice of one over the other one is problem dependent, as was observed in the Bladder Cancer results, where the fast-SICFIS outperformed the normalized-SICFIS. This, in fact, can be attributed to the number of categorical variables present in the dataset.

Given the fast-SICFIS considerable reduction in computational time and the simple structure it was possible to improve upon the ANFIS model, by replacing the linear consequents with SICFISs models. The premises create a partition in the feature space, where each rule represents a local model. The global model is therefore composed of an ensemble of interpretable local SICFISs. The performance obtained is comparable with those obtained by a large ensemble of ANN. The interpretability of the model was assessed with a global-local performance index in all four datasets. Given the large number of categorical variables present in the Bladder Cancer dataset, there was a decrease in performance compared with the SICFIS.

The SICFIS model utilizes a complex singleton membership function. Type-1 singleton membership functions are known to be less interpretable and are less capable of modelling uncertainties compared with Gaussian membership functions. Therefore, in Chapter 6 the development of a complex Gaussian membership function is presented. A Mamdani-SICFIS is therefore created by replacing the complex singleton membership function with a complex Gaussian. The results obtained are comparable with other known neuro-FIS, but did not outperformed the singleton-SICFIS model, the reduction in accuracy can be compensated with the addition of a linguistic variables with context, making it potentially more interpretable than the SICFIS model.

The knowledge extracted from the SICFIS model may potentially be utilized for further applications. In Chapter 7 a feature selection algorithm is developed, based on the complex valued information obtained from the SICFIS output. The filter-SICFIS method assigns a score to each of the features based on their importance. The algorithm

performance is comparable with fuzzy rough sets and a wrapper-method, with a considerable reduction in computational time.

Fuzzy rough sets have been mostly utilized for feature selection. In Chapter 8 fuzzy rough sets are implemented into the Charpy and Bladder Cancer datasets. Both datasets present tough challenges given the number of inconsistencies, that where identified from the positive region of the lower approximation of the fuzzy rough sets. It was demonstrated that the prediction errors can be attributed greatly to the presence of inconsistencies.

9.2 Future Work

The considerable reduction in computation, shows promises for deploying the SICFIS model for real time applications. In areas such a control, the application of the fast-SICFIS model for nonlinear model predictive control may result in a reliable tool capable of producing accurate predictions in a timely manner. Furthermore, the complex component of the output may be utilized for real-time decision making in applications such as autonomous vehicles.

Overfitting was observed in the ANFIS-SICFIS model with the addition of rules. In order to improve upon the results and reduce overfitting, the implementation of regularization strategies may potentially solve this problem, while maintaining a good global-local performance. The implementation of better methods for rule elicitation may improve the results obtained even further. For datasets containing a large number of categorical variables further research needs to be conducted, such as the implementation of hyperparameter optimization.

While Gaussian membership functions are considered more interpretable, it is necessary to improve the Mamdani-SICFIS performance. By implementing interval

type-2 strategies the system may potentially model uncertainties and improve upon the results.

The application of complex-valued statistics for feature selection demonstrated the advantages of working in a higher dimensional plane. Further applications may be developed by researching further into the properties of CFS. Other areas of research may include the implementation of complex fuzzy rough sets, for developing better data-mining tools taking into consideration the “context”.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, “Delving Deep Into Rectifiers: Surpassing Human-Level Performance On Imagenet Classification,” In *2015 IEEE International Conference On Computer Vision (ICCV)*, 2015, pp. 1026–1034.

[2] P. Mamoshina, A. Vieira, E. Putin, and A. Zhavoronkov, “Applications Of Deep Learning In Biomedicine,” *Mol. Pharm.*, vol. 13, no. 5, pp. 1445–1454, May 2016.

[3] J. B. Heaton, N. G. Polson, and J. H. Witte, “Deep Learning In Finance,” *Arxiv160206561 Cs*, Feb. 2016.

[4] M. Kaminski, “The Right To Explanation, Explained,” *Berkeley Technol. Law J.*, vol. 34, no. 1, P. 189, May 2019.

[5] L. A. Zadeh, “Fuzzy Sets,” *Inf. Control*, vol. 8, no. 3, pp. 338–353, Jun. 1965.

