Thèse / ISGT / Août 2014

New Danger Classification Methods in an Imprecise Framework

Zeineb CHELLY

Université de Tunis
Institut Supérieur de Gestion
Ecole Doctorale Sciences de Gestion

New Danger Classification Methods in an Imprecise Framework

THESE
En vue de l'obtention du Doctorat
en Informatique de Gestion

Présentée et soutenue publiquement par:
Zeineb CHELLY

Le 28 Août 2014

Membres du Jury:

♦ *Mme. Nahla BEN AMOR*	*Professeur*	*à l'Université de Tunis*	*Président*
♦ *M. Zied ELOUEDI*	*Professeur*	*à l'Université de Tunis*	*Directeur de thèse*
♦ *M. Salah BENABDALLAH*	*Maître de conférences*	*à l'Université de Tunis*	*Rapporteur*
♦ *M. Talel LADHARI*	*Professeur*	*à l'Université de Tunis*	*Rapporteur*
♦ *Mme. Lamia LABED JILANI*	*Maître de conférences*	*à l'Université de Tunis*	*Membre*

Année Universitaire 2013 - 2014

Acknowledgments

Firstly, and most importantly, I would like to thank my Mum, ***Souad***, and Dad, ***Fradj***, for their support and encouragement that led me to where I am. I thank them very much for bearing me, raising me, supporting me, teaching me, and loving me. To them I dedicate this dissertation.

Huge thanks go to my supervisor ***Pr. Zied Elouedi*** for introducing me to an interesting area of research. I thank him for his continuous guidance and for all the valuable discussions we have had during the period of work. He has been very patient in supervising this work at all its stages. His comments and his observations have always been of great interest. I thank him for his support and for his constant encouragements.

People who deserve a special mention include ***Dr. Julie Greensmith***, who provided me with a deep knowledge in the danger theory area and most importantly the Dendritic Cell Algorithm; ***Dr. Robert Foster Oates*** and ***Dr. Feng Gu*** who always made themselves available for discussions and advices. Great thanks go to ***Pr. Richard Jensen*** for his valuable comments on the theoretical investigations of this Thesis.

I, also, want to thank the artificial immune system community who has heartily welcomed me and considered me one of them, specifically ***Pr. Uwe Aickelin***, ***Pr. Jon Timmis***, ***Pr. Emma Hart***, ***Dr. Christine Zarges*** and ***Dr. Thomas Jansen***.

I would, also, like to give my special thanks to my family, ***Imen, Souhaila, Insar*** and ***Zied*** as proof of love and gratitude for all their generous and voluntary help.

Heartfelt thanks to all of my friends for their encouragement; especially to ***Abir Smiti***, ***Imen Boukhris***, ***Souhir BenSoussi***, ***Amir Jlassi*** and ***Slim Bechikh***. Very special and heartfelt thanks to ***Stéphane Garcia***. I, also, thank very much my dearest friends ***Mounir Alibi***, ***Rym Ben Abed Allah***, ***Abir Bousmina***, ***Rabeb Berred***, ***Abdul Shukoor***, ***Radu Coliban***, ***Peter Scully***, ***Natalie Stickel***, ***Smita Rajpal***, ***Lavneet Singh***, ***Theo Heinze*** and ***Annie Joseph***.

Also, many thanks go to ***all LARODEC members***.

To all those who believed in me, who love me and who followed me and helped me in the way I chose.

Contents

Introduction **1**

Part I : Theoretical Aspects **5**

1 The Danger Theory **7**

1.1 Introduction 7

1.2 Artificial Immune Systems 7

1.2.1 Immunological Concepts 7

1.2.2 Synopsis of Artificial Immune Systems 8

1.3 Danger Theory Perception 10

1.3.1 Fundamental Concepts 10

1.3.2 Danger Description 10

1.3.3 Immunological Signals Taxonomy 11

1.4 The Dendritic Cell Algorithm 11

1.4.1 Biological Background 12

1.4.2 Algorithmic Details 13

1.4.3 DCA: An Example 15

1.4.4 Development Pathway 18

1.4.5 Strengths and Weaknesses of the DCA 19

1.5 Conclusion 20

2 Rough Sets and Fuzzy-Rough Sets for Feature Selection **21**

2.1 Introduction 21

2.2 Data Reduction Approaches 21

2.3 Rough Set Based Approach for Feature Selection 22

2.3.1 Decision and Information Systems 22

2.3.2 Indiscernibility Relation 23

2.3.3 Lower and Upper Approximations 24

2.3.4 Independency of Attributes 25

2.3.5 Core and Reduct of Attributes 25

2.3.6 Rough Sets and the Use of Heuristics 27

2.3.7 Rough Sets and the Link to Other Theories 27

2.4 Fuzzy-Rough Feature Selection 27

2.4.1 From Rough Sets to Fuzzy-Rough Sets 28

2.4.2 Fuzzy Equivalence Classes 28

2.4.3 Fuzzy-Rough Sets Basic Concepts 29

2.4.4 Fuzzy-Rough Reduction Process 29

2.4.5 Fuzzy-Rough QuickReduct 30

2.5 Conclusion 33

Part II : Contributions **35**

3 A Rough DCA Pre-Processing Module **37**

3.1 Introduction 37

3.2 Analysis of the Standard DCA Data Pre-processing Module 37

3.2.1 Criticisms of the Feature Extraction Step 37

3.2.2 Criticisms of the Signal Categorization Step 38

3.3 Rough DCA Based Methods 38

3.4 RST-DCA 39

3.4.1 RST-DCA Feature Selection Process 39

3.4.2 RST-DCA Signal Categorization Process 40

3.4.3 Experimental Setup 42
3.4.4 Results and Analysis 44
3.5 RC-DCA 45
3.5.1 RC-DCA Feature Selection Process 45
3.5.2 RC-DCA Signal Categorization Process 45
3.5.3 Experimental Setup, Results and Discussion 47
3.6 QR-DCA 49
3.6.1 Why Do We Need an Heuristic? 49
3.6.2 QR-DCA Feature Selection Process 49
3.6.3 QR-DCA Signal Categorization Process 50
3.6.4 Experimental Setup, Results and Analysis 52
3.7 Conclusion 56

4 A Fuzzy-Rough DCA Pre-Processing Module 57
4.1 Introduction 57
4.2 Why Adopting Fuzzy-Rough Sets for Data Pre-processing? 57
4.2.1 Criticisms of the Rough DCAs 57
4.2.2 The Fuzzy-Rough Proposed Solutions 58
4.3 FRST-DCA 58
4.3.1 The FRST-DCA Feature Selection Process 58
4.3.2 The FRST-DCA Signal Categorization Process 59
4.3.3 Experimental Setup, Results and Analysis 60
4.4 FLA-DCA 61
4.4.1 FRST Main Concepts Based on Similarity Relations 61
4.4.2 The FLA-DCA Feature Selection Process 62
4.4.3 The FLA-DCA Signal Categorization Process 63
4.4.4 Experimental Setup, Results and Discussion 65
4.5 FBR-DCA 66
4.5.1 FRST Concepts Based on the Fuzzy Boundary Region 66

4.5.2 The FBR-DCA Feature Selection Process . . . 67
4.5.3 The FBR-DCA Signal Categorization Process . . . 67
4.5.4 Experimental Setup, Results and Discussion . . . 69
4.6 Comparison of the Proposed Fuzzy-Rough DCAs . . . 70
4.6.1 Experimental Setup . . . 70
4.6.2 Discussion of the Obtained Results . . . 71
4.6.3 Comparison with the State-of-the-art Classifiers . . . 73
4.7 Conclusion . . . 73

5 An Automated Fuzzy Dendritic Cell Classification Method 75

5.1 Introduction . . . 75
5.2 Problem Statement . . . 75
5.3 Overview of the Fuzzy Clustering Techniques . . . 76
5.3.1 The Fuzzy c-means Algorithm . . . 76
5.3.2 Gustafson-Kessel Algorithm . . . 77
5.3.3 Shell Clustering Algorithm . . . 78
5.4 The Fuzzy Classification Dendritic Cell Method . . . 79
5.4.1 FCDCM Data Pre-Processing Phase . . . 80
5.4.2 FCDCM Detection Phase . . . 80
5.4.3 FCDCM Fuzzy Process . . . 81
5.4.4 FCDCM Classification Phase . . . 85
5.4.5 Worked Example . . . 85
5.5 Experimental Methodology, Results and Discussion . . . 90
5.5.1 Parameters Settings and Evaluating Criteria . . . 90
5.5.2 Results and Discussion . . . 91
5.5.3 Comparison with State-of-the-art Recent Methods . . . 96
5.6 Conclusion . . . 97

6 The Fuzzy Maintained Dendritic Cell Classification Method 99

6.1 Introduction . . . 99

6.2 Overview of the Maintenance Database Policies 99
6.2.1 The Selective Reduction Approaches 100
6.2.2 The Deletion Reduction Approaches 100
6.3 The Fuzzy Maintained COID-$FCDCM_{GK}$ Algorithm 101
6.3.1 COID-$FCDCM_{GK}$ Layout 101
6.3.2 COID-FCDCM Functioning 103
6.3.3 COID-$FCDCM_{GK}$: Worked Example 105
6.4 Experimental Methodology, Results and Discussion 107
6.4.1 Experimental Hypotheses and Evaluating Criteria 107
6.4.2 Results and Analysis of Maintenance Policies 108
6.4.3 Results and Discussion about H2: Comparing DCA and COID-DCA 109
6.4.4 Results and Discussion about COID-$FCDCM_{GK}$ 111
6.4.5 Comparison with State-of-the-art Recent Methods 113
6.5 Conclusion 114

7 The Hybrid Automated and Maintained Fuzzy Dendritic Cell Immune Classifier 115
7.1 Introduction 115
7.2 Analysis of the COID-$FCDCM_{GK}$ Algorithm 115
7.2.1 Description of the Current State 115
7.2.2 Check Points 116
7.3 The Rough $FCDCM_{GK}$ Classifiers 118
7.3.1 Description of the Hybrid Rough $FCDCM_{GK}$ Algorithms 118
7.3.2 Results and Discussion 119
7.4 The Fuzzy-Rough $FCDCM_{GK}$ Classifiers 123
7.4.1 Description of the Hybrid Fuzzy-Rough $FCDCM_{GK}$ Algorithms 123
7.4.2 Results and Discussion 124
7.5 The COID-FLA-$FCDCM_{GK}$ Immune Classifier 127
7.5.1 COID-FLA-$FCDCM_{GK}$ Description 127
7.5.2 COID-FLA-$FCDCM_{GK}$ Algorithmic Details 128

7.5.3 Experimental Setup, Results and Analysis 131
7.6 Summary 134
7.7 Conclusion 135

Conclusion **137**

Appendix **141**

A Fuzzy Set Theory **143**
A.1 Introduction 143
A.2 Fuzzy sets: Definition and Notations 143
A.3 Membership Functions 144
A.4 Operations on Fuzzy Sets 144
A.5 Fuzzy Logic 146
A.6 Worked Example 148
A.7 Conclusion 149

References **151**

List of Figures

1.1 Apoptosis and Necrosis Illustrations 11

1.2 DCs Behavior 12

2.1 Fuzzification for Conditional Features 31

3.1 The Rough DCA Proposed Model 39

3.2 Comparison of Classifiers Average Accuracies 55

4.1 Comparison of Classifiers Average Accuracies 73

5.1 Global View of FCDCM 79

5.2 Five Steps of the Fuzzy Process 80

5.3 An Illustrative Schema of the Centroid Value Generation 89

5.4 Comparison of Classifiers Average Accuracies 96

6.1 The COID-$FCDCM_{GK}$ Steps 102

6.2 The COID-$FCDCM_{GK}$ Detection Phase Steps 102

6.3 Instance Categories 103

6.4 Overview of the COID-$FCDCM_{GK}$ Maintenance Process 104

6.5 Reduced Signal Data Set 105

6.6 Comparison of Classifiers Average Accuracies 114

7.1 The COID-FLA-$FCDCM_{GK}$ Architecture 128

7.2 Comparative Results 133

7.3 A Progress Summary: from DCA to COID-FLA-$FCDCM_{GK}$ 134

A.1 Fuzzy Logic Controller 146

A.2 Defining the Membership Functions Value's of Each Input 148

A.3 Composition Process 149

A.4 Centroid Method for Defuzzification 149

List of Tables

1.1 Bank Database 16

1.2 Signal Data Set 16

1.3 Example of Weights Used for Signal Processing 16

1.4 Worked Example of MCAV Output 17

2.1 Information System 23

2.2 Decision System 23

2.3 First Reduct 26

2.4 Second Reduct 26

2.5 Example Database 30

2.6 Membership Values of Objects to Corresponding Fuzzy Sets 31

3.1 The RST-DCA Decision System 42

3.2 Description of Databases 43

3.3 PCA-DCA and RST-DCA Comparison Results 44

3.4 The RC-DCA Decision System 46

3.5 PCA'-DCA and RC-DCA Comparison Results 48

3.6 The QR-DCA Decision System 50

3.7 Comparison Results of the Rough DCAs 54

4.1 Comparison Results of QR-DCA and FRST-DCA 61

4.2 Comparison Results of FRST-DCA and FLA-DCA 65
4.3 Comparison Results of FLA-DCA and FBR-DCA 70
4.4 Comparison Results of FRST-DCA, FLA-DCA and FBR-DCA 72

5.1 The Fuzzy Rule Base 84
5.2 Initial Data Set 85
5.3 Reduced Data Set 86
5.4 Process of Calculating the Absolute Distance 87
5.5 Signal Feature Vectors 87
5.6 Numerical Values of Validity Measures (%) 92
5.7 Experimental Measures 93
5.8 Experimental Measure: Accuracy (%) 95

6.1 Signal Data Set 105
6.2 Comparing Database Maintenance Schemes 109
6.3 Experimental Measures: Accuracy (%) 110
6.4 Experimental Measures 112
6.5 Comparison in Terms of Execution Time (s) 113

7.1 Comparison Results of the $FCDCM_{GK}$ PCA and RST Based Approaches 120
7.2 Comparison Results of the Rough $FCDCM_{GK}$ Versions 122
7.3 Comparison Results of the Fuzzy-Rough $FCDCM_{GK}$ Algorithms 125
7.4 Comparison Results in Terms of Running Time (s) 126
7.5 Comparison Results of the COID-FLA-$FCDCM_{GK}$ and Its Basic Approaches 132
7.6 Comparison in Terms of Running Time (s) 133

List of Algorithms

1.1 The Dendritic Cell Algorithm 13
2.1 The Fuzzy-Rough QuickReduct Algorithm 30
3.1 The QuickReduct Algorithm 49
5.1 The Fuzzy c-means Algorithm 77
5.2 The Gustafson-Kessel Algorithm 78
5.3 The Fuzzy Adaptive C-Shell Algorithm 79
7.1 The COID-FLA-$FCDCM_{GK}$ Algorithm 129

Introduction

The immune system is highly distributed, highly adaptive, self-organizing in nature, maintains a memory of past encounters, and has the ability to continually learn about new invaders. From a computational viewpoint, the immune system has much to offer by way of inspiration. Recently, there has been growing interest in the use of the natural immune system as inspiration for the creation of novel approaches to computational problems. This field of research is referred as Immunological Computation (IC) or Artificial Immune System (AIS) (Eiben & Smith., 2007).

At the heart of the human immune system is its ability to discriminate between the body's own cells, the self elements, and the foreign cells, non-self elements, by a process called "self-nonself discrimination". This fundamental process is performed by several AIS algorithms. However, it was noted that these algorithms cannot produce the same high performance as the human immune system. Hence, in (Matzinger, 2001), a relatively newer immunological discovery using more rigorous and up-to-date immunology, known as the "Danger Theory" (DT), has been proposed. The DT states that the recognition of a pathogen is based on environmental context rather than the simple self-nonself principle.

The DT is now developing in a way that aims at designing more efficiently a new foundation of artificial immune systems known as the "2nd generation AISs" (Matzinger, 2001). The DT, which is a new idea challenging the classical self-nonself viewpoint and which has become popular amongst immunologists, clarifies that the immune system, instead of discriminating between self and non-self, looks for danger producing elements and events. More precisely, the DT is based on the behavior of special immune cells called the Dendritic Cells (DCs). An inspiration from the behavior of the DCs led to the development of an immune algorithm termed the "Dendritic Cell Algorithm" (DCA) (Greensmith & Aickelin, 2005).

DCA, as a binary classifier, has been successfully applied to various applications. This is due to the worthy characteristics expressed by the algorithm as it exhibits several interesting and potentially beneficial features for binary classification problems.

Firstly, the algorithm performs correlations linking the identified anomalies to their potential causes (Greensmith & Aickelin, 2008b; Greensmith, Aickelin, & Tedesco, 2010).

DCA is, also, used as a classifier for machine learning data sets, where it was shown that the algorithm can process data classification while generating high and satisfactory classification results in comparison to other state-of-the-art classifiers (Greensmith & Aickelin, 2005). Current researches with the DCA have suggested that the algorithm shows not only excellent performance on detection rate, but also promise in reducing the number of false positive errors shown with similar systems (Greensmith & Aickelin, 2006a, 2007).

Despite of the above mentioned DCA interesting characteristics, the DCA is still a fairly recent algorithm whose algorithmic characteristics have not yet been fully explored to reveal its strengths and weaknesses. This led us to analyze and understand the DCA algorithmic aspects while proposing solutions to some DCA limitations extracted from the examinations carried out on the DCA. Throughout this dissertation, several investigations of the algorithm have been conducted to further understand and explore its various properties.

Aim and Scope

As mentioned, the dendritic cell algorithm is a new foundation in the field of artificial immune systems. As a new paradigm, more works have to be done on the algorithm in order to check its properties and its performance as a binary classifier. In this context, the main objective of this Thesis is to investigate a number of algorithmic properties of the DCA. This is to improve the dendritic cell algorithm's applicability and accessibility to future users. With respect to this objective, a set of the main contributions are presented within this dissertation and they are listed as follows:

We perform a detailed study of the dendritic cell algorithm behavior. While doing so, we notice that the algorithm has two main critical limitations. The first limitation is linked to the DCA data pre-processing phase. The second limitation is related to the algorithm restriction as it should be applied to ordered classes; i.e., the DCA is sensitive to the input class data order. In order to obtain satisfactory classification results, the DCA should be applied to ordered classes; only. More precisely, these two shortcomings and the proposed solutions are highlighted as follows:

- Automated pre-processing: While investigating the DCA data pre-processing phase, we notice that this phase is not robust as it is based on the use of the Principal Component Analysis technique (PCA). In fact, applying PCA does not guarantee that the first selected principal components that capture most of the variance will be the most adequate features to retain (Cantú-Paz, 2004). Furthermore, applying PCA destroys the underling meaning behind the features present in the used data set. However, losing the semantics of the features contradicts the specificity of the DCA as it is important to know the source (feature) of each signal category based on its meaning. Therefore, we focus on studying the DCA data pre-processing phase while proposing new robust automated pre-processing modules. To achieve this, we use Rough Set Theory and Fuzzy Rough Set Theory as powerful feature extraction techniques in the DCA data pre-processing phase.

- Sources of the DCA sensitivity to the input class data order: As previously stated, DCA was successfully applied to a diverse range of applications while generating satisfactory classification results. However, it was noticed in (Greensmith & Aickelin, 2008b) that the DCA is sensitive to the input class data order. To guarantee satisfactory classification results, the algorithm should be applied to ordered classes. To deal with this restriction, we focus our investigations on understanding and analyzing this DCA main limitation while

proposing adequate solutions. To handle this DCA issue, we propose solutions based on the use of Fuzzy Set Theory, a maintenance database technique and a fuzzy clustering technique.

Thesis Outline

This Thesis is structured into two main parts.

The first part, Theoretical Aspects, is composed of two main Chapters which are the following:

- Chapter 1: The Danger Theory. This Chapter gives the necessary background regarding the basic concepts of the danger theory with a special focus on the dendritic cell algorithm.
- Chapter 2: Rough Sets and Fuzzy-Rough Sets for Feature Selection. This Chapter involves the main tools and techniques used across the development of our new DCAs throughout this dissertation.

The second part of this Thesis presents our contributions. We can decompose this part into three main blocks. The first block is composed of two Chapters dealing with new developed danger theory approaches based on robust data pre-processing modules:

- Chapter 3: A Rough DCA Pre-Processing Module. In this Chapter, we propose a first module for data pre-processing based on rough set theory.
- Chapter 4: A Fuzzy-Rough DCA Pre-Processing Module. In this Chapter, we develop a second module for data pre-processing based on fuzzy rough set theory as an extension to the work presented, previously, in Chapter 3.

Chapter 3 proposes a more robust data pre-processing module, based on rough set theory, in comparison to the traditional DCA data pre-processing module. Yet, the application of rough set theory has some restrictions as it has to perform data discretization before feature selection. Thus, to avoid this task, we may extend our work proposed in Chapter 3. The proposed solution for such an extension is to apply fuzzy rough set theory as a pre-processor. This was presented in Chapter 4.

The second block is composed of two Chapters and deals with the development of new DCAs handling the problem of the algorithm sensitivity to the input class data order. We hypothesize that the causes of such sensitivity may be either tight to the DCA context assessment phase, first hypothesis, or to the DCA detection phase, second hypothesis. On this light, we propose to investigate the first hypothesis in Chapter 5 and the second hypothesis in Chapter 6.

- Chapter 5: An Automated Fuzzy Dendritic Cell Classification Method. In this Chapter, we define a new DCA based on fuzzy set theory.
- Chapter 6: The Fuzzy Maintained Dendritic Cell Classification Method. In this Chapter, we develop a fuzzy DCA based on a maintenance technique.

Lastly, the third block is seen as the fruit of all the previously studied parts. This block deals with the development of a more general DCA taking all the studied facts into account. The final fruit is a more robust danger theory classifier based on dendritic cells within an imprecise framework, and is presented in Chapter 7.

- Chapter 7: The Hybrid Automated and Maintained Fuzzy Dendritic Cell Immune Classifier. Based on the studies performed on the previous Chapters and based on the obtained notes, in Chapter 7 we develop a global automated-maintained fuzzy DCA. The proposed classifier is based on fuzzy-rough set theory, on fuzzy set theory, on a fuzzy clustering technique and on a maintenance signal base technique.

Finally, our dissertation ends with a conclusion and an Appendix. The conclusion summarizes all the work presented in this report and proposes further works to be done with the dendritic cell algorithm.

An Appendix presenting the main concepts of fuzzy set theory is provided since the latter theory is used to develop our fuzzy DCA versions.

Part I

Theoretical Aspects

Part I presents the theoretical aspects of this Thesis. It provides the necessary background regarding the basic concepts of the danger theory and more precisely the dendritic cell algorithm. Besides, it introduces the theories of rough sets and fuzzy-rough sets as main techniques adopted in this dissertation.

Chapter 1

The Danger Theory

1.1 Introduction

Artificial Immune Systems (AISs) can be defined as a computational paradigm that is inspired by theoretical immunology, observed immune functions, principles and mechanisms (Zuben & Timmis, 2003). The latest immunological theory in the AIS field is called the "Danger Theory" (DT) (Matzinger, 2001). The DT which has become popular among immunologists is based on the behavior of special immune cells called the "Dendritic Cells" (DCs). An inspiration from the behavior of these special immune cells led to the development of an immune algorithm termed the "Dendritic Cell Algorithm" (DCA).

In this Chapter, first, we start by presenting the artificial immune system field. Next, we detail the basics of the danger theory. Finally in Section 1.4, we describe the dendritic cell algorithm including its biological principals.

1.2 Artificial Immune Systems

The study and design of artificial immune systems represent a relatively new area of research that tries to build computational systems that are inspired by the natural immune system. In this Section, we give a brief overview of the main natural immune system functionalities on which the AIS is based.

1.2.1 Immunological Concepts

The human immune system is a network of cells, tissues, and organs that work together to defend the body against attacks by "foreign" invaders that are trying to do it harm (Goldsby, Kindt, Osborne, & Kuby, 2003). This main task is achieved thanks to its capability to recognize the presence of infectious foreign cells and substances, known as "non-self" elements and to respond to them by eliminating them or neutralizing them. This distinction between the

"non-self" and the body's "self" cells is based on a process called "self-non-self discrimination" (Janeway, 1992). The non-self elements, also called "antigens", are mainly microbes; tiny organisms such as bacteria, parasites, and fungi. All of these can, under the right conditions, cause damage and destruction to parts of the body and if these were left unchecked, the human body would not be able to function appropriately. Thus, it is the purpose of the immune system to act as the body's own army. More precisely, the immune system does not rely on one single mechanism to deter invaders, but instead uses many strategies. The main division between the strategies is that between innate immunity and adaptive immunity.

The innate immunity is our first line of defense system against invading antigens. It is those parts of the immune system that work no matter what the damage is caused by. They are always at work and do not need to have seen the offending invader before to be able to start attacking it. The innate immune system includes anatomical barriers, secretory molecules and cellular components. In addition, the innate immune system employs a different group of cells, e.g. Antigen Presenting Cells (APCs), to eliminate threats or to interact with the rest of the immune system.

Our second line of defense is the adaptive immune system which affords protection against re-exposure to the same pathogen. The adaptive immune system is called into action against pathogens that are able to evade or overcome innate immune defenses. The cells of the adaptive immune system are mainly the B-cells and the T-cells, but there are also other important parts of the adaptive immune system, such as the complement cascade and the production of antibodies. These mentioned elements of the immune system do not work separately, but all work together in a co-operative fashion. If they have to work effectively then they need a good system for communicating messages. This system is provided by specific released immune proteins called the "cytokines".

Although these two arms of the immune system have distinct functions, there is an interplay between them. This linkage is managed by a group of APCs called "Dendritic Cells" (DCs) (Lutz & Schuler, 2002).

1.2.2 Synopsis of Artificial Immune Systems

An inspiration from the remarkable properties expressed by the natural immune system, seen above, leads to the conception and the design of artificial immune systems exhibiting similar functionalities. These systems are discussed in what follows.

Clonal Selection Theory

Clonal selection theory is used to clarify the basic response of the adaptive immune system to antigenic stimulus. Clonal selection involves two processes which are pattern recognition and selection. More precisely, it establishes the idea that only those cells capable of recognizing an antigen will proliferate while other cells are selected against. Clonal selection calls both B and T cells. In fact, when B-cells antibodies bind with an antigen, cells become activated and differentiated either to be plasma cells or memory cells. Plasma cells make large amounts of a specific antibody that work against a specific antigen to destroy it. Memory cells remain with the host and promote a rapid secondary response. However, before this process, clones of B cells are produced and undergo somatic hypermutation. Consequently, diversity is introduced into the B cell population (Castro & Zuben, 2001).

Based on this theory, various AIS algorithms have been proposed in literature and most of them are devoted to optimization problems. A detailed description and comparison of AIS clonal selection algorithms can be found in (Brownlee, 2007).

Immune Network Theory

The immune system is a network of cells and antibodies that have a profound sense of self and the ability to remember and learn. The immune network theory states that the "recognizers" of the immune system, the B-cells and antibodies, not only recognize foreign particles but also recognize and interact with each other. This created network is based on interconnected B-cells for antigen recognition. The strength of the B-cells connections is directly proportional to the affinity that they share. Indeed the B-cells can both stimulate and suppress each other in order to stabilize the network.

Basic concepts of the immune network theory are implemented leading to several immune algorithms dedicated to data analysis, unsupervised clustering and visualization. A detailed study of some immune network algorithms can be found in (Whitbrook, Aickelin, & Garibaldi, 2010).

Self-Non-Self Theory

The self-non-self theory is able to tell the difference between what is foreign and potentially harmful, and what is actually a part of its own system. The representative self-non-self theories are the negative selection and the positive selection. The purpose of the negative selection theory is to provide tolerance for self cells. During the generation of T-cells, receptors are made through a pseudo-random genetic rearrangement process. Then, they undergo a censoring process in the thymus, called the negative selection. There, T-cells that react against self-proteins are destroyed; thus, only those that do not bind to self-proteins are allowed to leave the thymus. These matured T-cells then circulate throughout the body to perform immunological functions and protect the body against foreign antigens. As for the positive selection theory, it works as the opposite of the negative selection process (Gonzalez, 2003).

An inspiration from the negative selection and positive selection theories gave rise to numerous AIS algorithms which are mainly used for classification.

Danger Theory

As mentioned, several AIS algorithms are based on the "self-non-self discrimination" theory. However, in (Matzinger, 2001), it was noticed that these algorithms cannot produce the same high performance as the human immune system. Hence, the Danger Theory (DT) has been proposed. The DT states that the recognition of a pathogen is based on environmental context (signals) rather than the simple self-non-self principle. The DT which is dedicated to classification problems is based on the functioning of the dendritic immune cells. An inspiration from these cells gave rise to the development of the Dendritic Cell Algorithm (DCA) (Greensmith & Aickelin, 2005). The DCA is based on the principal that dendritic cells classify each antigen as being either anomalous or normal. The DCA has been successfully applied to a diverse range of applications where it was shown that the algorithm can process data classification while generating high and accurate classification results.

The objective of this Thesis is to focus on this new AIS promising field and more specifically to investigate the DCA classification algorithmic properties.

1.3 Danger Theory Perception

1.3.1 Fundamental Concepts

The main goal of the immune system is to protect the body from invading entities which cause damage and diseases. Initially, immunologists believe that the protection was done by distinguishing self and non-self inside the body and by eliminating the non-self. However, it was noticed that current artificial immune systems were unable to scale up to real world requirements and this is caused by their reliability on the self-non-self discrimination. Immunologists are increasingly finding faults with the traditional self-non-self thinking and a new danger theory is emerging. The DT points out that there must be a discrimination happening that goes beyond the self-non-self distinction, for instance:

- There is no immune reaction to foreign bacteria in the gut or to the food we eat although both are foreign entities.
- The human body changes over its lifetime and, thus, self changes as well. Therefore, the question arises whether defences against non-self learned early in life might be auto-reactive later.
- Other aspects that seem to be at odds with the traditional viewpoint are autoimmune diseases; certain types of tumors that are fought by the immune system and successful transplants.

The DT offers an alternative to the self-non-self discrimination approach. It stipulates that the immune system actually discriminates some self from some non-self. The immune response is done by reaction to a danger and not to a non-self entity.

1.3.2 Danger Description

Among the definitions for “danger” proposed in the immunological glossary, is the following one: “danger is anything that causes cell stress or lytic cell death” (Cayzer & Aickelin, 2002). Cell death can be found in different parts of the body. Nevertheless, this kind of cell death which does not appear dangerous to the immune system is normal and it is called “apoptosis”. The dying cells caused by apoptosis are scavenged by specialized cells; i.e., the APCs. Immunologically, in case of apoptosis, cells send out signals to the nearby scavenging APCs to devour them. This phenomenon prevent the dying cell from releasing harmful toxins (see Figure 1.1(a)).

Another type of cell death is called “necrosis” which means that cells get killed accidentally by harmful pathogens. In this case, the cell death is not organized. This disorderly death does not send signals which inform the nearby phagocytes to engulf the injured cells. This, makes it hard for the cleanup cells (phagocytes) to locate and digest the cells that die due to necrosis. The cell membrane stores special digestive enzymes. Thus, the release of this harmful toxin accelerates the unorganized chemical reaction (see Figure 1.1(b)).

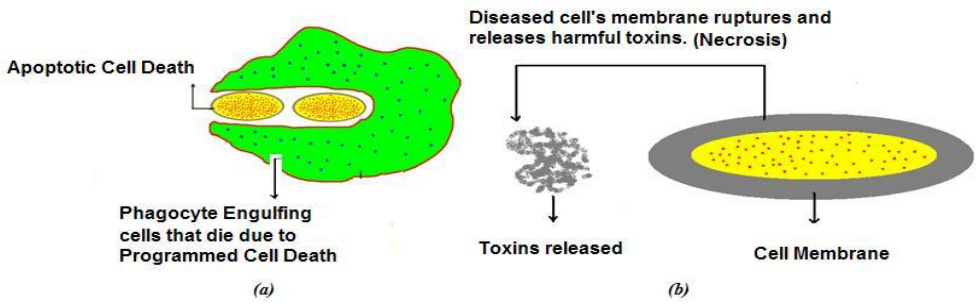

Figure 1.1: Apoptosis and Necrosis Illustrations

DT was built on the concept that the contents released by any damaged cell were actually a form of danger signals that alerted the nearby APCs - the dendritic cells - and activated them. Only cells that die due to necrosis would send out alarm signals. Healthy cells and cells that die due to apoptosis should not.

1.3.3 Immunological Signals Taxonomy

Upon cell death, cells release specific signals that urge the behavior of the system since they are a reflection of the state of the environment. There are four main categories of released signals:

- **Safe Signals (SSs):** SSs are released as a result of apoptosis. They are indicators of normality which means that the antigen collected by the DC was found in a normal context. Hence, tolerance is generated to that antigen.
- **Pathogen Associated Molecular Pattern Signals (PAMPs):** PAMPs are essential molecules produced by microbes but not produced by the host. They are definite indicators of abnormality indicating the presence of a non-host entity.
- **Danger Signals (DSs):** DSs are signals released as a result of necrosis. They are indicators of abnormality but with lower value of confidence than PAMP signals.
- **Inflammatory Cytokines (CKs):** CKs are signals proving that there is an increase in temperature in the affected tissue. Inflammation signals have the effect of amplifying the other three categories of input signals but they have no efficiency when they are present alone in the system.

1.4 The Dendritic Cell Algorithm

As previously stated, the most prominent players of the DT are the dendritic cells (DCs). An inspiration from the DCs behavior led to the development of an immune algorithm termed the dendritic cell algorithm (DCA) (Greensmith & Aickelin, 2005). Before describing the functioning of the algorithm, we give a general overview of the biological principles used by the DCA.

1.4.1 Biological Background

A DC is a type of antigen-presenting cell. DCs are in charge of catching, processing and revealing antigens to T-cells. They, also, express receptors on their surfaces to receive signals from their neighborhood. The behavior of DCs depends on the concentration of the signals received. As a result, they differentiate into three different maturity levels described as follows (Lutz & Schuler, 2002):

- **Immature DCs:** On arrival in the tissue, DCs are found in an immature state. Here, immature DCs (iDCs) collect antigen which could be a "safe" molecule or something foreign. Furthermore, DCs can collect and sense the various signals that may be present in the tissue. Receipt of signals causes changes to the function, morphology and behavior of the iDC. In other words, the relative proportions and potency of the different signals lead to a full or partial maturation state of iDCs.
- **Mature DCs:** For an iDC to become a "mature DC" (mDC), the iDC must be exposed to a greater quantity of either PAMPs or DSs than SSs. Sufficient exposure to PAMPs and DSs causes the DC to cease antigen collection and migrate from the tissue to the lymph node. Most importantly, mDCs produce an inflammatory cytokine called "interleukin-12" which stimulates T-cell activation in order to be reactive to antigen presentation. Additionally, mDCs produce costimulatory molecules (CSMs) which are known to facilitate the antigen presenting process.
- **Semi-mature DCs:** In the presence of apoptosis conditions, exposure to SSs diverts the iDC to become a "semi-mature DC" (smDC). smDCs appear morphologically very similar to mDCs and can present antigen, yet they do not have the ability to activate T-cells. The smDC produces "interleukin-10" which suppresses T-cells matching the presented antigen. Antigens collected with SSs are presented in a tolerogenic context and lead to unresponsiveness to those antigens.

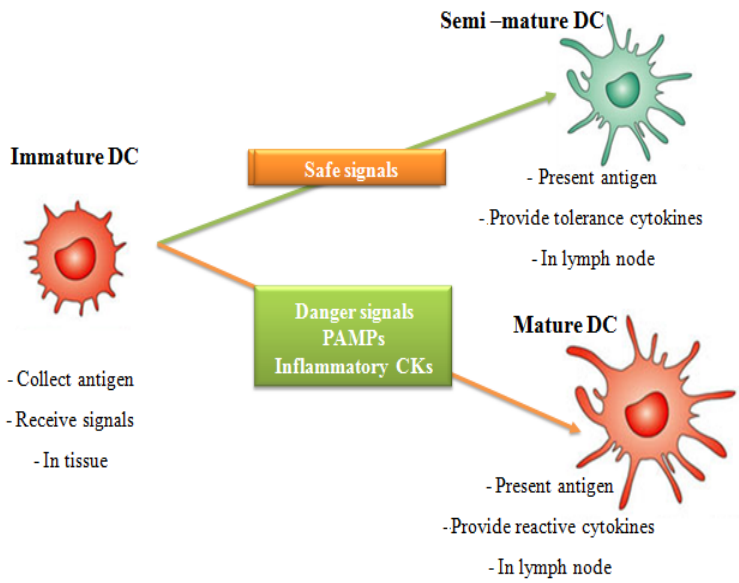

Figure 1.2: DCs Behavior

Swapping from one DC state to another is dependent upon the receipt of different signals throughout the initial state (iDC). This is shown in Figure 1.2. The migration to the mature state or to the semi-mature state depends on the

concentration of the input signals received from the environment. Immunologically, if the concentration of SSs is greater than the other categories of signals then DCs migrate to the semi-mature state. However, if the concentration of PAMPs and DSs is greater than the concentration of SSs then DCs migrate to the mature context.