[6] E. H. Mamdani, “Application Of Fuzzy Algorithms For Control Of Simple Dynamic Plant,” *Proc. Inst. Electr. Eng.*, vol. 121, no. 12, pp. 1585–1588, Dec. 1974.

[7] J. M. Alonso and L. Magdalena, “Special Issue On Interpretable Fuzzy Systems,” *Inf. Sci.*, vol. 181, no. 20, pp. 4331–4339, Oct. 2011.

[8] D. Ramot, R. Milo, M. Friedman, and A. Kandel, “Complex Fuzzy Sets,” *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 2, pp. 171–186, 2002.

[9] D. Dubois and H. Prade, “Putting Rough Sets and Fuzzy Sets Together,” In *Intelligent Decision Support: Handbook Of Applications and Advances Of The Rough Sets Theory*, R. Słowiński, Ed. Dordrecht: Springer Netherlands, 1992, pp. 203–232.

[10] Jang , J.S.R., Sun, C.T, and Mizutani, E., *Neuro-Fuzzy and Soft Computing; A Computational Approach To Learning and Machine Intelligence*. Prentice Hall, 1997.

[11] T. J. Ross, *Fuzzy Logic With Engineering Applications*, 3rd Ed. Chichester, U.K: John Wiley, 2010.

[12] O. Nelles, *Nonlinear System Identification: From Classical Approaches To Neural Networks and Fuzzy Models*. Berlin; London: Springer, 2011.

[13] J. Espinosa, J. Vandewalle, and V. Wertz, *Fuzzy Logic, Identification, and Predictive Control*. London ; New York: Springer, 2004.

[14] T. Takagi and M. Sugeno, “Fuzzy Identification Of Systems and Its Applications To Modeling and Control,” no. 1, P. 17, 1985.

[15] N. Yubazaki, J. Yi, M. Otani, and K. Hirota, “Sirm’s Connected Fuzzy Inference Model and Its Applications To First-Order Lag Systems and Second-Order Lag Systems,” In *Soft Computing In Intelligent Systems and Information Processing. Proceedings Of The 1996 Asian Fuzzy Systems Symposium*, 1996, pp. 545–550.

[16] H. Seki and M. Mizumoto, "On The Equivalence Conditions Of Fuzzy Inference Methods—Part 1: Basic Concept and Definition," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 6, pp. 1097–1106, Dec. 2011.

[17] W. E. Combs and J. E. Andrews, "Combinatorial Rule Explosion Eliminated By A Fuzzy Rule Configuration," *IEEE Trans. Fuzzy Syst.*, vol. 6, no. 1, pp. 1–11, Feb. 1998.

[18] M. Delgado, A. F. Gómez-Skarmeta, and F. Martín, "A Methodology To Model Fuzzy Systems Using Fuzzy Clustering In A Rapid-Prototyping Approach," *Fuzzy Sets Syst.*, vol. 97, no. 3, pp. 287–301, Aug. 1998.

[19] H. Sun, S. Wang, and Q. Jiang, "FCM-Based Model Selection Algorithms For Determining The Number Of Clusters," *Pattern Recognit.*, vol. 37, no. 10, pp. 2027–2037, Oct. 2004

[20] J. C. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*. Springer Us, 1981.

[21] H. E, Y. Cui, W. Pedrycz, and Z. Li, "Enhancements Of Rule-Based Models Through Refinements Of Fuzzy C-Means," *Knowl.-Based Syst.*, vol. 170, pp. 43–60, Apr. 2019.

[22] W. Pedrycz and H. Izakian, "Cluster-Centric Fuzzy Modeling," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 6, pp. 1585–1597, Dec. 2014.

[23] S. L. Chiu, "Fuzzy Model Identification Based On Cluster Estimation," *J Intell Fuzzy Syst*, vol. 2, no. 3, pp. 267–278, May 1994.

[24] S. Chiu, "Method and Software For Extracting Fuzzy Classification Rules By Subtractive Clustering," In *Proceedings Of North American Fuzzy Information Processing*, 1996, pp. 461–465.

[25] G. Panoutsos and M. Mahfouf, "A Neural-Fuzzy Modelling Framework Based On Granular Computing: Concepts and Applications," *Fuzzy Sets Syst.*, vol. 161, no. 21, pp. 2808–2830, Nov. 2010.