1.4.2 Algorithmic Details

To transform the abstract model of DC biology into an immune-inspired algorithm, it must be formalized into the structure of a generic algorithm and into a set of logical processes. A generic form of the DCA binary classifier is described by Algorithm 1.1.

Algorithm 1.1 The Dendritic Cell Algorithm

```
 1: input: signals from all categories and antigens;
 2: output: antigens plus context values;
 3: for each DC do                    /* Pre-processing & Initialization phase */
 4:    initialize DC;
 5: end for
 6: while CSM output signal < migration threshold do        /* Detection phase */
 7:    get antigens;
 8:    store antigens;
 9:    get signals;
10:    calculate interim output signals;
11:    update cumulative output signals;
12: end while
13: cell location update to lymph node;
14: if semi-mature output > mature output then        /*Context Assessment phase */
15:    cell context is assigned as 0;
16: else
17:    cell context is assigned as 1;
18: end if
19: print collected antigens plus cell contexts;
20: for all antigens in total list do                        /* Classification phase */
21:    increment antigen count for this antigen type;
22:    if antigen context equals 1 then
23:        increment antigen type mature count;
24:    end if
25: end for
26: for all antigen types do
27:     MCAV of antigen type = mature count / antigen count;
28: end for
```

For the ease of analysis, the algorithm is divided into the following four main phases:

1. *Pre-Processing & Initialization phase* - Line 1 to Line 5;
2. *Detection phase* - Line 6 to Line 13;

3. *Context Assessment phase* - Line 14 to Line 19;
4. *Classification phase* - Line 20 to Line 28;

Pre-Processing and Initialization Phase

The application of the DCA often requires a data pre-processing phase to appropriately map a given problem domain to the input space of the algorithm. The pre-processing stage involves two main steps: feature reduction and signal categorization. The process of feature reduction and signal categorization involves selecting/extracting the most interesting features from the original feature set of a given problem, and then categorizing each of these derived features into one of the defined signal categories of the DCA; i.e., PAMP, DS or SS. Some DCA works deal with involving users or experts in order to select or extract the most interesting features and map them into the appropriate signal categories. Other DCA works apply, principally, the Principal Component Analysis (PCA) statistical method for an automated data pre-processing task. More precisely, for feature reduction, DCA applies the PCA that selects the first "principal components" which reveal the internal structure of the given data with the focus on data variance. Once features are selected, PCA is applied to assign each attribute to its specific signal type.

For signal categorization, some DCA versions use one attribute among the selected set of features and assign it to both PAMP and SS as they are both considered as positive indicators of an anomalous and normal signal. Using one attribute for these two signals requires a threshold level to be set: values greater than this can be classed as SS otherwise as PAMP. As for the DS attribute assignment and since the DS is less than certain to be anomalous, the combination of the rest of the selected attributes are chosen to represent it. Other DCA works handle the signal categorization step using the PCA ranking procedure. This is performed by the use of the PCA attributes' ranking in terms of variability. Once ranking is performed, the attributes are mapped into the DCA input signal categories by correlating the PCA ranking with the ranking of signal categories which is in the order Safe, PAMP, and Danger (GU, Oates, Greensmith, & Aickelin, 2009).

Detection Phase

Once the DCA data pre-processing stage is achieved, the algorithm calculates the SS, PAMP and DS values (Greensmith, 2007) inducing a signal database and adheres these signals and antigen to fix the context of each DC. This is preformed during the detection phase. In fact, the input signals of the system which are pre-categorized as "PAMP", "danger" and "safe" are processed by the algorithm in order to get three output signals: costimulation signal (CSM), semi mature signal (smDC) and mature signal (mDC). The detection phase occurs within DCs of the immature state. The DC has the following three functions which are performed each time a single DC is updated:

1. *Sample antigen:* the DC collects signals and antigens from an external source (in this case, from the tissue) and places the antigen in its own antigen storage data structure (Line 7 to Line 9).
2. *Update input signals:* the DC collects values of all input signals present in the signal storage area.
3. *Calculate interim output signals:* at each iteration, each DC calculates three temporary output signal values from the received input signals. These signals are used to assess the state of the DC upon termination of the detection phase of a DC's life span. The three output signals of a DC perform two roles, to determine if an antigen type is anomalous and to limit the time spent sampling data.

To calculate the interim output signals, DCA applies the following weighted sum equation:

$$C_{[CSM,smDC,mDC]} = \frac{((W_{PAMP} * \sum_i PAMP_i) + (W_{SS} * \sum_i SS_i) + (W_{DS} * \sum_i DS_i))}{(W_{PAMP} + W_{SS} + W_{DS})} * \frac{1+I}{2} \tag{1.1}$$

Assuming that there are multiple signals per category, $PAMP_i$, DS_i and SS_i are the input signal values of category PAMP, danger and safe for all signals (i) of that category. W_{PAMP}, W_{SS} and W_{DS} represent the weights used for PAMP, SS and DS, respectively. I represents the inflammation signal. This equation is repeated three times, once per output signal. This is to calculate the interim output signal values for the CSM output, the smDC output and the mDC output. These values are cumulatively summed over time (Greensmith, 2007). The weights used by the DCA are either derived empirically from the data or are user defined values.

Each DC in the population is assigned a migration threshold value upon its creation. Following the update of the cumulative output signals, a DC compares the value it contains for CSM with the value it is assigned as its migration threshold. If the value of CSM exceeds the value of the migration threshold then the DC is removed from the sampling area and its life span is terminated.

Context Assessment Phase

Once the cell has migrated and through the context assessment phase, each DC has the ability to process and collect signals and antigens. Through the generation of cumulative output signals, the DC forms a cell context that is used to perform anomaly detection in the assessment of antigens. In fact, upon migration, the cumulative output signals are assessed and the greater of semi-mature or mature output signal becomes the cell context. This cell context is used to label all antigens collected by the DC with the derived context value of 1 or 0. This information is ultimately used in the generation of an anomaly coefficient which will be deal with in the final step; i.e., the classification phase.

Classification Phase

The derived value for the cell context is used to derive the nature of the response by measuring the number of DCs that are fully mature and is represented by the Mature Context Antigen Value (MCAV). The MCAV is used to assess the degree of anomaly of a given antigen. The closer the MCAV is to 1, the greater the probability that the antigen is anomalous. By applying thresholds at various levels, analysis can be performed to assess the anomaly detection capabilities of the algorithm. Those antigens whose MCAVs are greater than the anomalous threshold are classified into the anomalous category while the others are classified into the normal one.

1.4.3 DCA: An Example

This example consists of applying DCA to the problem of credit management. An extract of the data set for this example is presented in Table 1.1.

Table 1.1: Bank Database

Client	Age	Income	Number of credit cards	Duration of the loan	Credit
Client1	24	650	1	30	no
Client2	30	1000	3	10	no
Client3	36	1300	3	8	yes
Client4	20	600	1	20	no
Client5	32	900	2	13	yes

The dendritic cell algorithm selects, first of all, some attributes and pre-categorizes them as PAMP, DS and SS. Then, the obtained data set is transformed into a signal data set. An extracted set of the signal database is illustrated in Table 1.2.

Table 1.2: Signal Data Set

Client (antigen)	PAMP	SS	DS
Client1	100	100	0
Client2	0	0	100
Client3	20	50	40

To show the calculations under different input signal conditions, three iterations (cycles) with three sets of signals are shown. The derived output signal values are used to demonstrate how to perform the MCAV calculation for three different antigen types (Ag1, Ag2 and Ag3). In this example, three DCs are required, one for each iteration, termed DC1, DC2 and DC3 for the purpose of identification. Each DC is assigned an identical migration threshold value (*tm*) which is set to 100. The sets of signals used in this example are presented in Table 1.2. The signal processing equation is the following:

$$C_{[CSM,smDC,mDC]} = (W_{PAMP} * PAMP) + (W_{SS} * SS) + (W_{DS} * DS)$$

The weights are presented in Table 1.3.

Table 1.3: Example of Weights Used for Signal Processing

	PAMP	SS	DS
CSM	2	1	2
smDC	0	0	1
mDC	2	1	- 1.5

The worked example is performed in the following itemized list:

1. We assume that the antigen vector (A) is the following:
   A = {Ag1; Ag1; Ag1; Ag1; Ag1; Ag2; Ag2; Ag2; Ag2; Ag3; Ag3; Ag3}

2. Cycle l = 0:
   DC samples antigens, so DC1 a(m) = {Ag1; Ag1; Ag1; Ag2; Ag2}
   DC samples input signals, so DC1 s(m) = {100; 100; 0}
   DC calculates output signals, so DC1 outputs:
   C_{CSM} = (100 * 2) + (100 * 1) + (0 * 2) = 300
   C_{smDC} = (100 * 0) + (100 * 0) + (0 * 1) = 0
   C_{mDC} = (100 * 2) + (100 * 1) + (0 * -1.5) = 300
   For DC1, t(m) = 100, therefore this DC has now exceeded its migration threshold as the value for C_{CSM} is greater than t(m). Also, $C_{smDC} < C_{mDC}$ and therefore DC1 is assigned a cell context value of 1 indicating that its collected antigens may be anomalous.

3. By removing the antigens used by DC1, the antigen vector now consists of: A = {Ag1; Ag1; Ag2; Ag2; Ag3; Ag3; Ag3}

4. Cycle l = 1:
   DC samples randomly antigens, so DC2 a(m) = {Ag2; Ag2; Ag1}
   DC samples input signals, so DC2 s(m) = {0; 0; 100}
   DC calculates output signals, so DC2 outputs:
   C_{CSM} = (0 * 2) + (0 * 1) + (100 * 2) = 200
   C_{smDC} = (0 * 0) + (0 * 0) + (100 * 1) = 100
   C_{mDC} = (0 * 2) + (0 * 1) + (100 * -1.5) = -150
   For DC2, t(m) = 100, therefore this DC has now exceeded its migration threshold as the value for C_{CSM} is greater than t(m). Also, $C_{smDC} > C_{mDC}$ and therefore DC2 is assigned a cell context value of 0 indicating that its collected antigens are likely to be normal.

5. The antigen vector now consists of: A = {Ag1; Ag3; Ag3; Ag3}

6. Cycle l = 2:
   DC samples antigens, so DC3 a(m) = {Ag1; Ag3; Ag3; Ag3}
   DC samples input signals, so DC3 s(m) = {20; 50; 40}
   DC calculates output signals, so DC3 outputs:
   C_{CSM} = (20 * 2) + (50 * 1) + (40 * 2) = 170
   C_{smDC} = (20 * 0) + (50 * 0) + (40 * 1) = 40
   C_{mDC} = (20 * 2) + (50 * 1) + (40 * -1.5) = 30
   For DC3, t(m) = 100, therefore this DC has now exceeded its migration threshold as the value for C_{CSM} is greater than t(m). Indeed, $C_{smDC} > C_{mDC}$ and therefore DC3 is assigned a cell context value of 0.

7. Now antigens can be analyzed and the derived MCAV coefficients are shown in Table 1.4.

Table 1.4: Worked Example of MCAV Output

Antigen Type	num presentations	num mature presentations	MCAV
Ag1	5	3	0.6
Ag2	4	2	0.5
Ag3	3	0	0.0

8. To perform anomaly detection, a threshold must be applied to the MCAVs. This threshold is a user defined parameter which requires some expert knowledge to define and is specific to the application. In this case, the

anomaly threshold is defined by the bank manager and is set to 0.47. Therefore, client1 (Ag1) and client2 (Ag2) are classed as anomalous (they are not allowed to have a credit). However, client3 (Ag3) is classified as normal.

1.4.4 Development Pathway

Different proposals have been developed to extend the danger theory and more precisely the DCA. Extensive researches have been carried out to modify the standard principals of the DCA to simplify it by removing or replacing some of its components. Other researches have been fulfilled to combine the DCA with other theories; mainly the theory of fuzzy sets. This combination aims at handling the imprecision found in danger theory principals. Thus, the DCA has undergone many revisions since its original inception resulting in multiple versions of the algorithm. These revisions may be categorized into two main branches: DCAs under a crisp environment and DCAs under an imprecise environment.

DCAs under a Crisp Environment

The two main DCA versions, developed under a crisp environment, are the standard or classical version of the DCA (Greensmith & Aickelin, 2005) and the deterministic version of the DCA (dDCA) (Greensmith & Aickelin, 2008b). The main difference between the two is that the dDCA eases the calculation of the DCA crucial algorithmic information. dDCA was, also, modified in (Musselle, 2010) where the author used a Markov chain to generate synthetic data sets to investigate the antigen sample specificity of the dDCA.

Other modified versions have, also, been presented which have tended to either add additional components to the algorithm or differ in their methods of implementation. For instance, in (Greensmith & Aickelin, 2009), an additional step which is applying segmentation was added to the DCA. This involves segmenting the output of the DCA into slices enabling the system to perform periodic analysis on the processed information presented by the DCs. As a result, it can effectively improve the anomaly detection speed.

These DCA versions were successfully applied to a wide range of applications such as anomaly detection (Greensmith & Aickelin, 2006b), syn scan detection (Greensmith & Aickelin, 2007) and bot detection (Greensmith & Aickelin, 2008a).

DCAs under an Imprecise Environment

The DCA as a binary classifier gives satisfactory results in a context where data are known with certainty and precision. However, the reality is connected to uncertainty and imprecision by nature. Such imperfection may affect the classification performance of the DCA. Therefore, one idea was to combine theories managing imprecision, mainly fuzzy set theory, with the dendritic cell algorithm to deal with imprecise contexts. Works such as (Fu & Li, 2008; Fu & Zhang, 2009) focused on various aspects such as the definition of imprecise terms like the term “danger”, the smoothness of some applied crisp hypotheses like the migration threshold and replacing it with a fuzzy one, etc.

Another fuzzy DCA model was proposed in (Chelly & Elouedi, 2010) that sheds light on the standard DCA context assessment phase. The developed algorithm proposes a novel fuzzy context assessment phase by smoothing the crisp separation between the two DCs contexts; i.e., semi-mature and mature.

1.4.5 Strengths and Weaknesses of the DCA

Strengths of the DCA

The DCA as well as its extensions were successfully applied to a wide range of applications. This is due to the worthy characteristics expressed by the DCA as it exhibits several interesting and potentially beneficial features for binary classification problems.

Firstly, the algorithm performs temporal correlation that links identified anomalies to their potential causes (Greensmith & Aickelin, 2006a; Greensmith et al., 2010). Another advantage is that the DCA is capable of generating the Mature Context Antigen Value (MCAV) for each antigen; i.e., the anomaly coefficient value. Based on these values, the user or expert may take adequate decisions on how to react against the seen antigen; i.e., data item.

Another addendum to these DCA strengths may be mentioned which is the following: DCA as a binary classifier was compared to several state-of-the-art classifiers and had shown promising results in terms of classification accuracy (Greensmith et al., 2010).

Weaknesses of the DCA

Despite of the DCA's advantages, there also exist some weaknesses of the algorithm that limit its applicability to problems and accessibility to users. The main DCA limitations are discussed as follows:

The first criticism of the DCA is related to its data pre-processing phase. The DCA data pre-processing phase is performed either manually by users based on their expert knowledge of a given problem domain, either automatically using basically the PCA. In case where the pre-processing stage is performed manually; this causes the algorithm being application dependent. In case where the pre-processing phase is performed using the PCA; this does not guarantee that the first selected principal components will be the most adequate features for classification. Furthermore, the DCA categorization process is based on the PCA attributes' ranking in terms of variability. However, this categorization process could not be considered as a coherent categorization procedure.

Another limitation of the algorithm is linked to its used weights for signal transformation, studied in the algorithm detection phase. These weights are either statically assigned or are user defined parameters. This could extensively constrain the algorithm's ability to fully adapt to a given problem domain. Another criticism of the DCA is the mandatory respect of the data order in the used databases. In other words, the algorithm should only be applied to ordered data sets, all class 1 followed by all class 2 data items, to obtain satisfactory classification results.

In addition, the application of the DCA is essentially restricted to binary classification problems. Since its

development, the algorithm has been only applied to binary classification problems where one class is “normal” and the other is “anomalous”. This restriction is due to the algorithm’s foundation related to the fact of classifying each antigen either as a normal item or as a dangerous one.

1.5 Conclusion

In this Chapter, we have presented the basic notions of the danger theory and basically the DCA. We have, also, discussed the major DCA works where we have clarified the strengths and the weaknesses of the algorithm. In the next Chapter, we will present the basics of rough set and fuzzy rough set theories as main tools to be used for the data pre-processing phase of our new DCA proposed versions.

Chapter 2

Rough Sets and Fuzzy-Rough Sets for Feature Selection

2.1 Introduction

Data reduction is a main point of interest across a wide variety of fields. In fact, focusing on this step is crucial as it often presents a source of significant data loss. Many techniques were proposed in literature to achieve the task of data reduction. However, most of them tend to destroy the underlying semantics of the features after reduction or require additional information about the given data set for thresholding. Thus, it seems necessary to think about a technique that can on the one hand reduce data dimensionality using information contained within the data set and on the other hand capable of preserving the meaning of the features. Rough Set Theory (RST) and Fuzzy-Rough Set Theory (FRST) can be used as such tools to discover data dependencies and to reduce the number of attributes contained in a data set using the data alone, requiring no additional information (Jensen, 2005).

This Chapter focuses on introducing RST and FRST for feature selection. In the next Section, a brief refresh on some tools for data reduction is proposed. In Section 2.3, the basic concepts of RST are highlighted and the fundamentals of FRST are presented next in Section 2.4.

2.2 Data Reduction Approaches

Dealing with high-dimensional data sets presents an urgent need for a set of tools to reduce data dimensionality. These techniques can be categorized into two heads: those that transform the underlying meaning of the features, called the *"transformation-based approaches"*, and those that are semantic-preserving techniques known as the *"selection-based approaches"*.