[26] G. Panoutsos and M. Mahfouf, "Granular Computing and Evolutionary Fuzzy Modelling For Mechanical Properties Of Alloy Steels," *IFAC Proc. vol.*, vol. 38, no. 1, pp. 205–210, 2005.

[27] G. Tsekouras, H. Sarimveis, E. Kavakli, and G. Bafas, "A Hierarchical Fuzzy-Clustering Approach To Fuzzy Modeling," *Fuzzy Sets Syst.*, vol. 150, no. 2, pp. 245–266, Mar. 2005.

[28] F. Herrera, M. Lozano, and J. L. Verdegay, "Tuning Fuzzy Logic Controllers By Genetic Algorithms," *Int. J. Approx. Reason.*, vol. 12, no. 3, pp. 299–315, Apr. 1995.

[29] S.- Horikawa, T. Furuhashi, and Y. Uchikawa, "On Fuzzy Modeling Using Fuzzy Neural Networks With The Back-Propagation Algorithm," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 801–806, Sep. 1992.

[30] K. Hornik, M. Stinchcombe, and H. White, "Multilayer Feedforward Networks Are Universal Approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, Jan. 1989.

[31] B. Kosko, "Fuzzy Systems As Universal Approximators," *IEEE Trans. Comput.*, vol. 43, no. 11, pp. 1329–1333, Nov. 1994.

[32] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.

[33] B. Widrow and M. A. Lehr, "30 Years Of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation," *Proc. IEEE*, vol. 78, no. 9, pp. 1415–1442, Sep. 1990.

[34] J.-R. Jang and C.- Sun, "Functional Equivalence Between Radial Basis Function Networks and Fuzzy Inference Systems," *IEEE Trans. Neural Netw.*, vol. 4, no. 1, pp. 156–159, Jan. 1993.

[35] Min-You Chen and D. A. Linkens, "A Systematic Neuro-Fuzzy Modeling Framework With Application To Material Property Prediction," *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 31, no. 5, pp. 781–790, Oct. 2001.

[36] L. A. Zadeh, "The Concept Of A Linguistic Variable and Its Application To Approximate Reasoning—I," *Inf. Sci.*, vol. 8, no. 3, pp. 199–249, Jan. 1975.

[37] Qilian Liang and J. M. Mendel, "Interval Type-2 Fuzzy Logic Systems: Theory and Design," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 5, pp. 535–550, Oct. 2000.

[38] L. A. T. Salomao, M. Mahfouf, E. El-Samahy, and C. Ting, "Psychophysiologically Based Real-Time Adaptive General Type 2 Fuzzy Modeling and Self-Organizing Control Of Operator's Performance Undertaking A Cognitive Task," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 1, pp. 43–57, Feb. 2017.

[39] O. Obajemu, M. Mahfouf, and J. W. F. Catto, "A New Fuzzy Modeling Framework For Integrated Risk Prognosis and Therapy Of Bladder Cancer Patients," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 3, pp. 1565–1577, Jun. 2018.

[40] O. Obajemu, M. Mahfouf, and L. A. Torres-Salomao, "A New Interval Type-2 Fuzzy Clustering Algorithm For Interval Type-2 Fuzzy Modelling With Application To Heat Treatment Of Steel," *IFAC Proc. vol.*, vol. 47, no. 3, pp. 10658–10663, 2014.

[41] Z. Pawlak, "Rough Sets," *Int. J. Comput. Inf. Sci.*, vol. 11, no. 5, pp. 341–356, Oct. 1982.

[42] A. G. Jackson, Z. Pawlak, and S. R. Leclair, "Rough Sets Applied To The Discovery Of Materials Knowledge," *J. Alloys Compd.*, vol. 279, no. 1, pp. 14–21, Sep. 1998.

[43] P. Lingras and G. Peters, "Rough Clustering," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 1, no. 1, pp. 64–72, 2011.

[44] N. Zhong, J. Dong, and S. Ohsuga, "Using Rough Sets With Heuristics For Feature Selection," *J. Intell. Inf. Syst.*, vol. 16, pp. 199–214, 2001.

[45] R. W. Swiniarski and A. Skowron, "Rough Set Methods In Feature Selection and Recognition," *Pattern Recognit. Lett.*, vol. 24, no. 6, pp. 833–849, Mar. 2003.

[46] X. Wang, J. Yang, X. Teng, W. Xia, and R. Jensen, "Feature Selection Based On Rough Sets and Particle Swarm Optimization," *Pattern Recognit. Lett.*, vol. 28, no. 4, pp. 459–471, Mar. 2007.

[47] U. Qamar, "A Rough-Set Feature Selection Model For Classification and Knowledge Discovery," In *2013 IEEE International Conference On Systems, Man, and Cybernetics*, 2013, pp. 788–793.

[48] A. M. Radzikowska and E. E. Kerre, “A Comparative Study Of Fuzzy Rough Sets,” *Fuzzy Sets Syst.*, vol. 126, no. 2, pp. 137–155, Mar. 2002.
[49] R. Jensen and Q. Shen, “New Approaches To Fuzzy-Rough Feature Selection,” *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 4, pp. 824–838, Aug. 2009.
[50] R. Jensen and Q. Shen, “Fuzzy-Rough Sets Assisted Attribute Selection,” *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 1, pp. 73–89, Feb. 2007.
[51] X. Wang, J. Yang, X. Teng, W. Xia, and R. Jensen, “Feature Selection Based On Rough Sets and Particle Swarm Optimization,” *Pattern Recognit. Lett.*, vol. 4, no. 28, pp. 459–471, 2007.
[52] R. B. Bhatt and M. Gopal, “On Fuzzy-Rough Sets Approach To Feature Selection,” *Pattern Recognit. Lett.*, vol. 26, no. 7, pp. 965–975, May 2005.
[53] J. Stefanowski, “On Rough Set Based Approaches To Induction Of Decision Rules,” 1998.
[54] K. Y. Huang, “Applications Of An Enhanced Cluster Validity Index Method Based On The Fuzzy C-Means and Rough Set Theories To Partition and Classification,” *Expert Syst. Appl.*, vol. 37, no. 12, pp. 8757–8769, Dec. 2010.
[55] D. Ramot, M. Friedman, G. Langholz, and A. Kandel, “Complex Fuzzy Logic,” *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 4, pp. 450–461, 2003.
[56] S. Dick, “Toward Complex Fuzzy Logic,” *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 3, pp. 405–414, 2005.
[57] O. Yazdanbakhsh and S. Dick, “A Systematic Review Of Complex Fuzzy Sets and Logic,” *Fuzzy Sets Syst.*, vol. 1, pp. 1–22, 2016.
[58] D. E. Tamir, L. Jin, and A. Kandel, “A New Interpretation Of Complex Membership Grade,” *Int. J. Intell. Syst.*, vol. 26, no. 4, pp. 285–312, Apr. 2011.
[59] L. Běhounek and P. Cintula, “Fuzzy Class Theory,” *Fuzzy Sets Syst.*, vol. 154, no. 1, pp. 34–55, Aug. 2005.
[60] K. T. Atanassov, “Intuitionistic Fuzzy Sets,” *Fuzzy Sets Syst.*, vol. 20, no. 1, pp. 87–96, Aug. 1986.
[61] R. R. Yager, “Pythagorean Membership Grades In Multicriteria Decision Making,” *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 4, pp. 958–965, Aug. 2014.
[62] A. (Moh’d J. S. Alkouri and A. R. Salleh, “Complex Intuitionistic Fuzzy Sets,” Presented At The International Conference On Fundamental and Applied Sciences 2012: (Icfas2012), Kuala Lumpur Convention Centre, Kuala Lumpur, Malaysia, 2012, pp. 464–470.
[63] M. Ali and F. Smarandache, “Complex Neutrosophic Set,” *Neural Comput. Appl.*, vol. 28, no. 7, pp. 1817–1834, Jul. 2017.
[64] S. Greenfield and F. Chiclana, “Fuzzy In 3-D: Contrasting Complex Fuzzy Sets With Type-2 Fuzzy Sets,” *Proc. 2013 Jt. IFSA World Congr. Nafips Annu. Meet. IFSANAFIPS 2013*, no. Cci, pp. 1237–1242, 2013.
[65] S. Greenfield, F. Chiclana, and S. Dick, “Join and Meet Operations For Interval-Valued Complex Fuzzy Logic,” In *2016 Annual Conference Of The North American Fuzzy Information Processing Society (NAFIPS)*, 2016, pp. 1–5.