Transformation based approaches, also called *"feature extraction approaches"*, involve simplifying the amount

of resources required to describe a large set of data accurately. Feature extraction is a general term for methods that construct combinations of variables to represent the original set of features but with new variables while still describing the data with sufficient accuracy. The transformation based techniques are employed in situations where the semantics of the original database are not needed by any future process.

In contrast to the semantics-destroying dimensionality reduction techniques, the semantics-preserving techniques, also called *"feature selection techniques"*, attempt to retain the meaning of the original feature set. The main aim of this kind of techniques is to determine a minimal feature subset from a problem domain while retaining a suitably high accuracy in representing the original features.

It is important to mention that data dimensionality reduction techniques suffer from some limitations. Most of these techniques involve the user for parameterizing the algorithms and this is a significant drawback. Some feature selectors require noise levels to be specified by the user beforehand, some simply rank features leaving the user to choose their own subset. There are those that require the user to state how many features are to be chosen, or they must supply a threshold that determines when the algorithm should terminate. All of these require the users to make a decision based on their own (possibly faulty) judgment (Jensen, 2005). To overcome the shortcomings of the existing methods, it would be interesting to look for a method that does not require any external or additional information to function appropriately. Rough Set Theory (RST) and Fuzzy-Rough Set Theory (FRST) can be used as such tools.

2.3 Rough Set Based Approach for Feature Selection

In this Section, rudiments of the RST will be outlined and basic concepts of the theory will be illustrated by a simple tutorial example.

2.3.1 Decision and Information Systems

Data are represented as a table where each row represents an object and where each column represents an attribute that can be measured for each object. Such table is called an *"Information System"* (IS) or an *"Information Table"* (IT). To fit this definition to our research field, an information table can be seen as a representation of antigens which are defined via a set of attributes. Formally, an information system can be defined as a pair $IS = (U, A)$ where $U = \{x_1, x_2, \ldots, x_n\}$ is a non-empty, finite set of objects called the *"universe"* and $A = \{a_1, a_2, \ldots, a_k\}$ is a non-empty, finite set of *"condition"* attributes. In supervised learning, a special case of the defined information table is considered, called a *"Decision Table"* (DT) or a *"Decision System"* (DS). A DT is an information system of the form $IS = (U, A \cup \{d\})$, where $d \notin A$ is a distinguished attribute called *"decision"*. The value set of d, called $\theta = \{d_1, d_2, \ldots, d_s\}$.

Example 2.1 *Using the terminology of RST, the data set presented in Table 2.1 can be considered as an information system $IS = (U, A)$ consisting of 4 conditional features (a, b, c, d) and 8 objects. To illustrate an example of a decision system, Table 2.2 is used. It consists of 4 conditional features (a, b, c, d), 1 decision feature (e) and 8 objects.*

Table 2.1: Information System

$x \in U$	a	b	c	d
x_1	1	0	2	2
x_2	0	1	1	1
x_3	2	0	0	1
x_4	1	1	0	2
x_5	1	0	2	0
x_6	2	2	0	1
x_7	2	1	1	1
x_8	0	1	1	0

Table 2.2: Decision System

$x \in U$	a	b	c	d	e
x_1	1	0	2	2	0
x_2	0	1	1	1	2
x_3	2	0	0	1	1
x_4	1	1	0	2	2
x_5	1	0	2	0	1
x_6	2	2	0	1	1
x_7	2	1	1	1	2
x_8	0	1	1	0	1

2.3.2 Indiscernibility Relation

The indiscernibility relation is the mathematical basis of rough set theory. Every object of the universe is described by certain amount of information expressed by means of some attributes used for object description. Objects characterized by the same information are indiscernible in view of the available information about them. For every set of attributes $P \subset A$, an indiscernibility relation, denoted by $IND(P)$ or U/P, is defined in the following way: two objects, x_i and x_j, are indiscernible by the set of attributes P in A, if $p(x_i) = p(x_j)$ for every $p \subset P$. In other words, two objects are considered to be indiscernible or equivalent if and only if they have the same values for all attributes in the set.

The equivalence class of $IND(P)$ is called elementary set in P because it represents the smallest discernible groups of objects. For any element x_i of U, the equivalence class of x_i in relation $IND(P)$ is represented as $[x_i]_{IND(P)}$. For every object $x_j \in U$, we will use $a_i(x_j)$ to denote the value of a condition attribute a_i for an object x_j. Similarly, $d(x_j)$ is the value of the decision attribute for an object x_j. We further extend these notations for a set of attributes $P \subseteq A$, by defining $P(x_j)$ to be value tuple of attributes in P for an object x_j. The indiscernibility relation based on a subset of the condition attributes P, denoted by $IND(P)$, is defined as follows:

$$IND(P) = U/P = \{[x_j]_P | x_j \in U\} \tag{2.1}$$

$$where \;\; [x_j]_P = \{x_i | P(x_i) = P(x_j)\} \tag{2.2}$$

The indiscernibility relation based on the decision attribute d, denoted by $IND_{\{d\}}$, is defined as follows:

$$IND_{\{d\}} = U/\{d\} = \{[x_j]_{\{d\}} | x_j \in U\} \tag{2.3}$$

Example 2.2 *In order to illustrate how a decision table from Table 2.2 defines an indiscernibility relation, we consider the following three non-empty subsets of the conditional attributes: $\{a\}$, $\{b, c\}$ and $\{a, b, c\}$. The relation IND may define three partitions of the universe.*

$$IND(a) = \{\{x_1, x_4, x_5\}, \{x_2, x_8\}, \{x_3, x_6, x_7\}\}$$
$$IND(b, c) = \{\{x_3\}, \{x_1, x_5\}, \{x_4\}, \{x_2, x_7, x_8\}, \{x_6\}\}$$

$$IND(a, b, c) = \{\{x_1, x_5\}, \{x_2, x_8\}, \{x_3\}, \{x_4\}, \{x_6\}, \{x_7\}\}$$

If we take into consideration the set a, the objects x_1, x_4 and x_5 belong to the same equivalence class; they are indiscernible.

2.3.3 Lower and Upper Approximations

The rough set approach to data analysis hinges on two basic concepts, namely the lower and upper approximations of a set, referring to the elements that doubtless belong to the set, and to the elements that possibly belong to the set. Let $P \subseteq A$ and $X \subseteq U$. We can approximate X using only the information contained by constructing the P-lower and P-upper approximations of X, denoted $\underline{P}(X)$ and $\overline{P}(X)$ respectively where:

$$\underline{P}(X) = \{x | [x]_P \subseteq X\} \tag{2.4}$$

$$\overline{P}(X) = \{x | [x]_P \cap X \neq \emptyset\} \tag{2.5}$$

Objects in $\underline{P}(X)$ can be with certainty classified as members of X on the basis of knowledge in P, while objects in $\overline{P}(X)$ can be only classified as possible members of X on the basis of knowledge in P.

Let P and Q be equivalence relations over U, then the positive, negative and boundary regions can be defined as:

$$POS_P(Q) = \bigcup_{X \in U/Q} \underline{P}(X) \tag{2.6}$$

$$NEG_P(Q) = U - \bigcup_{X \in U/Q} \overline{P}(X) \tag{2.7}$$

$$BND_P(Q) = \bigcup_{X \in U/Q} \overline{P}(X) - \bigcup_{X \in U/Q} \underline{P}(X) \tag{2.8}$$

The positive region contains all objects of U that can be classified to classes of U/Q using the information in attributes P. The boundary region, $BND_P(Q)$, is the set of objects that can possibly, but not certainly, be classified in this way. The negative region, $NEG_P(Q)$, is the set of objects that cannot be classified to classes of U/Q.

Example 2.3 *Using the same decision table presented in Table 2.2, an illustrative example of the above mentioned calculations is given in what follows where $P = \{b, c\}$ and $Q = \{e\}$:*

$$\begin{aligned}
POS_P(Q) &= \bigcup\{\emptyset, \{x_3, x_6\}, \{x_4\}\} = \{x_3, x_4, x_6\} \\
NEG_P(Q) &= U - \bigcup\{\{x_1, x_5\}, \{x_3, x_1, x_5, x_2, x_7, x_8, x_6\}, \{x_4, x_2, x_7, x_8\}\} = \emptyset \\
BND_P(Q) &= \bigcup\{\{x_1, x_5\}, \{x_3, x_1, x_5, x_2, x_7, x_8, x_6\}, \{x_4, x_2, x_7, x_8\}\} - \{x_3, x_4, x_6\} \\
&= \{x_1, x_2, x_5, x_7, x_8\}
\end{aligned}$$

This means that objects x_3, x_4 and x_6 can certainly be classified as belonging to a class in attribute e, when considering attributes b and c. The rest of the objects cannot be classified as the information that would make them discernible is absent.

2.3.4 Independency of Attributes

An important issue in data analysis is discovering dependencies between attributes. Intuitively, a set of attributes Q depends totally on a set of attributes P, denoted $P \Rightarrow Q$, if all attribute values from Q are uniquely determined by values of attributes from P. If there exists a functional dependency between values of Q and P, then Q depends totally on P. In rough set theory, dependency is defined in the following way: For $P, Q \subset A$, it is said that Q depends on P in a degree $k (0 \leq k \leq 1)$, denoted $P \Rightarrow kQ$, if

$$k = \gamma_P(Q) = \frac{|POS_P(Q)|}{|U|} \tag{2.9}$$

If $k = 1$, Q depends totally on P, if $0 < k < 1$, Q depends partially (in a degree k) on P, and if $k = 0$ then Q does not depend on P.

By calculating the change in dependency when an attribute is removed from the set of considered conditional attributes, a measure of the significance of the attribute can be obtained. The higher the change in dependency, the more significant the attribute is. If the significance is 0, then the attribute is dispensable. More formally, given P, Q and an attribute $a \in P$:

$$\sigma_P(Q, a) = \gamma_P(Q) - \gamma_{P-\{a\}}(Q) \tag{2.10}$$

Example 2.4 *From Table 2.2, the dependency of attribute $\{e\}$ from the attributes $\{b, c\}$ is:*

$$\gamma_{\{b,c\}}(\{e\}) = \frac{|POS_{\{b,c\}}(\{e\})|}{|U|} = \frac{|\{x_3,x_4,x_6\}|}{|\{x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8\}|} = \frac{3}{8}$$

if $P = \{a, b, c\}$ and $Q = e$ then

$$\gamma_{\{a,b,c\}}(\{e\}) = \frac{|\{x_3,x_4,x_6,x_7\}}{8} = \frac{4}{8}\ ; \qquad \gamma_{\{a,b\}}(\{e\}) = \frac{|\{x_3,x_4,x_6,x_7\}}{8} = \frac{4}{8}$$
$$\gamma_{\{b,c\}}(\{e\}) = \frac{|\{x_3,x_4,x_6\}}{8} = \frac{3}{8}\ ; \qquad \gamma_{\{a,c\}}(\{e\}) = \frac{|\{x_3,x_4,x_6,x_7\}}{8} = \frac{4}{8}$$

And calculating the significance of the three attributes gives:

$$\sigma_P(Q, a) = \gamma_{\{a,b,c\}}(\{e\}) - \gamma_{\{b,c\}}(\{e\}) = \frac{1}{8}$$
$$\sigma_P(Q, b) = \gamma_{\{a,b,c\}}(\{e\}) - \gamma_{\{a,c\}}(\{e\}) = 0$$
$$\sigma_P(Q, c) = \gamma_{\{a,b,c\}}(\{e\}) - \gamma_{\{a,b\}}(\{e\}) = 0$$

From this, it follows that attribute a is indispensable, but attributes b and c can be dispensed with when considering the dependency between the decision attribute and the given individual conditional attributes.

2.3.5 Core and Reduct of Attributes

The reduction of attributes is achieved by comparing equivalence relations generated by sets of attributes. Attributes are removed so that the reduced set provides the same predictive capability of the decision feature, D, as the original.

A subset $R \subseteq A$ is a reduct of A with respect to D, if R is independent and

$$\gamma_R(D) = \gamma_A(D) \tag{2.11}$$

Hence, a reduct is a set of attributes from A that preserves dependency and, consequently, set approximation. It means that a reduct is the minimal subset of attributes that enables the same classification of elements of the universe as the whole set of attributes.

The core is the intersection of all reducts. Hence, it is included in every reduct. In a sense, the core is the most important subset of attributes, for none of its elements can be removed without affecting the classification power of attributes.

Example 2.5 *Using the same decision table of Table 2.2, the dependencies for all possible subsets of A can be calculated:*

$\gamma_{\{a,b,c,d\}}(\{e\}) = \frac{8}{8}$; $\gamma_{\{a,b,c\}}(\{e\}) = \frac{4}{8}$; $\gamma_{\{a,b,d\}}(\{e\}) = \frac{8}{8}$; $\gamma_{\{a,c,d\}}(\{e\}) = \frac{8}{8}$;

$\gamma_{\{b,c,d\}}(\{e\}) = \frac{8}{8}$; $\gamma_{\{a,b\}}(\{e\}) = \frac{4}{8}$; $\gamma_{\{a,c\}}(\{e\}) = \frac{4}{8}$; $\gamma_{\{a,d\}}(\{e\}) = \frac{3}{8}$;

$\gamma_{\{b,c\}}(\{e\}) = \frac{3}{8}$; $\gamma_{\{b,d\}}(\{e\}) = \frac{8}{8}$; $\gamma_{\{c,d\}}(\{e\}) = \frac{8}{8}$; $\gamma_{\{a\}}(\{e\}) = \frac{0}{8}$;

$\gamma_{\{b\}}(\{e\}) = \frac{1}{8}$; $\gamma_{\{c\}}(\{e\}) = \frac{0}{8}$; $\gamma_{\{d\}}(\{e\}) = \frac{2}{8}$

Note that the given data set is consistent since $\gamma_{\{a,b,c,d\}}(\{e\}) = 1$. *The minimal reduct set for this example is: Reduct* $= \{\{b, d\}, \{c, d\}\}$

Table 2.3: First Reduct

$x \in U$	b	d	e
x_1	0	2	0
x_2	1	1	2
x_3	0	1	1
x_4	1	2	2
x_5	0	0	1
x_6	2	1	1
x_7	1	1	2
x_8	1	0	1

Table 2.4: Second Reduct

$x \in U$	c	d	e
x_1	2	2	0
x_2	1	1	2
x_3	0	1	1
x_4	0	2	2
x_5	2	0	1
x_6	0	1	1
x_7	1	1	2
x_8	1	0	1

In Table 2.2, there are two possible reducts with respect to the decision attribute $\{e\}$; $\{b, d\}$ *and* $\{c, d\}$. *These reducts are independent with respect to the decision attribute* $\{e\}$ *and have the same dependency as the whole subset of condition attributes A. That means that either the attribute b or c can be eliminated from the table and*

consequently instead of Table 2.2, we can use either Table 2.3 or Table 2.4. The core is the attribute d. It is the intersection of the two possible reducts.

2.3.6 Rough Sets and the Use of Heuristics

The problem of finding a reduct has been the subject of much research (Swiniarski & Skowron, 2003). The most basic solution to locating such a subset is to simply generate all possible subsets and retrieve those with a maximum rough set dependency degree. However, this is an expensive solution to the problem and is only practical for very simple data sets. Most of the time, only one reduct is required as, typically, only one subset of features is used to reduce a data set, so all the calculations involved in discovering the rest are pointless. Another shortcoming of finding all possible reducts using rough sets is to inquire about which is the best reduct for the classification process. The solution to these issues is to apply a *"heuristic attribute selection"* method (Zhong, Dong, & Ohsuga, 2001).

Among the most interesting heuristic methods proposed in literature, we mention the *"QuickReduct"* algorithm (Chouchoulas & Shen, 2001). QuickReduct attempts to calculate a reduct without exhaustively generating all possible subsets. It starts off with an empty set and adds in turn, one at a time, those attributes that result in the greatest increase in the rough set dependency metric. According to the QuickReduct algorithm, the dependency of each attribute is calculated and the best candidate is chosen. This process continues until the dependency of the reduct equals the consistency of the data set (Jensen, 2005). Other approaches generating reducts from information systems have been developed and can be found in (Slezak, 1996).

2.3.7 Rough Sets and the Link to Other Theories

Rough set theory was combined with other theories such as the *probability theory*, *fuzzy set theory* (Zadeh, 1965) and the *belief function theory* (Shafer, 1976). As each of these theories has inherent characteristics, works tended to explore the effectiveness of the mentioned theories to handle specific kinds of problems. For instance, both rough sets and the probability theory were hybridized in order to tackle problems dealing with incomplete data, and to solve the same kind of problems rough set theory was combined with the belief function theory. Another combination of rough sets was its hybridization with fuzzy sets which gave rise to the Fuzzy-Rough Set Theory (FRST). The latter was, also, dedicated to handle incomplete data sets. Besides, it is applied to estimate the missing values in the learning process. FRST was first of all proposed as an extension to the rough set theory as the latter has some limitations. This will be discussed in the next Section.

2.4 Fuzzy-Rough Feature Selection

In this Section, the basics of FRST are described while focusing on the feature selection process performed by this theory.

2.4.1 From Rough Sets to Fuzzy-Rough Sets

In most databases, values of attributes may be both crisp and real-valued, and this is where many feature selectors including RST encounter a problem. More precisely, it is not possible to decide whether two attribute values are similar and to what extent they are the same. For instance, in RST, the values -0.1 and -0.11 are as different as -0.1 and 300. One solution to this problem is to discretize the data set beforehand producing a new database with crisp values. However, this is often still inadequate since it presents a kind of information loss (Jensen, 2005).

To deal with the issue stated above, it is clearly necessary to call for methods providing the means of feature selection for both crisp and real-value attributed data sets. One theory which is capable to handle such type of data is fuzzy set theory (Zadeh, 1965). An introduction to fuzzy set theory can be found in Appendix A. Fuzzy sets and the process of fuzzification provide a mechanism by which real-valued features can be effectively managed. This is achieved by allowing values to belong to more than one label with various degrees of membership. Consequently, the vagueness presented in data can be modeled. And therefore, the hybridization of RST and fuzzy set theory, giving rise to the fuzzy-rough set theory, is seen as a promising combination to be applied for the feature selection task. In fact, FRST is an extension of rough set theory allowing all memberships to take values in the range $[0, 1]$. This permits a higher degree of flexibility compared to the strict requirements of rough sets that only deal with full or zero set memberships (Jensen, 2005).