[66] S. Greenfield, F. Chiclana, and S. Dick, “Interval-Valued Complex Fuzzy Logic,” In *2016 IEEE International Conference On Fuzzy Systems (Fuzz-IEEE)*, 2016, pp. 2014–2019.

[67] S. Greenfield and F. Chiclana, “Fuzzy In 3–D: Two Contrasting Paradigms,” *Arch. Philos. Hist. Soft Comput.*, vol. 0, no. 2, Jan. 2016.

[68] Y. Li and Y.-T. Jang, “Complex Adaptive Fuzzy Inference Systems,” In *Soft Computing In Intelligent Systems and Information Processing. Proceedings Of The 1996 Asian Fuzzy Systems Symposium*, 1996, pp. 551–556.

[69] A. Malekzadeh-A and M. Akbarzadeh-T, “Complex-Valued Adaptive Neuro Fuzzy Inference System-CANFIS,” In *Proceedings World Automation Congress, 2004.*, 2004, vol. 17, pp. 477–482.

[70] A. Y. Deshmukh, A. B. Bavaskar, D. P. R. Bajaj, and D. A. G. Keskar, “Implementation Of Complex Fuzzy Logic Modules With VLSI Approach,” P. 7, 2008.

[71] K. Subramanian, R. Savitha, and S. Suresh, “A Complex-Valued Neuro-Fuzzy Inference System and Its Learning Mechanism,” *Neurocomputing*, vol. 123, pp. 110–120, 2014.

[72] R. Hata, M. Islam, and K. Murase, “Generation Of Fuzzy Rules Based On Complex-Valued Neuro-Fuzzy Learning Algorithm,” *J Jpn Soc Fuzzy Theory Intell Inf.*, vol. 27, no. 1, pp. 533–548, 2015.

[73] J. Y. Man, Z. Chen, and S. Dick, “Towards Inductive Learning Of Complex Fuzzy Inference Systems,” 2007, pp. 415–420.

[74] C. Li and T.-W. Chiang, “Complex Neuro-Fuzzy Self-Learning Approach To Function Approximation,” In *Intelligent Information and Database Systems*, vol. 5991, N. T. Nguyen, M. T. Le, and J. Świątek, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 289–299.

[75] R. Shoorangiz and M. H. Marhaban, “Complex Neuro-Fuzzy System For Function Approximation,” *Int. J. Appl. Electron. Phys. Robot.*, vol. 1, no. 2, pp. 5–9, 2013.

[76] J.-R. Jang, “ANFIS: Adaptive-Network-Based Fuzzy Inference System,” *IEEE Trans. Syst. Man Cybern.*, vol. 23, no. 3, pp. 665–685, May 1993.

[77] H. Leung and S. Haykin, “The Complex Backpropagation Algorithm,” *IEEE Trans. Signal Process.*, vol. 39, no. 9, pp. 2101–2104, Sep. 1991.

[78] Z. Chen, S. Aghakhani, J. Man, and S. Dick, “ANCFIS: A Neurofuzzy Architecture Employing Complex Fuzzy Sets,” *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 2, pp. 305–322, 2011.

[79] S. Aghakhani and S. Dick, “An On-Line Learning Algorithm For Complex Fuzzy Logic,” 2010, pp. 1–7.

[80] O. Yazdanbaksh, A. Krahn, and S. Dick, “Predicting Solar Power Output Using Complex Fuzzy Logic,” In *2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA / NAFIPS)*, 2013, pp. 1243–1248.

[81] O. Yazdanbakhsh and S. Dick, “Multi-Variate Timeseries Forecasting Using Complex Fuzzy Logic,” In *2015 Annual Conference Of The North American Fuzzy*

Information Processing Society (NAFIPS) Held Jointly With 2015 5th World Conference On Soft Computing (Wconsc), 2015, pp. 1–6.

[82] O. Yazdanbakhsh and S. Dick, "Forecasting Of Multivariate Time Series Via Complex Fuzzy Logic," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 47, no. 8, pp. 2160–2171, Aug. 2017.

[83] O. Yazdanbakhsh and S. Dick, "ANCFIS-ELM: A Machine Learning Algorithm Based On Complex Fuzzy Sets," 2016, pp. 2007–2013.

[84] C. Li and T.-W. Chiang, "Intelligent Financial Time Series Forecasting: A Complex Neuro-Fuzzy Approach With Multi-Swarm Intelligence," *Int. J. Appl. Math. Comput. Sci.*, vol. 22, no. 4, pp. 787–800, Dec. 2012.

[85] C. Li and T.-W. Chiang, "Complex Neurofuzzy Arima Forecasting—A New Approach Using Complex Fuzzy Sets," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 3, pp. 567–584, Jun. 2013.

[86] C. Li and F. Chan, "Complex-Fuzzy Adaptive Image Restoration – An Artificial-Bee-Colony-Based Learning Approach," In *Intelligent Information and Database Systems*, 2011, pp. 90–99.

[87] C. Li and T.-W. Chiang, "Complex Fuzzy Computing To Time Series Prediction — A Multi-Swarm PSO Learning Approach," In *Intelligent Information and Database Systems*, 2011, pp. 242–251.

[88] C. Li and F.-T. Chan, "Knowledge Discovery By An Intelligent Approach Using Complex Fuzzy Sets," In *Intelligent Information and Database Systems*, vol. 7196, J.-S. Pan, S.-M. Chen, and N. T. Nguyen, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 320–329.

[89] C. Li, T.-W. Chiang, and L.-C. Yeh, "A Novel Self-Organizing Complex Neuro-Fuzzy Approach To The Problem Of Time Series Forecasting," *Neurocomputing*, vol. 99, pp. 467–476, 2013.

[90] C. Mencar and A. M. Fanelli, "Interpretability Constraints For Fuzzy Information Granulation," *Inf. Sci.*, vol. 178, no. 24, pp. 4585–4618, Dec. 2008.

[91] Z. C. Lipton, "The Mythos Of Model Interpretability," *Arxiv160603490 Cs Stat*, Jun. 2016.

[92] A. Riid, R. Isotamm, and E. Rüstern, "Transparency Analysis Of First-Order Takagi-Sugeno Systems," P. 7.

[93] M. J. Gacto, R. Alcalá, and F. Herrera, "Interpretability Of Linguistic Fuzzy Rule-Based Systems: An Overview Of Interpretability Measures," *Inf. Sci.*, vol. 181, no. 20, pp. 4340–4360, Oct. 2011.

[94] T. R. Razak, J. M. Garibaldi, C. Wagner, A. Pourabdollah, and D. Soria, "Interpretability Indices For Hierarchical Fuzzy Systems," In *2017 IEEE International Conference On Fuzzy Systems (Fuzz-IEEE)*, 2017, pp. 1–6.

[95] J. M. Alonso, C. Castiello, and C. Mencar, "Interpretability Of Fuzzy Systems: Current Research Trends and Prospects," In *Springer Handbook Of Computational Intelligence*, Springer, Berlin, Heidelberg, 2015, pp. 219–237.

[96] J. M. Alonso, L. Magdalena, and G. González-Rodríguez, "Looking For A Good Fuzzy System Interpretability Index: An Experimental Approach," *Int. J. Approx. Reason.*, vol. 51, no. 1, pp. 115–134, Dec. 2009.

[97] G. A. Miller, "The Magical Number Seven, Plus Or Minus Two: Some Limits On Our Capacity For Processing Information," *Psychol. Rev.*, vol. 63, no. 2, pp. 81–97, 1956.

[98] J. Yen, L. Wang, and C. W. Gillespie, "Improving The Interpretability Of Tsk Fuzzy Models By Combining Global Learning and Local Learning," *IEEE Trans. Fuzzy Syst.*, vol. 6, no. 4, pp. 530–537, Nov. 1998.

[99] M. Á. Vélez, O. Sánchez, S. Romero, and J. M. Andújar, "A New Methodology To Improve Interpretability In Neuro-Fuzzy Tsk Models," *Appl. Soft Comput.*, vol. 10, no. 2, pp. 578–591, Mar. 2010.

[100] D. A. Brandt and J. C. Warner, *Metallurgy Fundamentals*, 5th Ed. Tinley Park, Il: Goodheart-Willcox, 2009.