2.4.2 Fuzzy Equivalence Classes

In the same way that crisp equivalence classes are central to rough sets, fuzzy equivalence classes are central to the fuzzy-rough set approach. For typical rough set attribute reduction applications, this means that the decision values and the conditional values may all be fuzzy. The concept of crisp equivalence classes can be extended by the inclusion of a fuzzy similarity relation S on the universe, which determines the extent to which two elements are similar in S. For example, if $\mu_S(x, y) = 0.9$, then objects x and y are considered to be quite similar. The usual properties of reflexivity ($\mu_S(x, x) = 1$), symmetry ($\mu_S(x, y) = \mu_S(y, x)$) and transitivity ($\mu_S(x, z) \geq \mu(x, y) \wedge \mu_S(y, z)$) hold (Jensen, 2005). Using the fuzzy similarity relation, the fuzzy equivalence class $[x]_S$ for objects close to x can be defined:

$$\mu_{[x]_S}(y) = \mu_S(x, y) \tag{2.12}$$

This definition degenerates to the normal definition of equivalence classes when S is non-fuzzy. The following axioms should hold for a fuzzy equivalence class F:

1. $\exists x, \mu_F(x) = 1$ (μ_F is normalized)
2. $\mu_F(x) \wedge \mu_S(x, y) \leq \mu_F(y)$ class of y
3. $\mu_F(x) \wedge \mu_F(y) \leq \mu_S(x, y)$

The first axiom corresponds to the requirement that an equivalence class is nonempty. The second axiom states that elements in y's neighborhood are in the equivalence class of y. The final axiom states that any two elements in F are related via S. The family of normal fuzzy sets produced by a fuzzy partitioning of the universe of discourse can play the role of fuzzy equivalence classes.

Example 2.6 *Consider the crisp partitioning of a universe of discourse, U, by the attributes in* $Q : U/Q = \{\{x_1, x_3, x_6\}, \{x_2, x_4, x_5\}\}$. *This contains two equivalence classes (*$\{x_1, x_3, x_6\}$ *and* $\{x_2, x_4, x_5\}$*) that can be thought of as degenerated fuzzy sets, with those elements belonging to the class possessing a membership of one, zero otherwise. For the first class, for instance, the objects* x_2, x_4 *and* x_5 *have a membership of zero. Extending this to the case of fuzzy equivalence classes is straightforward: objects can be allowed to assume membership values, with respect to any given class, in the interval* $[0, 1]$.

2.4.3 Fuzzy-Rough Sets Basic Concepts

Same as RST, FRST is based on the fuzzy P-lower and fuzzy P-upper approximations which are defined as:

$$\mu_{\underline{P}X}(F_i) = inf_x max\{1 - \mu_{F_i}(x), \mu_X(x)\}\forall i \tag{2.13}$$

$$\mu_{\overline{P}X}(F_i) = sup_x min\{\mu_{F_i}(x), \mu_X(x)\}\forall i \tag{2.14}$$

where F_i denotes a fuzzy equivalence class belonging to U/P, and X is the fuzzy concept to be approximated. The tuple $< \underline{P}X, \overline{P}X >$ is called a *fuzzy-rough set*. These definitions diverge a little from the crisp upper and lower approximations, as the memberships of individual objects to the approximations are not explicitly available. As a result of this, the fuzzy lower and upper approximations are herein redefined as:

$$\mu_{\underline{P}X} = sup_{F\in U/P} min(\{\mu_F(x), inf_{y\in U} max\{1 - \mu_F(y), \mu_X(y)\}) \tag{2.15}$$

$$\mu_{\overline{P}X} = sup_{F\in U/P} min(\{\mu_F(x), sup_{y\in U} min\{\mu_F(y), \mu_X(y)\}) \tag{2.16}$$

2.4.4 Fuzzy-Rough Reduction Process

The fuzzy-rough set based feature selection approach was built on the notion of fuzzy lower approximation to enable reduction of data sets containing real-valued features. As will be shown, the process becomes identical to RST when dealing with nominal well-defined features. The crisp positive region in traditional rough set theory is defined as the union of the lower approximations. By the extension principle, the membership of an object $x \in U$, belonging to the fuzzy positive region can be defined by:

$$\mu_{POS_{P(Q)}}(x) = sup_{X\in U/Q} \mu_{\underline{P}X}(x) \tag{2.17}$$

Object x will not belong to the positive region only if the equivalence class it belongs to is not a constituent of the positive region. This is equivalent to RST where objects belong to the positive region only if their underlying equivalence class does so. Similarly, the negative and boundary regions can be defined. Using the definition of the fuzzy positive region, the new dependency function can be defined as follows:

$$\gamma'_P(Q) = \frac{|\mu_{POS_{P(Q)}}(x)|}{|U|} = \frac{|\sum_{x\in U} \mu_{POS_{P(Q)}}(x)|}{|U|} \tag{2.18}$$

As with rough sets, the dependency of Q on P is the proportion of objects that are discernible out of the entire data set. In the present approach, this corresponds to determining the fuzzy cardinality of $\mu_{POS_{P(Q)}}(x)$ divided by the total number of objects in the universe. The definition of dependency degree covers the crisp case as its specific instance.

2.4.5 Fuzzy-Rough QuickReduct

A problem may arise when this approach is compared to RST. In conventional rough set attribute reduction, a reduct is defined as a subset R of the features which have the same information content as the full feature set A. In terms of the dependency function this means that the values $\gamma(R)$ and $\gamma(A)$ are identical and equal to 1 if the data set is consistent. However, in the fuzzy-rough approach this is not necessarily the case as the uncertainty encountered when objects belong to many fuzzy equivalence classes results in a reduced total dependency. With these issues in mind, a fuzzy-rough QuickReduct algorithm, described by Algorithm 2.1, was developed in (Jensen, 2005).

Algorithm 2.1 The Fuzzy-Rough QuickReduct Algorithm

```
C: the set of all conditional features;
D: the set of decision features;
R ← { } ; γ'_best = 0 ; γ'_prev = 0
do
        T ← R
        γ'_prev = γ'_best
        ∀x ∈ (C − R)
            if γ'_{R∪{x}}(D) > γ'_T(D)
                T ← R ∪ {x}
                γ'_best = γ'_T(D)
        R ← T
until γ'_best = γ'_prev
return R
```

The fuzzy-rough QuickReduct algorithm employs the fuzzy-rough dependency function γ' to choose those attributes to add to the current reduct candidate. The algorithm terminates when the addition of any remaining attribute does not increase the dependency.

Example 2.6 *To illustrate the operation of the fuzzy-rough QuickReduct algorithm, an example database is given in Table 2.5. This Table contains 3 real-valued conditional attributes and 1 crisp-valued decision attribute.*

Table 2.5: Example Database

Object	a	b	c	q
x_1	-0.4	-0.3	-0.5	No
x_2	-0.4	0.2	-0.1	Yes
x_3	-0.3	-0.4	-0.3	No
x_4	0.3	-0.3	0	Yes
x_5	0.2	-0.3	0	Yes
x_6	0.2	0	0	No

Let us remind that in rough set feature selection, the data set would be discretized using the non-fuzzy sets. However, in the fuzzy-rough approach fuzzy sets, defined in Figure 2.1, are used in calculating the fuzzy lower ap-

proximations and fuzzy positive regions. This will be illustrated in the following example.

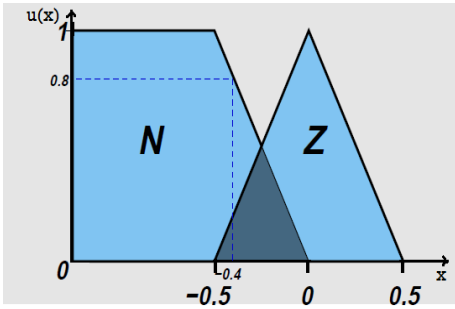

Figure 2.1: Fuzzification for Conditional Features

To start with, and by using the fuzzy-rough QuickReduct algorithm, the potential reduct is initialized to the empty set.

Using fuzzy sets which are defined in Figure 2.1 and setting $A = \{a\}$, $B = \{b\}$, $C = \{c\}$ and $Q = \{q\}$, the following equivalence classes are obtained:

$$U/A = \{N_a, Z_a\}$$
$$U/B = \{N_b, Z_b\}$$
$$U/C = \{N_c, Z_c\}$$
$$U/Q = \{\{x_1, x_3, x_6\}, \{x_2, x_4, x_5\}\}$$

The first step is to calculate the lower approximations of the sets A, B and C, using Equation 2.15. To clarify the calculations involved, Table 2.6 contains the membership degrees of objects to fuzzy equivalence classes. For simplicity, only A will be considered here; i.e, using A to approximate Q.

Table 2.6: Membership Values of Objects to Corresponding Fuzzy Sets

Object	a		b		c		q	
	N_a	Z_a	N_b	Z_b	N_c	Z_c	$\{x_1, x_3, x_6\}$	$\{x_2, x_4, x_5\}$
x_1	0.8	0.2	0.6	0.4	1.0	0.0	1.0	0.0
x_2	0.8	0.2	0.0	0.6	0.2	0.8	0.0	1.0
x_3	0.6	0.4	0.8	0.2	0.6	0.4	1.0	0.0
x_4	0.0	0.4	0.6	0.4	0.0	1.0	0.0	1.0
x_5	0.0	0.6	0.6	0.4	0.0	1.0	0.0	1.0
x_6	0.0	0.6	0.0	1.0	0.0	1.0	1.0	0.0

For the first decision equivalence class $X = \{x_1, x_3, x_6\}$, $\mu_{\underline{A}\{x_1,x_3,x_6\}}(x)$ needs to be calculated:

$$\mu_{\underline{A}\{x_1,x_3,x_6\}}(x) = sup_{F\in U/A} min(\{\mu_F(x), inf_{y\in U} max\{1-\mu_F(y), \mu_{\{x_1,x_3,x_6\}}(y)\}\})$$

Considering the first fuzzy equivalence class of A, N_a:

$$min(\mu_{N_a}(x), inf_{y\in U} max\{1-\mu_{N_a}(y), \mu_{\{x_1,x_3,x_6\}}(y)\})$$

For object x_2 this can be calculated as follows: From Table 2.6, it can be seen that the membership of object x_2 to the fuzzy equivalence class N_a, $\mu_{N_a}(x_2)$, is 0.8. The remainder of the calculation involves finding the smallest of the following values:

$$\begin{aligned}
max(1-\mu_{N_a}(x_1), \mu_{\{x_1,x_3,x_6\}}(x_1)) &= max(0.2, 1.0) = 1.0\\
max(1-\mu_{N_a}(x_2), \mu_{\{x_1,x_3,x_6\}}(x_2)) &= max(0.2, 0.0) = 0.2\\
max(1-\mu_{N_a}(x_3), \mu_{\{x_1,x_3,x_6\}}(x_3)) &= max(0.4, 1.0) = 1.0\\
max(1-\mu_{N_a}(x_4), \mu_{\{x_1,x_3,x_6\}}(x_4)) &= max(1.0, 0.0) = 1.0\\
max(1-\mu_{N_a}(x_5), \mu_{\{x_1,x_3,x_6\}}(x_5)) &= max(1.0, 0.0) = 1.0\\
max(1-\mu_{N_a}(x_6), \mu_{\{x_1,x_3,x_6\}}(x_6)) &= max(1.0, 1.0) = 1.0
\end{aligned}$$

From the calculations above, the smallest value is 0.2, hence:

$$min(\mu_{N_a}(x), inf_{y\in U} max\{1-\mu_{N_a}(y), \mu_{\{x_1,x_3,x_6\}}(y)\}) = min(0.8, inf\{1, 0.2, 1, 1, 1, 1\}) = 0.2$$

Similarly for Z_a:

$$min(\mu_{Z_a}(x), inf_{y\in U} max\{1-\mu_{Z_a}(y), \mu_{\{x_1,x_3,x_6\}}(y)\}) = min(0.2, inf\{1, 0.8, 1, 0.6, 0.4, 1\}) = 0.2$$

Thus,

$$\mu_{\underline{A}\{x_1,x_3,x_6\}}(x_2) = 0.2$$

Calculating the A-lower approximation of $X = \{x_1, x_3, x_6\}$ for every object gives:

$$\begin{aligned}
&\mu_{\underline{A}\{x_1,x_3,x_6\}}(x_1) = 0.2 \ ; \ \mu_{\underline{A}\{x_1,x_3,x_6\}}(x_2) = 0.2\\
&\mu_{\underline{A}\{x_1,x_3,x_6\}}(x_3) = 0.4 \ ; \ \mu_{\underline{A}\{x_1,x_3,x_6\}}(x_4) = 0.4\\
&\mu_{\underline{A}\{x_1,x_3,x_6\}}(x_5) = 0.4 \ ; \ \mu_{\underline{A}\{x_1,x_3,x_6\}}(x_6) = 0.4
\end{aligned}$$

The corresponding values for $X = \{x_2, x_4, x_5\}$ can, also, be determined:

$$\begin{aligned}
&\mu_{\underline{A}\{x_2,x_4,x_5\}}(x_1) = 0.2 \ ; \ \mu_{\underline{A}\{x_2,x_4,x_5\}}(x_2) = 0.2\\
&\mu_{\underline{A}\{x_2,x_4,x_5\}}(x_3) = 0.4 \ ; \ \mu_{\underline{A}\{x_2,x_4,x_5\}}(x_4) = 0.4\\
&\mu_{\underline{A}\{x_2,x_4,x_5\}}(x_5) = 0.4 \ ; \ \mu_{\underline{A}\{x_2,x_4,x_5\}}(x_6) = 0.4
\end{aligned}$$

It is a coincidence here that $\mu_{\underline{A}\{x_2,x_4,x_5\}}(x) = \mu_{\underline{A}\{x_1,x_3,x_6\}}(x)$ *for this example. Using these values, the fuzzy positive region for each object can be calculated using Equation 2.17:*

$$\mu_{POS_A(Q)}(x) = sup_{X \in U/Q} \mu_{\underline{A}X}(x)$$

This results in:

$$\mu_{POS_{A(Q)}}(x_1) = 0.2 \; ; \; \mu_{POS_{A(Q)}}(x_2) = 0.2$$
$$\mu_{POS_{A(Q)}}(x_3) = 0.4 \; ; \; \mu_{POS_{A(Q)}}(x_4) = 0.4$$
$$\mu_{POS_{A(Q)}}(x_5) = 0.4 \; ; \; \mu_{POS_{A(Q)}}(x_6) = 0.4$$

The next step is to determine the degree of dependency of Q on A using Equation 2.18:

$$\gamma'_A(Q) = \frac{|\sum_{x \in U} \mu_{POS_A(Q)}(x)|}{|U|} = \frac{2}{6}$$

Calculating for B and C gives:

$$\gamma'_B(Q) = \frac{2.4}{6} \; ; \; \gamma'_C(Q) = \frac{1.6}{6}$$

From this, it can be seen that attribute b will cause the greatest increase in dependency degree. This attribute is chosen and added to the potential reduct. The process iterates and the two dependency degrees calculated are:

$$\gamma'_{\{a,b\}}(Q) = \frac{3.4}{6} \; ; \; \gamma'_{\{b,c\}}(Q) = \frac{3.2}{6}$$

Adding attribute a to the reduct candidate causes the larger increase of dependency, so the new candidate becomes $\{a, b\}$*. Lastly, attribute c is added to the potential reduct:*

$$\gamma'_{\{a,b,c\}}(Q) = \frac{3.4}{6}$$

As this causes no increase in dependency, the algorithm stops and outputs the reduct $\{a, b\}$*. The data set can now be reduced to only those attributes appearing in the reduct. When rough set attribute reduction is performed on this data set (after using the same fuzzy sets to discretize the real-valued attributes), the reduct generated is* $\{a, b, c\}$*; i.e. the full conditional attribute set (Jensen & Shen, 2002). Unlike rough set attribute reduction, the true minimal reduct was found using the information on degrees of membership. It is clear from this example that the information lost by using rough set attribute reduction can be important when trying to discover the smallest reduct from a data set.*

2.5 Conclusion

In this Chapter, both rough sets and fuzzy-rough sets for feature selection were introduced. The application of these theories form the main contribution of this Thesis. New hybridized models will be presented in the second part of this dissertation where the mentioned theories are combined with the DCA. This is to cover the DCA limitations which are linked to its data pre-processing step.

Part II

Contributions

Part II presents the contributions of this Thesis. This part focuses on presenting our newly developed approaches solving some of the DCA mentioned limitations. The two first Chapters present new data pre-processing modules dedicated to the standard version of the dendritic cell algorithm. These developed modules are based on the theory of rough sets and the theory of fuzzy-rough sets. Secondly, the next two Chapters detail solutions to overcome the DCA limitation as it is sensitive to the input class data order. The proposed algorithms are based on the use of fuzzy set theory, fuzzy clustering techniques and a database maintenance technique. Finally, the last Chapter deals with the development of a more general DCA taking into account all the previously studied parts.

Chapter 3

A Rough DCA Pre-Processing Module

3.1 Introduction

In this Chapter, we aim to investigate the DCA data pre-processing phase including the step of feature selection/reduction and the step of signal categorization. We aim to detect and to handle the shortcomings of these two steps. To do so, we suggest to apply Rough Set Theory (RST) as a feature selection and signal categorization technique in the DCA data pre-processing phase. This will result on different DCA new automated approaches that will be compared to each other while pointing out their characteristics.

This Chapter is organized as follows: Firstly, criticisms of the standard DCA data pre-processing module are given in Section 3.2. Secondly, the development of the proposed rough pre-processing modules is based on a set of hypotheses. These hypotheses are presented in Section 3.3. Then, the proposed rough DCA hybrid approaches are explained in details in Sections 3.4, 3.5 and 3.6.

3.2 Analysis of the Standard DCA Data Pre-processing Module

Focusing on the DCA data pre-processing phase, the algorithm applies the Principal Component Analysis technique (PCA). PCA is applied to achieve the task of feature extraction and the task of signal categorization. Criticisms of both steps are presented in what follows.

3.2.1 Criticisms of the Feature Extraction Step

PCA, for the task of feature extraction, transforms a finite number of possibly correlated vectors into a smaller number of uncorrelated vectors termed “principal components”. These extracted principal components reveal the

internal structure of the given data with the focus on data variance.