[101] M. A. Meyers and K. K. Chawla, *Mechanical Behavior Of Materials*.

[102] J. Rösler, H. Harders, and M. Bäker, *Mechanical Behaviour Of Engineering Materials: Metals, Ceramics, Polymers, and Composites*. Berlin ; New York: Springer, 2007.

[103] A. Pineau, A. A. Benzerga, and T. Pardoen, "Failure Of Metals I: Brittle and Ductile Fracture," *Acta Mater.*, vol. 107, pp. 424–483, Apr. 2016.

[104] M. Mahfouf, Y. Y. Yang, and Q. Zhang, "Characterisation Of Model Error For Charpy Impact Energy Of Heat Treated Steels Using Probabilistic Reasoning and A Gaussian Mixture Model," *IFAC Proc. vol.*, vol. 42, no. 23, pp. 225–230, 2009.

[105] R. Colas-Marquez and M. Mahfouf, "Data Mining and Modelling Of Charpy Impact Energy For Alloy Steels Using Fuzzy Rough Sets," *IFAC-Pap.*, vol. 50, no. 1, pp. 14970–14975, Jul. 2017.

[106] *Materials Characterization Using Nondestructive Evaluation (NDE) Methods*. Elsevier, 2016.

[107] D. Collett, *Modelling Survival Data In Medical Research*. 2015.

[108] R. Dybowski and V. Gant, *Clinical Applications Of Artificial Neural Networks*. Cambridge; New York: Cambridge University Press, 2007.

[109] K. Hamidieh, "A Data-Driven Statistical Model For Predicting The Critical Temperature Of A Superconductor," *Comput. Mater. Sci.*, vol. 154, pp. 346–354, Nov. 2018.

[110] "UCI Machine Learning Repository: Superconductivity Data Set." [Online]. Available: Https://Archive.Ics.Uci.Edu/Ml/Datasets/Superconductivty+Data. [Accessed: 27-May-2019].

[111] D. M. Hawkins, "The Problem Of Overfitting," *J. Chem. Inf. Comput. Sci.*, vol. 44, no. 1, pp. 1–12, Jan. 2004.

[112] H. Seki and T. Nakashima, "Complex-Valued SIRMS Connected Fuzzy Inference Model," In *2014 IEEE International Conference On Granular Computing (GRC)*, Noboribetsu, Japan, 2014, pp. 250–253.

[113] J. H. Lilly, *Fuzzy Control and Identification*. Hoboken, Nj, USA: John Wiley & Sons, Inc., 2010.

[114] Q. Zhang and M. Mahfouf, "Fuzzy Modelling Using A New Compact Fuzzy System: A Special Application To The Prediction Of The Mechanical Properties Of Alloy Steels," 2011, pp. 1041–1048.

[115] A. Botev, H. Ritter, and D. Barber, "Practical Gauss-Newton Optimisation For Deep Learning," *Arxiv170603662 Stat*, Jun. 2017.

[116] M. T. Hagan and M. B. Menhaj, "Training Feedforward Networks With The Marquardt Algorithm," *IEEE Trans. Neural Netw.*, vol. 5, no. 6, pp. 989–993, Nov. 1994.

[117] R. Muscat and M. Mahfouf, "Predicting Charpy Impact Energy For Heat-Treated Steel Using A Quantum-Membership-Function-Based Fuzzy Model," *IFAC-Pap.*, vol. 49, no. 20, pp. 138–142, 2016.

[118] Shen Wang and M. Mahfouf, "Multi-Objective Optimisation For Fuzzy Modelling Using Interval Type-2 Fuzzy Sets," 2012, pp. 1–8.

[119] J. R. Davis, Ed., *Alloying: Understanding The Basics*. Materials Park, Ohio: Asm International, 2011.

[120] G. R. Speich, D. S. Dabkowski, and L. F. Porter, "Strength and Toughness Of Fe-10ni Alloys Containing C, Cr, Mo, and Co," *Metall. Trans.*, vol. 4, no. 1, pp. 303–315, Jan. 1973.

[121] H. Takagi, N. Suzuki, T. Koda, and Y. Kojima, "Neural Networks Designed On Approximate Reasoning Architecture and Their Applications," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 752–760, Sep. 1992.