The fact of applying PCA for the DCA feature reduction step suffers from some issues. Firstly, applying PCA does not guarantee that the first selected principal components that capture most of the variance will be the most adequate features to retain (Cantú-Paz, 2004). Furthermore, applying PCA destroys the underling meaning behind the features present in the used data set (the semantics). This contradicts the specificity of the DCA as it is important to know the source (feature) of each signal category. Adding to these issues, an informed guess has to be made as to how many variables should be kept for the PCA data reduction process. This may introduce a potential source of error (Cantú-Paz, 2004).

3.2.2 Criticisms of the Signal Categorization Step

Once features are extracted, DCA moves to its second data pre-processing substep which is signal categorization. For signal categorization, the DCA uses the PCA calculated standard deviations and selects the highest values. As both PAMPs and SSs are positive indicators of an anomalous and normal signal, one attribute is used to form both of these two signals. Thus, the attribute having the lowest standard deviation out of the selected attribute set is used to form both PAMPs and SSs. Using one attribute for these two signals requires a threshold level to be set: values greater than this can be classed as SSs otherwise as PAMPs. As for the DSs attribute assignment and since the DSs are "less than certain to be anomalous", the combination of the rest of the selected attributes is chosen to represent it.

As explained, the categorization process is based on the PCA attributes ranking in terms of variability. However, this categorization process could not be considered as a coherent and a reliable categorization procedure. Thus, the standard DCA pre-processing phase should be achieved with the use of the right data pre-processing technique. In this Chapter, we propose to develop novel DCAs based on the rough set theory concepts. Our new rough DCAs will be based on new feature selection and signal categorization processes.

3.3 Rough DCA Based Methods

To handle the mentioned drawbacks of the standard DCA data pre-processing module, we propose three automated DCAs based on the theory of rough sets (RST). The difference between the three proposed rough DCAs is based on a set of hypotheses that will be tested later in the experimental part of this Chapter. The drawn hypotheses are constructed as follows:

- **H1:** The categorization of different features to different signals leads to improve the performance of the dendritic cell algorithm. We will show that assigning for each selected feature a specific signal category leads to a better performance than assigning the same attribute to both SSs and PAMPs.
- **H2:** The use of a rough set heuristic can keep the DCA main characteristic which is its lightweight in terms of processing time. This will be achieved by finding a trade-off between generating satisfactory classification results and preserving the lightweight of the algorithm.

H1 focuses on checking whether designating the same attribute to both SSs and PAMPs is more convenient for the DCA signal categorization substep than assigning different features for each signal category. By designating the

same attribute to both SSs and PAMPs, we develop our first DCA model based on Rough Set Theory (RST) named RST-DCA. By attaching different attributes to different signals, we develop our second rough DCA model based on the "Reduct" and the "Core" RST concepts. The proposed algorithm is named RC-DCA. Conversely, H2 deals with the development of our third rough DCA based on the use of a rough heuristic; i.e., the QuickReduct algorithm. The proposed rough DCA is named QR-DCA. By developing the latter algorithm, we try to find a trade-off between generating satisfactory classification results and preserving the lightweight of the standard DCA.

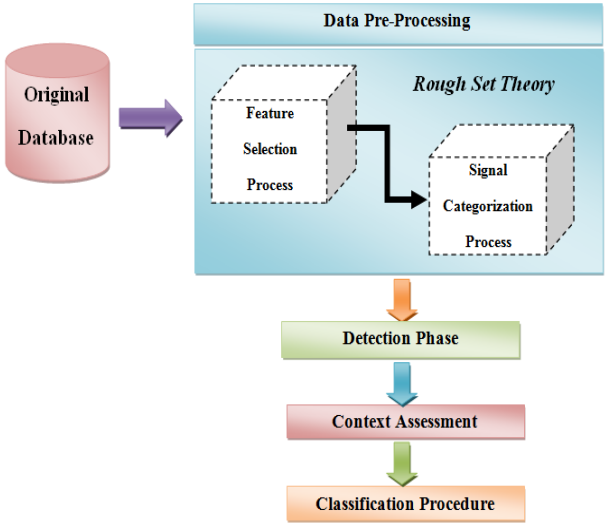

Figure 3.1: The Rough DCA Proposed Model

The three developed rough DCAs, including RST-DCA, RC-DCA and QR-DCA, function under four levels of abstraction as shown in Figure 3.1. We will mainly discuss the algorithms data pre-processing phase as the rest of the algorithms steps are performed the same as the standard DCA and as explained, previously, in Chapter 1.

3.4 RST-DCA

RST-DCA is based on assigning the same selected attribute to both SSs and PAMP signals. The data pre-processing step of our RST-DCA rough algorithm is divided into two substeps which are the feature selection and the signal categorization processes that will be dealt with in what follows.

3.4.1 RST-DCA Feature Selection Process

For antigen classification, our learning problem has to select high discriminating features from the original input database which corresponds to the antigen information data set. We may formalize this problem as an information table where universe $U = \{x_1, x_2, \ldots, x_N\}$ is a set of antigen identifiers, the conditional attribute set $C = \{c_1, c_2, \ldots, c_N\}$ contains each feature of the information table to select and the decision attribute D of our learning problem corresponds to the class label of each sample. As DCA is applied to binary classification problems, the input database

has a single binary decision attribute. Hence, the decision attribute D has binary values d: either the antigen is collected under safe circumstances reflecting a normal behavior or the antigen is classified as anomalous. The condition attribute feature D is defined as follows:

$$D = \{normal, anomalous\}$$

For feature selection and based on the RST computations seen previously in Chapter 2, RST-DCA computes the positive region for the whole attribute set C for both label classes of D; defined as $POS_C(\{d\})$. Then, RST-DCA computes the positive region of each feature c, defined as $POS_c(\{d\})$, and the positive region of all the composed features $C - \{c\}$; defined as $POS_{C-\{c\}}(\{d\})$. $POS_{C-\{c\}}(\{d\})$ corresponds to the positive region of each attribute when discarding each time one feature c from C. This process continues until finding the minimal subset of attributes R from C that preserves the positive region as the whole attribute set C does. In each computation level, RST-DCA removes the unnecessary features that may affect negatively its classification accuracy.

The result of these computations is either one reduct $R = RED_D(C)$ or a family of reducts $RED_D^F(C)$. Any reduct of $RED_D^F(C)$ can be used to replace the original antigen information table. Consequently, if the RST-DCA generates only one reduct then for the feature selection process, RST-DCA chooses this specific R which represents the most informative features that preserve nearly the same classification power of the original data set. If the RST-DCA generates a family of reducts then RST-DCA chooses randomly one reduct R among $RED_D^F(C)$ to represent the original input antigen information table. This random choice is argued by the same priority of all the reducts in $RED_D^F(C)$. In other words, any reduct R of the reducts $RED_D^F(C)$ can be used to replace the original information table.

These attributes which constitute the reduct will describe all concepts in the original training data set. Using the reduct concept, our method can guarantee that attributes of extracted feature patterns will be the most relevant for its classification task. Once the reduct is generated, our proposed solution moves to its second pre-processing substep which is signal categorization.

3.4.2 RST-DCA Signal Categorization Process

Throughout this substep, RST-DCA has to assign for each selected attribute included in the generated reduct set its definite and specific signal category. The general guidelines for signal assignment are presented in the list below:

- **Safe signals:** The presence of safe signals certainly indicates that no anomalies are present.
- **PAMPs:** The presence of PAMPs usually means that there is an anomalous situation.
- **Danger signals:** The presence of danger signals may or may not show an anomalous situation, however the probability of an anomaly is higher than under normal circumstances.

From the previous definitions, both PAMPs and SSs are positive indicators of an anomalous and normal signal. DSs are measuring situations where the risk of anomalousness is high but there is no signature of a specific cause. In other words, PAMPs and SSs have a certain final context (either an anomalous or a normal behavior) while the DSs cannot specify exactly the final context to assign to the collected antigen. This is because the information returned by the DSs is not certain as the collected antigen may or may not indicate an anomalous situation. This problem can

be formulated as follows:

Both PAMPs and SSs are more informative than DSs which means that both of these signals can be seen as indispensable attributes. To represent this level of importance, RST-DCA uses the "core" RST concept. On the other hand, DSs are less informative than PAMPs and SSs. Therefore, RST-DCA applies the rest of the reduct attributes, discarding the attribute chosen to represent both the SSs and the PAMPs, to represent the DSs.

As stated in the previous substep, our method may either produce only one reduct R or a family of reducts $RED_D^F(C)$. The process of signal categorization for both cases are described in what follow.

The Process of One Reduct

In case where RST-DCA generates only one reduct; it means that $CORE_D(C) = RED_D(C)$. In other words, all the features of the reduct are indispensable. In this case, RST-DCA selects randomly one attribute c from $CORE_D(C)$ and assigns it to both PAMPs and SSs as they are the most informative signals. Using one attribute for these two signals requires a threshold level to be set: values greater than this can be classed as SSs, otherwise as a PAMPs. The rest of the attributes $CORE_D(C) - \{c\}$ are combined and the resulting value is assigned to the DSs as it is less than certain to be anomalous.

The Process of a Family of Reducts

In case where RST-DCA generates a family of reducts $RED_D^F(C)$, both RST concepts appear: the core $CORE_D(C)$ and the reducts $RED_D^F(C)$. Let us remind that $CORE_D(C) = \bigcap RED_D(C)$. This means that on the one hand, we have the minimal set of attributes preserving the positive region of the whole set, the reducts sets, and on the other hand we have the set of attributes that are contained in all reducts, the core set, which cannot be removed without changing the positive region. This means that all the attributes present in the core are indispensable.

For signal categorization, PAMPs and SSs are assigned randomly one attribute c among the features in $CORE_D(C)$. As for the DSs assignment, first, RST-DCA chooses randomly a reduct $RED_D(C)$ among $RED_D^F(C)$. Then, RST-DCA combines all the $RED_D(C)$ features except that c attribute already chosen and assigns the resulting value to the DSs.

As seen in this Section, RST-DCA applies the concepts of rough set theory to select first of all the most interesting features from the input database; then assigns each selected attribute to its specific signal category.

Example 3.1 *To illustrate an example of the RST-DCA functioning, a decision system, Table 3.1, is used. It consists of 4 conditional features (a, b, c, d), 1 decision feature (e) and 8 objects. By using the RST concepts explained previously in Chapter 2, RST-DCA generates first of all the core and the reduct sets. The computations on how to generate these sets are already described in Section 2.3.5.*

Table 3.1: The RST-DCA Decision System

$x \in U$	a	b	c	d	e
x_1	1	0	2	2	1
x_2	0	1	1	1	2
x_3	2	0	0	1	1
x_4	1	1	0	2	2
x_5	1	0	2	0	1
x_6	2	2	0	1	1
x_7	2	1	1	1	2
x_8	0	1	1	0	1

The RST-DCA feature selection phase consists in selecting the most informative features from the decision table which is, also, called the antigen data set. By applying the needed computations, RST-DCA generates the following reducts:

- $R = \{\{b, d\}, \{c, d\}\}$

Once the reduct set is generated, RST-DCA moves to the signal categorization step. Since, RST-DCA has generated a family of reducts then RST-DCA has to specify the core concept:

- $core = \{b, d\} \cap \{c, d\} = \{d\}$

As RST-DCA uses one feature to represent both SS and PAMP signals, then it selects randomly one attribute from the core; i..e, the feature $\{d\}$. Since one attribute is used to represent the two mentioned signals, a threshold should be applied to categorize each antigen either as a PAMP signal or as a safe signal. The threshold to be used is the median value of the $\{d\}$ attribute. Once the right feature is assigned to represent both SSs and PAMPs, RST-DCA has to select randomly one reduct among the family of the generated reducts. This is needed for the DSs categorization process. In this case, RST-DCA selects the following reduct:

- $R = \{c, d\}$

RST-DCA discards the $\{d\}$ attribute, first, since it is already chosen to represent the SSs and the PAMPs. Then, the algorithm uses the rest of the reduct attributes to represent the DSs. In this case, RST-DCA selects $\{c\}$ which will be seen as the DS. This is summarized in the following list:

- $SS = PAMP = \{d\}$
- $DS = \{c\}$

3.4.3 Experimental Setup

To test the validity of our RST-DCA, our experiments are performed using binary databases from (Asuncion & Newman, 2007). The data sets are described in Table 3.2. We will compare the RST-DCA results to the results obtained from the standard DCA version which is based on the application of PCA (PCA-DCA).

Table 3.2: Description of Databases

Database	Ref	♯ Instances	♯ Attributes
Sonor	SN	208	61
Molecular-Bio	Bio	106	59
Cylinder Bands	CylB	540	40
Chess	Ch	3196	37
Ionosphere	IONO	351	35
Sick	Sck	3772	30
Horse Colic	HC	368	23
German-Credit	GC	1000	21
Labor Relations	LR	57	16
Red-White-Win	RWW	6497	13

For the PCA-DCA data pre-processing phase, the algorithm calculates the standard deviation of each attribute. The attribute having the lowest standard deviation is used to derive the PAMP and safe signals. The rest of the attribute set is used to calculate the DS values. For our RST-DCA data pre-processing phase, the algorithm uses RST as described in the previous Section. For both algorithms, each data item is mapped as an antigen with the value of the antigen equals to the data ID of the item. In spite of incorporating inflammation signals into the model, they are not used in our experiments as no obvious mapping is available. All featured parameters are derived from empirical immunological data.

In all experiments, a population of 100 cells is used and 10 DCs sample the antigen vector each cycle. The migration threshold of an individual DC is set to 10 to ensure this DC to survive over multiple iterations. To perform anomaly detection, a threshold which is automatically generated from the data is applied to the MCAVs. The MCAV threshold is derived from the proportion of anomalous data instances of the whole data set. Items below the threshold are classified as class one and above as class two. The resulting classified antigens are compared to the labels given in the original data sets. For each experiment, the results presented are based on mean MCAVs generated across 10 runs.

We evaluate the performance of the algorithms in terms of number of extracted features, sensitivity, specificity and accuracy. The sensitivity, specificity and accuracy criteria are defined as:

$$Sensitivity = \frac{TP}{TP + FN} \tag{3.1}$$

$$Specificity = \frac{TN}{TN + FP} \tag{3.2}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{3.3}$$

where TP, FP, TN, and FN refer, respectively, to true positive, false positive, true negative and false negative. These criteria may vary from 0% to 100%. The higher the values, the better the classification of the algorithm is. The true positive indicates the correctly identified items, the false positive indicates the incorrectly identified items, the true negative indicates the correctly rejected and the false negative indicates the incorrectly rejected items.

3.4.4 Results and Analysis

In this Section, we show that using RST instead of PCA is much convenient for the DCA data pre-processing phase as it improves the algorithm's classification performance. This is confirmed by the results given in Table 3.3.

Table 3.3: PCA-DCA and RST-DCA Comparison Results

Database	♯ Attributes		Sensitivity (%)		Specificity (%)		Accuracy (%)	
	DCA		DCA		DCA		DCA	
	PCA	RST	PCA	RST	PCA	RST	PCA	RST
SN	28	20	81.08	88.28	74.22	82.47	77.88	85.57
Bio	24	19	43.39	52.83	39.62	52.83	41.50	52.83
CylB	16	7	91.50	96.50	92.94	96.79	92.38	96.67
Ch	14	11	94.06	97.84	93.64	98.23	93.86	98.02
IONO	24	19	93.65	95.23	94.22	96.88	94.58	96.29
Sck	22	20	90.90	96.53	94.32	96.83	94.11	96.81
HC	19	14	88.88	87.50	76.97	91.44	83.96	89.13
GC	17	17	80.29	83.21	89.53	90.22	87.00	88.30
LR	10	5	85.00	90.00	83.78	86.48	84.21	87.71
RWW	10	6	97.95	98.01	97.62	98.24	97.87	98.07

From Table 3.3, it is clearly seen that the number of features selected by our RST-DCA is less than the one generated by PCA-DCA. This can be explained by the appropriate use of RST for feature selection. In fact, RST-DCA keeps only the most informative features which constitute the reduct. For instance, by applying our RST-DCA method to the CylB data set, the number of selected features is only 7 attributes. However, when applying the PCA-DCA to the same database (CylB), the number of the retained features is 16. We can notice that PCA preserves additional features which is the result of the PCA overestimation of the number of factors to retain. This overestimation affects the DCA classification task by producing lessened results. On the other hand, RST-DCA based on the reduct concept selects the minimal set of features from the original database. In fact, by reducing more the number of features while preserving the classification power of the original data set, our RST-DCA has the advantages of decreasing the cost of acquiring data and making the classification model easier to understand unlike when applying the PCA. In addition, RST-DCA has sufficient advantages over the PCA-DCA as it does not require any additional information about data a priori such as thresholds or expert knowledge on a particular domain. Thus, RST-DCA results will not be influenced by any external information.

As for the classification accuracy, from Table 3.3, we can easily remark that the RST-DCA accuracy is notably better than the one given by the PCA-DCA. For example, when applying the RST-DCA to the CylB database the RST-DCA accuracy is set to 96.67%. Nevertheless, when applying the PCA-DCA to the same database the accuracy is set to 92.38%. Same remark is noticed for both the sensitivity and the specificity criteria.

These encouraging RST-DCA results are explained by the appropriate use of the reduct RST fundamental concept to select only the essential part of the original database. This pertinent set of minimal features can guarantee a solid base for the signal categorization step. The RST-DCA good classification results are, also, explained by the appropriate categorization of each selected signal to its right signal type by using both the reduct and the core

concepts. As for PCA-DCA, it produces less accuracy in comparison to our RST-DCA method which is explained by the inappropriate use of PCA for data pre-processing. In fact, the first components selected are not necessarily the right set of features to retain since this set may contains extra/less features that do not add anything new to the target concept while increasing the cost of acquiring data. The set may, also, contain misleading features which have a negative effect on the algorithm classification accuracy. Furthermore, the PCA-DCA categorization step does not make "sense" as a coherent categorization procedure as it is based on the PCA ranking in terms of variability. The major results of this work were published in (Chelly & Elouedi, 2012b).

RST-DCA is based on assigning the same feature to both the SS and the PAMP signals. We believe that it is important to check if we designate different features to different signal categories then RST will still be more appropriate than PCA for data pre-processing. This will be discussed in the next Section with the development of a new rough DCA.

3.5 RC-DCA

RC-DCA is our second proposed rough DCA. The algorithm is based on the "Reduct" and the "Core" RST concepts. RC-DCA is based on designating different attributes to different signal categories. It means that each signal category has its specific feature. In what follows we will detail the RC-DCA data pre-processing phase.