[122] E. Mizutani and J.-R. Jang, "Coactive Neural Fuzzy Modeling," In *Proceedings Of Icnn'95 - International Conference On Neural Networks*, 1995, vol. 2, pp. 760–765 vol.2.

[123] R. Rajesh and M. R. Kaimal, "T–S Fuzzy Model With Nonlinear Consequence and Pdc Controller For A Class Of Nonlinear Control Systems," *Appl. Soft Comput.*, vol. 7, no. 3, pp. 772–782, Jun. 2007.

[124] A. Sala and C. Ariño, "Polynomial Fuzzy Models For Nonlinear Control: A Taylor Series Approach," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 6, pp. 1284–1295, Dec. 2009.

[125] K. Tanaka, H. Yoshida, H. Ohtake, and H. O. Wang, "A Sum-Of-Squares Approach To Modeling and Control Of Nonlinear Dynamical Systems With Polynomial Fuzzy Systems," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 4, pp. 911–922, Aug. 2009.

[126] J. Dong, Y. Wang, and G. Yang, "Output Feedback Fuzzy Controller Design With Local Nonlinear Feedback Laws For Discrete-Time Nonlinear Systems," *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 40, no. 6, pp. 1447–1459, Dec. 2010.

[127] M. Delgado, A. F. Gomez-Skarmeta, and F. Martin, "A Fuzzy Clustering-Based Rapid Prototyping For Fuzzy Rule-Based Modeling," *IEEE Trans. Fuzzy Syst.*, vol. 5, no. 2, pp. 223–233, May 1997.

[128] J. V. De Oliveira and W. Pedrycz, “Advances In Fuzzy Clustering and Its Applications,” P. 457.

[129] Y. Y. Yang, M. Mahfouf, and G. Panoutsos, “Development Of A Parsimonious GA–NN Ensemble Model With A Case Study For Charpy Impact Energy Prediction,” *Adv. Eng. Softw.*, vol. 42, no. 7, pp. 435–443, Jul. 2011.

[130] M. I. Rey, M. Galende, M. J. Fuente, and G. I. Sainz-Palmero, “Multi-Objective Based Fuzzy Rule Based Systems (FRBSS) For Trade-Off Improvement In Accuracy and Interpretability: A Rule Relevance Point Of View.,” *Knowl.-Based Syst.*, vol. 127, pp. 67–84, Jul. 2017.

[131] F. Rudziński, “A Multi-Objective Genetic Optimization Of Interpretability-Oriented Fuzzy Rule-Based Classifiers,” *Appl. Soft Comput.*, vol. 38, pp. 118–133, Jan. 2016.

[132] S. Greenfield and F. Chiclana, “Fuzzy In 3-D: Contrasting Complex Fuzzy Sets With Type-2 Fuzzy Sets,” 2013, pp. 1237–1242.

[133] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, “Recent Advances and Emerging Challenges Of Feature Selection In The Context Of Big Data,” *Knowl.-Based Syst.*, vol. 86, pp. 33–45, Sep. 2015.

[134] S. Shilaskar and A. Ghatol, “Feature Selection For Medical Diagnosis : Evaluation For Cardiovascular Diseases,” *Expert Syst. Appl.*, vol. 40, no. 10, pp. 4146–4153, Aug. 2013.

[135] M. F. Akay, “Support Vector Machines Combined With Feature Selection For Breast Cancer Diagnosis,” *Expert Syst. Appl.*, vol. 36, no. 2, Part 2, pp. 3240–3247, Mar. 2009.

[136] I. Guyon and A. Elisseeff, “An Introduction To Variable and Feature Selection,” p. 26, 2003.

[137] K. I. Park, *Fundamentals Of Probability and Stochastic Processes With Applications To Communications*. Cham: Springer International Publishing, 2018.

[138] R. O. ; S. Duda David G. ;. Hart, Peter E., *Pattern Classification*. Hoboken, United States: Wiley, 2001.

[139] R. Bouchard, G. Shen, and W. R. Tyson, “Fracture Toughness Variability Of Structural Steel,” *Eng. Fract. Mech.*, vol. 75, no. 12, pp. 3735–3742, Aug. 2008.