3.5.1 RC-DCA Feature Selection Process

For feature selection, our RC-DCA applies the same process as our RST-DCA. In other words and as explained previously, RC-DCA will compute the positive regions $POS_C(\{d\})$, $POS_c(\{d\})$ and $POS_{C-\{c\}}(\{d\})$ until finding the minimal subset of attributes R from C. RC-DCA will, also, generate either one reduct or a family of reducts. Thus and in the RC-DCA signal categorization substep, adequate solutions should be proposed to each of these generated cases.

3.5.2 RC-DCA Signal Categorization Process

The Process of One Reduct

In case where RC-DCA generates only one reduct, we have $CORE_D(C) = RED_D(C)$. So for signal categorization, RC-DCA follows the ranking of the signal categories that implies the significance of each signal category to the signal transformation of the standard DCA. The ranking is in the order Safe, PAMP and Danger. Based on this information, our method processes as follows:

First of all, RC-DCA calculates the positive region $POS_C(\{d\})$ of the whole core $CORE_D(C)$. Then, our method calculates the positive regions $POS_{C-\{c\}}(\{d\})$ that correspond to the positive regions of the core when removing each time one attribute c from it. Our method calculates the difference values between $POS_C(\{d\})$ and each $POS_{C-\{c\}}(\{d\})$ already calculated. The removed attribute c causing the highest difference value $POS_C(\{d\}) - POS_{C-\{c\}}(\{d\})$ is

considered as the most important attribute in the core. In other words, when removing that attribute c from the core then the effectiveness and the reliability of the core will be strongly affected. Therefore, our method selects that attribute c to form the SSs. The second attribute c having the next highest difference value $POS_C(\{d\}) - POS_{C-\{c\}}(\{d\})$ is used to form the PAMP signals. Finally, the rest of the core attributes is combined and assigned to represent the DSs.

The Process of a Family of Reducts

In case where RC-DCA produces more than one reduct R, a family of reducts $RED_D^F(C)$, the algorithm uses both of the core $CORE_D(C)$ and the reducts $RED_D^F(C)$ concepts. For signal assignment and based on the positive regions calculations, our method assigns the convenient attributes from the core to determine the SS and PAMP signals following the same reasoning as if RC-DCA produces only one reduct. As for the DSs categorization, our solution chooses first of all randomly a reduct $RED_D(C)$ among $RED_D^F(C)$. Then, RC-DCA combines all the $RED_D(C)$ features except the c attributes already chosen for the SSs and PAMPs and assigns the resulting value to the DSs.

Example 3.2 *To illustrate an example of the RC-DCA functioning, a decision system, Table 3.4, is used. It consists of 9 conditional features (a, b, c, d, e, f, g, l, m), 1 decision feature (X) and 8 objects.*

Table 3.4: The RC-DCA Decision System

$x \in U$	a	b	c	d	e	f	g	l	m	X
x_1	1	2	4	0	1	2	5	3	2	1
x_2	0	3	3	2	1	3	2	2	1	2
x_3	2	3	1	3	3	1	6	1	3	2
x_4	1	1	2	1	2	3	5	3	2	1
x_5	0	2	0	1	2	4	2	3	0	1
x_6	1	1	2	4	3	1	3	1	0	2
x_7	2	2	1	3	2	4	2	2	3	2
x_8	1	2	0	2	2	4	7	0	2	1

As previously stated, RC-DCA follows the same reasoning for feature selection as RST-DCA. Let us assume that RC-DCA generates the following reducts:

- $R = \{\{a, b, c, d, e\}, \{a, b, c, f, g\}, \{a, b, c, l, m\}\}$

Once the reduct set is generated, RC-DCA moves to the signal categorization step. Since, RC-DCA has generated a family of reducts then RC-DCA has to specify the core concept:

- $Core = \{a, b, c, d, e\} \cap \{a, b, c, f, g\} \cap \{a, b, c, l, m\} = \{a, b, c\}$

As RC-DCA uses different features to represent SS and PAMP signals, then the dendritic cell ranking of signal categories has to be respected. Let us remind that the ranking is set, first of all, to SS, then PAMP and finally to DS. For signal categorization, RC-DCA computes, first, the positive region of the whole core:

- $POS(a, b, c) = 6/8$

RC-DCA calculates now the positive regions of the core when discarding each time one feature from it:

- When removing $\{a\}$: $POS(b, c) = 2/8$
- When removing $\{b\}$: $POS(a, c) = 3/8$
- When removing $\{c\}$: $POS(a, b) = 1/8$

RC-DCA calculates the difference values between $POS(a, b, c)$ and each of the three positive regions calculated in the last itemized list:

- $POS(a, b, c) - POS(b, c) = 6/8 - 2/8 = 4/8$
- $POS(a, b, c) - POS(a, c) = 6/8 - 3/8 = 3/8$
- $POS(a, b, c) - POS(a, b) = 6/8 - 1/8 = 5/8$

The removed attribute causing the highest calculated difference value is selected to represent the safe signal as it is seen as the most important feature in the core. The next attribute causing the second highest calculated difference value is chosen to represent the PAMP signal as it is ranked the second one in the signal categories ranking list.

- $SS = \{c\}$
- $PAMP = \{a\}$

As for the danger signal categorization, RC-DCA chooses first of all one reduct among the family of the generated reducts. For example, RC-DCA selects the following reduct:

- $R = \{a, b, c, d, e\}$

Then, RC-DCA removes the already chosen attributes in the previous step and keeps the rest of the features to represent the danger signal.

- $DS = \{b, d, e\}$

The DS value is the mean value of the absolute distances calculated between the DS attribute set values and the mean values calculated using the values of class 1 for each attribute.

3.5.3 Experimental Setup, Results and Discussion

To test the validity of our RC-DCA hybrid model, our experiments are performed on the same databases described in Table 3.2 and same parameters as presented in Section 3.4.3. In this Section, we show that applying RST to DCA is more convenient than when applying PCA for the feature selection and signal categorization processes. This comparison is based on the fact that both RC-DCA and the DCA based PCA, referred as PCA'-DCA, adopt the same reasoning of assigning different features to different signals. PCA'-DCA differs from the previous PCA-DCA in the

signal categorization step. PCA-DCA is based on the idea of assigning the same attribute to both SSs and PAMPs while PCA'-DCA assigns different features to different signals. Applying RST for the DCA data pre-processing phase shows that RC-DCA outperforms the PCA'-DCA in terms of the used evaluation criteria. This is confirmed by the results presented in Table 3.5.

Table 3.5: PCA'-DCA and RC-DCA Comparison Results

Database	♯ Attributes		Sensitivity (%)		Specificity (%)		Accuracy (%)	
	DCA		DCA		DCA		DCA	
	PCA'	RC	PCA'	RC	PCA'	RC	PCA'	RC
SN	28	20	87.38	90.10	81.44	93.82	84.61	91.82
Bio	24	19	54.71	77.35	49.05	79.24	51.88	78.30
CylB	16	7	95.50	97.00	96.47	97.75	96.09	97.46
Ch	14	11	94.36	98.80	94.30	98.88	94.33	98.84
IONO	24	19	94.44	96.82	96.44	97.33	95.72	97.15
Sck	22	20	95.23	96.96	96.30	97.68	96.23	97.64
HC	19	14	87.50	93.05	90.13	94.73	88.58	93.75
GC	17	17	83.57	89.05	90.35	90.77	88.50	90.30
LR	10	5	85.00	90.00	89.18	91.89	87.71	91.22
RWW	10	6	98.81	99.22	98.81	99.49	98.81	99.29

From Table 3.5 and in terms of number of selected features, we can see that our RC-DCA selects fewer features than PCA'-DCA. This characteristic holds for our RC-DCA as it is based on the same process of feature selection as RST-DCA. From Table 3.5, we can also notice that the classification accuracy of our proposed RC-DCA is notably better than the one given by PCA'-DCA. For example, when applying the RC-DCA to the Ch database, the RC-DCA accuracy is set to 98.84%. Nevertheless, when applying the PCA'-DCA to the same database, the accuracy is 94.33%. Same remark is noticed for both the sensitivity and the specificity criteria where our RC-DCA is capable of producing relatively higher true positive rates and higher true negative rates than the sensitivity and specificity values of the PCA'-DCA.

The RC-DCA good classification results are explained by the appropriate categorization of each selected signal to its right signal type. Using both the reduct and the core concepts, RC-DCA assigns to each attribute constituting the reduct its convenient signal; i.e., either a DS, a SS or a PAMP signal. Based on these indispensable and most informative features for signal categorization, RC-DCA can produce a coherent signal database for its next algorithmic steps. The major results of this work were published in (Chelly & Elouedi, 2012a).

In this part of this Chapter, we have introduced a second hybrid DCA based on RST. Yet, it is important to remind the reader that in both RST-DCA and RC-DCA Sections, the proposed algorithms generate either one reduct or a set of reducts. Obviously, only one reduct is needed. This leads us to think about the development of a new rough dendritic cell approach that should generate only one reduct and avoid the fact of generating all possible reducts. This third model will be discussed in the next Section.

3.6 QR-DCA

QR-DCA is based on the use of an heuristic, the QuickReduct algorithm, to perform the feature selection phase. This is to tackle the limitations of the previous rough proposed algorithms. Details of QR-DCA are presented in what follows.

3.6.1 Why Do We Need an Heuristic?

As explained in the previous Sections, both RST-DCA and RC-DCA generate either a family of reducts or one reduct and both algorithms propose solutions to handle both cases. However, most of the time only one reduct is required to reduce a data set. So, all the calculations involved in discovering the rest of the reducts are pointless. Moreover, the fact of generating a family of reducts is only practical for very simple data sets.

One way out of this exhaustive search is to attempt to calculate a reduct without exhaustively generating all possible reducts. On this idea is based our newly proposed QR-DCA. In fact, the objective of our QR-DCA is to find a trade-off between generating satisfactory classification results while avoiding all the extra calculations performed by both RST-DCA and RC-DCA. QR-DCA will basically generate only one reduct. This will be achieved by the use of the QuickReduct algorithm described by Algorithm 3.1. Based on the generated reduct, QR-DCA will categorize different features to different signal categories just like the idea of our previously discussed RC-DCA algorithm, and not like RST-DCA. The choice of this specific signal categorization methodology will be later argued in the experimental setup Section.

Algorithm 3.1 The QuickReduct Algorithm

```
C: the set of all conditional features;
D: the set of decision features;
R ← {};
do
    T ← R
    ∀ x ∈ (C - R);
    if γ_{R∪{x}}(D) > γ_T(D);
     T ← R ∪ {x};
    end if
    R ← T;
until γ_R(D) == γ_C(D)
return R
```

3.6.2 QR-DCA Feature Selection Process

For feature selection, QR-DCA computes, first of all, the dependency of the entire database $\gamma_C(D)$. To do so, QR-DCA has to calculate the positive region for the whole attribute set C: $POS_C(D)$. Once the consistency of the database is measured and by applying the QuickReduct algorithm, QR-DCA starts off with an empty set and moves to calculate the dependency of each attribute c apart: $\gamma_c(D)$. The attribute c having the greatest value of dependency

is added to the empty set. Once the first attribute c is selected, QR-DCA adds, in turn, one attribute to the selected first attribute and computes the dependency of each obtained attributes couple $\gamma_{\{c,c_i\}}(D)$. The algorithm chooses the couple having the greatest dependency degree. The process of adding each time one attribute to the subset of the selected features continues until the dependency of the obtained subset equals the consistency of the entire database already calculated; i.e., $\gamma_C(D)$.

3.6.3 QR-DCA Signal Categorization Process

From the previous definitions of each signal category, both PAMPs and SSs are more informative than DSs. To represent this level of importance, QR-DCA uses the first obtained couple of features through the reduct generation. On the other hand, the DSs are less informative than PAMPs and SSs. Therefore, QR-DCA applies the rest of the reduct attributes, discarding the two first selected attributes that are chosen to represent the SSs and PAMPs, to represent the DSs. More precisely, QR-DCA processes as follows:

As QR-DCA has already calculated the dependency of each attribute c apart, $\gamma_c(D)$, QR-DCA selects the attribute c having the greatest dependency degree to form the SSs. This is because that c feature is considered as the most informative attribute added to the reduct. With no additional computations and since QR-DCA has already computed the dependency of each attributes couple $\gamma_{\{c,c_i\}}(D)$ when adding, in turn, one attribute c_i to the selected first attribute c that represents the SSs, QR-DCA chooses the couple having the greatest dependency degree. More precisely, QR-DCA selects that second attribute c_i to form the PAMP signals. Finally, the rest of the reduct attributes are combined and assigned to represent the DSs as they are less than certain to be anomalous.

Example 3.3 *To illustrate an example of the QR-DCA functioning, a decision system, Table 3.6, is used. It consists of 5 conditional features (a, b, c, d, e), 1 decision feature (X) and 8 objects.*

Table 3.6: The QR-DCA Decision System

$x \in U$	a	b	c	d	e	X
x_1	1	2	4	0	1	1
x_2	0	3	3	2	1	2
x_3	2	3	1	3	3	2
x_4	1	1	2	1	2	1
x_5	0	2	0	1	2	1
x_6	1	1	2	4	3	2
x_7	2	2	1	3	2	2
x_8	1	2	0	2	2	1

As previously stated, QR-DCA follows the same reasoning for feature selection as the previous developed method, RC-DCA, when dealing with the signals ranking. QR-DCA applies the QuickReduct algorithm to select only one reduct to avoid all the extra non-needed calculations. To select the most important features from the mentioned antigen database, QR-DCA processes as follows: first of all, QR-DCA calculates the dependency of the whole antigen data set:

- $\gamma_{\{a,b,c,d,e\}}(X) = 8/8 = 1$

The reduct is set to the empty set:

- $R = \emptyset$

After this initialization, QR-DCA evaluates the dependency of each attribute:

- $\gamma_{\{a\}}(X) = 2/8$
- $\gamma_{\{b\}}(X) = 0/8$
- $\gamma_{\{c\}}(X) = 1/8$
- $\gamma_{\{d\}}(X) = 1/8$
- $\gamma_{\{e\}}(X) = 1/8$

As $\{a\}$ *generates the highest dependency degree, it is added to the reduct:*

- $R = \{a\}$

Again, QR-DCA evaluates the following dependency values:

- $\gamma_{\{a,b\}}(X) = 0/8$
- $\gamma_{\{a,c\}}(X) = 1/8$
- $\gamma_{\{a,d\}}(X) = 3/8$
- $\gamma_{\{a,e\}}(X) = 1/8$

As $\{d\}$ *generates the highest dependency degree, it is added to the reduct:*

- $R = \{a, d\}$

QR-DCA evaluates the following dependency values:

- $\gamma_{\{a,d,b\}}(X) = 8/8$
- $\gamma_{\{a,d,c\}}(X) = 5/8$
- $\gamma_{\{a,d,e\}}(X) = 6/8$

As $\{b\}$ *generates the highest dependency degree, it is added to the reduct. In addition, QR-DCA stops the iterations since the generated dependency is equal to the dependency of the whole data set* ($\gamma_{\{a,d,b\}}(X) = 8/8 = \gamma_{\{a,d,c,d,e\}}(X) = 1$)*. Thus, the generated reduct is the following:*

- $R = \{a, d, b\}$

For the signal categorization process and with respect to the signal categories ranking, QR-DCA processes as follows: the first attribute added to the reduct, as it has the highest dependency degree, is chosen to represent the SSs. The next attribute added to the reduct is chosen to represent the PAMP signals. The rest of the reduct attributes are combined and the resulting value is assigned to the DSs. This is summarized in the following list:

- $SS = \{a\}$
- $PAMP = \{d\}$
- $DS = \{b\}$

Once the selected features are assigned to their suitable signal types, RST-DCA, RC-DCA and QR-DCA calculate the values of each signal category using the same process as the standard DCA (Greensmith, 2007). The output is, thus, a new information table which reflects the signal database. In fact, the universe U of the induced signal data set is $U = \{x_1^{'}, x_2^{'}, \ldots, x_N^{'}\}$ a set of antigen identifiers and the conditional attribute set $C = \{SS, PAMP, DS\}$ contains the three signal types; i.e., the SSs, the PAMPs and the DSs. The induced signal database which is the input data for the next proposed models steps contains the values of each signal type for each antigen identifier. Finally and once data pre-processing is achieved, our solution approaches process their next steps which are the Detection Phase, the Context Assessment and the Classification Procedure as the standard DCA does and as described in Chapter 1.

3.6.4 Experimental Setup, Results and Analysis

To test the validity of our QR-DCA hybrid model, our experiments are performed on the same databases described in Table 3.2 and by using the same parameters presented in Section 3.4.3. In this Section and in order to compare QR-DCA to both RST-DCA and RC-DCA, we will add the execution time (t) criterion to the previously used criteria; i.e., number of selected features, specificity, sensitivity and accuracy. (t) quantifies the amount of time taken by an algorithm to run as a function of the size of the input to the problem. The execution time is measured in seconds (s). This criterion is essential in this part as we have to measure the effect of the heuristic on the feature selection process, with QR-DCA, in comparison to both RST-DCA and RC-DCA that generate several reducts. All experiments are run on an Sony Vaio G4 2.67 Ghz machine. Indeed, we will focus on the number of selected features to notice the influence of the QuickReduct algorithm on the generated reduct set. Based on these features, we will compare the algorithms' classification results.

Let us remind that the common idea between QR-DCA and RC-DCA is to assign for each selected feature a specific signal category; i.e., either as SSs, DSs or PAMPs. Nevertheless, RC-DCA generates all the possible reducts which is a time consuming task. In addition, RC-DCA differs from QR-DCA in the categorization step which focuses on proposing different solutions in case where this method produces one reduct or a family of reducts. Adding to this, the main difference between RST-DCA and both QR-DCA and RC-DCA is that RST-DCA assigns only one attribute to form both SSs and PAMPs as they are seen as the most important signals. As for the DSs categorization, RST-DCA combines the rest of the reduct features and assigns the resulting value to the DSs. Like RC-DCA, RST-DCA generates all the possible reducts and proposes solutions to handle both cases; i.e., generating one reduct or a family of reducts. The three algorithms are compared and the results are displayed in Table 3.7.

In Section 3.3, we have set two hypotheses, therefore, in this Section we will compare our three developed rough DCA versions while checking the validity of H1 and H2. From the results obtained from Table 3.7, we will first

show that assigning for each selected feature a specific signal category leads to a better performance than assigning the same attribute to both SSs and PAMPs. Secondly, we will show that our QR-DCA proposed solution can find the trade-off between generating satisfactory classification results and processing in less time than both RC-DCA and RST-DCA.

Comparison in Terms of Selected Features

From Table 3.7, we can notice that RST-DCA and RC-DCA have the same number of selected features. This is explained by the fact that both algorithms are based on the same feature selection phase. They generate all the possible reducts and choose the one having the smallest number of features. However, our QR-DCA new version has either the same number of features as both RST-DCA and RC-DCA or more features. This is explained by the fact that QR-DCA, by applying the QuickReduct algorithm, follows the features path that generates the highest dependency degree. Consequently, the taken path may either lead to a final reduct including the smallest number of features or to a path including more selected features.

As an example, we may take the IONO database. When applying QR-DCA to the latter base the number of selected attributes is 22. However, when applying RST-DCA or RC-DCA to the same data set the number of selected features is 19. When applying the three rough DCAs to the Ch database, the number of the selected features is the same: 11. We have, also, to mention that obtaining the same number of features does not mean that this reduct includes the same attributes; i.e., the attributes may differ.

Checking the Validity of H1

Based on the selected attributes, the accuracies of the algorithms are calculated. From Table 3.7, we can notice that the differences between the classification accuracies generated by both QR-DCA and RC-DCA are not significant. Thus, we can say that QR-DCA has nearly the same classification performance as RC-DCA. Same remark is noticed for the sensitivity and specificity criteria. For instance, when applying the algorithms to the Sck data set, the classification accuracy of QR-DCA is set to 97.58% and when applying RC-DCA to the same database the accuracy is set to 97.64%. In some databases, the QR-DCA classification accuracy is a bit less than the one generated by RC-DCA and sometimes a bit higher.

From Table 3.7, we also remark that in all databases QR-DCA and RC-DCA outperform RST-DCA in terms of sensitivity, specificity and classification accuracy. For instance, the classification accuracy of RST-DCA when applied to the SN database is set to 85.57% which is less than 90.86% and 91.82% generated respectively by QR-DCA and RC-DCA. This is explained by the fact that RST-DCA differs from QR-DCA and RC-DCA in the signal categorization phase. Both QR-DCA and RC-DCA assign different features to different signal categories. However, RST-DCA uses the same attribute to assign it for both SSs and PAMPs. As for the DSs categorization, RST-DCA combines the rest of the reduct features and assign the resulting value to the DSs. From these results, we can conclude that it is crucial to assign for each signal category a specific and different feature. Consequently, H1 is confirmed.

Table 3.7: Comparison Results of the Rough DCAs

Database	♯ Attributes			Sensitivity (%)			Specificity (%)			Accuracy (%)			Time (s)		
	DCA			DCA			DCA			DCA			DCA		
	QR	RC	RST	QR	RC	RST	QR	RC	RST	QR	RC	RST	QR	RC	RST
SN	22	20	20	89.19	90.10	88.28	92.79	93.82	82.47	90.86	91.82	85.57	7.79	1705.79	1687.55
Bio	19	19	19	77.35	77.35	52.83	79.24	79.24	52.83	78.30	78.30	52.83	5.25	1679.53	1601.20
CylB	7	7	7	97.00	97.00	96.50	97.75	97.75	96.79	97.46	97.46	96.67	12.68	1444.93	1398.12
Ch	11	11	11	98.80	98.80	97.84	98.88	98.88	98.23	98.84	98.84	98.02	571.05	1779.83	1697.01
IONO	22	19	19	96.03	96.82	95.23	96.88	97.33	96.88	96.58	97.15	96.29	15.88	668.32	591.13
Sck	22	20	20	96.53	96.96	96.53	97.65	97.68	96.83	97.58	97.64	96.81	510.05	1401.43	1387.11
HC	14	14	14	93.05	93.05	87.50	94.73	94.73	91.44	93.75	93.75	89.13	16.96	260.08	90.16
GC	17	17	17	89.05	89.05	83.21	90.77	90.77	90.22	90.30	90.30	88.30	136.02	533.72	499.74
LR	5	5	5	90.00	90.00	90.00	91.89	91.89	86.48	91.22	91.22	87.71	0.59	0.75	0.66
RWW	8	6	6	99.02	99.22	98.01	99.43	99.49	98.24	99.12	99.29	98.07	987.12	2201.98	2198.99

Checking the Validity of H2

QR-DCA takes less time to process than RC-DCA and RST-DCA. This is confirmed by the results appearing in Table 3.7. For example, when applying the algorithms to the CylB database, the amount of time taken by QR-DCA to process is 12.68(s) which is less than the times taken by RC-DCA and RST-DCA which are 1441.93(s) and 1398.12(s), respectively. The QR-DCA lightweight in terms of running time is explained by the advantage of using the QuickReduct algorithm as it attempts to calculate a reduct without exhaustively generating all possible subsets. In contrast, both RST-DCA and RC-DCA generate all possible subsets and retrieve those with a maximum rough set dependency degree. Obviously, this is an expensive solution to the problem. Most of the time, only one reduct is required to reduce a data set. So, all the calculations involved in discovering the rest of the reduct sets are pointless. Consequently, we can confirm our second hypothesis H2.

From these notes, we can confirm that QR-DCA which is based on an heuristic can find a trade-off between generating good classification results and processing in less time than both RC-DCA and RST-DCA. The major results of this part were published in (Chelly & Elouedi, 2013e).

Comparison with Some State-of-the-art Classifiers

The performance of our QR-DCA is compared to other classifiers including the Support Vector Machine (SVM), Artificial Neural Network (ANN) and the Decision Tree (DTree). Our main objective is to emphasize the performance of these classifiers in terms of classification accuracy. Hence, the comparison made is in terms of the average of accuracies of the databases presented in Table 3.2.

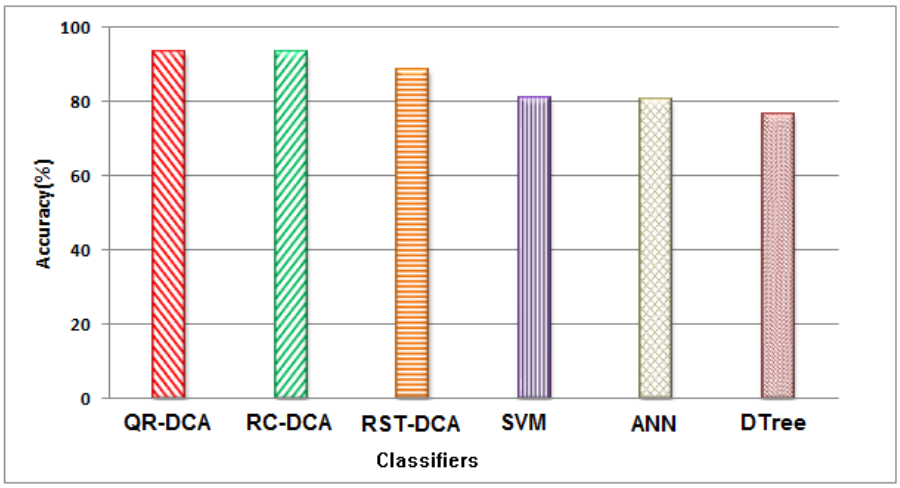

Figure 3.2: Comparison of Classifiers Average Accuracies

The parameters of SVM, ANN and DTree are set to the most adequate parameters to these algorithms using the Weka software. Figure 3.2 shows that QR-DCA has nearly the same classification performance as RC-DCA. It, also, shows that QR-DCA outperforms RST-DCA, SVM, ANN and DTree.

3.7 Conclusion

In this Chapter, we have elucidated how rough set theory could be used to select the right features and to categorize each selected attribute to its signal category. We have developed three rough dendritic cell algorithms, compared them and noted the appropriate observations while testing two crucial hypotheses.

Despite of the noticed advantages of our proposed rough set methods, they have some limitations when dealing with real-valued attributes. This prompted our research into the use of fuzzy-rough sets for feature selection and signal categorization. This will be dealt with in the next Chapter.

Chapter 4

A Fuzzy-Rough DCA Pre-Processing Module

4.1 Introduction

In this Chapter, we still focus on the DCA data pre-processing phase. We propose novel approaches to more improve the standard DCA data pre-processing task. This is achieved through the use of Fuzzy Rough Set Theory (FRST) (Dubois & Prade, 1992). The proposed algorithms are seen as extensions to the rough set algorithms which were previously presented in Chapter 3.

The rest of this Chapter is organized as follows: Section 4.2 deals with the motivation behind the use of FRST for data pre-processing. Our developed DCA variants are explained in details in Sections 4.3, 4.4 and 4.5. A comparison of our proposed algorithms is presented in Section 4.6.

4.2 Why Adopting Fuzzy-Rough Sets for Data Pre-processing?

4.2.1 Criticisms of the Rough DCAs

In most of cases, the values of attributes may be real-valued and this is where rough set theory encounters a problem. It is not possible in the theory to say whether two attribute values are similar and to what extent they are the same. For example, two close values may only differ as a result of noise. However, in RST they are considered to be as different as two values of a different order of magnitude. One answer to this problem has been to discretize the data set beforehand producing a new data set with crisp values. This is often still inadequate as it is a source of information loss which is against the rough set ideology of retaining information content (Jensen & Shen, 2002).

In Chapter 3, we have proposed different DCAs based on rough set theory for data pre-processing. As those methods apply RST then they have to perform data discretization before selecting the necessary features. We believe that this information loss, due to the quantization process, may influence the rough DCAs feature selection task. This is because the selected set of features may differ from the correct set that had to be generated, initially, from the input database. As a consequence, this will misguide the algorithms categorization phase by assigning different features to different signal categories instead of their right signal types. As a result, this will influence the algorithms classification task.

One possible solution to overcome these limitations is to apply Fuzzy Rough Set Theory (FRST) as a pre-processor. FRST encapsulates the related but distinct concepts of vagueness and indiscernibility both of which occur as a result of uncertainty in data. A method employing fuzzy-rough sets can handle this uncertainty. Therefore, in this Chapter, we propose to develop novel DCAs based on different fuzzy-rough feature selection and signal categorization processes.

4.2.2 The Fuzzy-Rough Proposed Solutions

We have proposed three main fuzzy-rough DCAs. The first proposed algorithm, named FRST-DCA, is based on the FRST concepts mainly fuzzy sets and membership functions. The second approach, dubbed FLA-DCA, is based on the Fuzzy Lower Approximation. Lastly, the third algorithm is named FBR-DCA as it is based on the Fuzzy Boundary Region. The proposed fuzzy-rough DCAs are all based on the same layout which was presented, previously, in Figure 3.1. However, instead of applying RST our models are based on the use of FRST in the data pre-processing phase. In what follows, we will principally describe the data pre-processing step of our proposed fuzzy-rough DCA variants. This is because the rest of the algorithms steps including the Detection phase, the Context Assessment and the Classification procedure are performed the same as the standard DCA and as described, previously, in the first Chapter.

4.3 FRST-DCA

In this Section, we present the data pre-processing phase of our first developed fuzzy-rough DCA. We detail its feature selection and its signal categorization steps.

4.3.1 The FRST-DCA Feature Selection Process

Same as previously done with our proposed rough DCAs, we firstly have to formalize the current context. Our problem may be seen as an information table where universe $U = \{x_1, x_2, \ldots, x_N\}$ is a set of antigen identifiers, the conditional attribute set $C = \{c_1, c_2, \ldots, c_A\}$ contains each feature of the information table to select and the decision attribute D, having the binary values d_k, corresponds to the class label of each sample.

For feature selection, FRST-DCA applies the Fuzzy-Rough QuickReduct algorithm which was presented in Chapter 2 and more precisely in Section 2.4.6. FRST-DCA computes, first of all, the fuzzy lower approximations of the two decision concepts d_k, for all attributes c_i and for all objects x_j; denoted by $\mu_{\underline{c_i}\{d_k\}}(x_j)$. Using these results,

FRST-DCA calculates the fuzzy positive regions for all c_i, for each object x_j defined as $\mu_{POS_{c_i}(D)}(x_j)$. To find the reduct, FRST-DCA starts off with an empty set and moves to calculate the fuzzy dependency degrees of D on c_i, defined as $\gamma'_{c_i}(D)$. The attribute c_m having the greatest value of the fuzzy-rough dependency degree is added to the empty reduct set. Once the first attribute c_m is selected, FRST-DCA adds in turn one attribute to the selected first attribute and computes the fuzzy-rough dependency degree of each obtained attributes couple $\gamma'_{\{c_m,c_i\}}(D)$. The algorithm chooses the couple having the greatest fuzzy-rough dependency degree. The process of adding each time one attribute to the subset of the selected features continues until no increase in the fuzzy-rough dependency degree is observed.

4.3.2 The FRST-DCA Signal Categorization Process

The second step of our FRST-DCA data pre-processing phase is signal categorization. More precisely, our method is based on the semantics of the DCA signals and it is, also, based on the same signal categorization methodology used by our RC-DCA/QR-DCA rough models. We have selected this same methodology because we have shown in (Chelly & Elouedi, 2013e) that assigning different features to different signal types is more appropriate for the DCA data pre-processing phase instead of assigning the same feature to both SSs and PAMPs. Based on these points, FRST-DCA processes as follows:

As FRST-DCA has already calculated the fuzzy-rough dependency degree of each attribute c_i apart, $\gamma'_{c_i}(D)$, FRST-DCA selects the first attribute c_m having the greatest fuzzy-rough dependency degree. That attribute is chosen to form the SSs as it is considered the most informative first feature added to the reduct. With no additional computations and since FRST-DCA has already computed the fuzzy-rough dependency degrees of each attributes couple $\gamma'_{\{c_m,c_i\}}(D)$, FRST-DCA chooses the couple having the greatest dependency degree. More precisely, FRST-DCA selects that second attribute c_r having the greatest $\gamma'_{\{c_m,c_r\}}(D)$ among the calculated $\gamma'_{\{c_m,c_i\}}(D)$. c_r is chosen to form the PAMP signals. Finally, the rest of the reduct attributes are combined and affected to represent the DSs.

Example 4.1 *To illustrate the operation of our FRST-DCA, we will use the database given, previously, in Table 2.5. Let us remind, that by applying the fuzzy-rough QuickReduct algorithm, the algorithm generated the following reduct:*

- $\{a, b\}$

For the signal categorization process and with respect to the signal categories ranking, FRST-DCA processes as follows:

The first attribute added to the reduct, as it has the highest fuzzy dependency degree, is chosen to represent the safe signals. The next attribute added to the reduct is chosen to represent the PAMP signals. The rest of the reduct attributes are combined and the resulting value is assigned to the danger signals. This is summarized in the following list:

- $SS = \{b\}$
- $PAMP = \{a\}$
- $DS = \emptyset$

4.3.3 Experimental Setup, Results and Analysis

Experimental Setup

To test the validity of our developed FRST-DCA, our experiments are performed on the same databases presented in Table 3.2 and the same parameters presented in Section 3.4.3. The used data sets are composed of real-valued attributes. Let us remind that in (Chelly & Elouedi, 2012b, 2012a, 2013e), the feature selection step is achieved by applying RST. Nevertheless, the data sets are entirely composed of real-valued attributes and therefore discretization had to be performed. This is clearly a potential source of information loss. By applying the present work, FRST-DCA, such loss can be reduced as attribute values are kept unchanged; i.e., no quantization is performed on the original databases.

In this Section, we will compare FRST-DCA to the rough QR-DCA approach. We will limit the comparison to QR-DCA, only, because both of these two algorithms nearly share the same characteristics. Both FRST-DCA and QR-DCA generate only one reduct. Both algorithms are based on the same process of feature selection and same procedure of signal categorization. The main difference is that QR-DCA, unlike FRST-DCA, has to perform data discretization before performing the feature selection step. RST-DCA and RC-DCA will not be added to the current comparison since these algorithms do not apply the QuickReduct heuristic. The comparison will be based on the number of selected features as we are focusing on the data pre-processing phase, sensitivity, specificity and accuracy. Yet, it would be interesting to check the influence of avoiding the discretization step and its replacement with the fuzzy process in FRST-DCA in terms of running time. Therefore, the execution time criterion is added for sake of comparison.

On the other hand, in (Jensen & Shen, 2007), FRST as a feature selection technique has been experimentally evaluated with other leading feature selection techniques such as Relif-F and entropy-based approaches. It has been shown that FRST outperforms several feature selection approaches in terms of resulting classification performance. Hence, only comparison to fuzzy rough and rough set theories are given here.

Discussion of the Obtained Results

From Table 4.1, we can notice that FRST-DCA has fewer features than QR-DCA. This is explained by the fact that FRST-DCA incorporates the information usually lost in crisp discretization by utilizing the generated fuzzy-rough sets to provide a more informed technique. Results show that FRST-DCA selects features without much loss in information content. For instance, applying FRST-DCA to the Bio database, the number of selected attributes is 13. However, when applying QR-DCA to the same database the number of selected features is set to 19. A second example can be the HC data set where the number of selected features, by applying FRST-DCA, is reduced by 50% (7 features) in comparison to the number of features selected by the QR-DCA algorithm which is set to 14.

From Table 4.1, we can also notice that FRST-DCA outperforms QR-DCA in terms of classification accuracy. For instance, when applying the algorithms to the Sck data set, the classification accuracy of FRST-DCA is set to 99.28%. However, when applying QR-DCA to the same database, the accuracy is set to 97.58%. Same remark is observed for the specificity and the sensitivity criteria. When comparing the results in terms of running time, we can notice that FRST-DCA needs more time to process than QR-DCA due to the attachment of the fuzzy component.

For example, when applying the algorithms to the Bio database, the amount of time taken by FRST-DCA to process is 47.29(s) which is a bit more than the time taken by QR-DCA which is set to 5.25(s). These results were published in (Chelly & Elouedi, 2013c, 2014b).

Table 4.1: Comparison Results of QR-DCA and FRST-DCA

Database	♯ Attributes		Sensitivity (%)		Specificity (%)		Accuracy (%)		Time (s)	
	DCA		DCA		DCA		DCA		DCA	
	QR	FRST	QR	FRST	QR	FRST	QR	FRST	QR	FRST
SN	22	16	89.19	94.60	92.79	95.88	90.86	95.19	7.79	95.91
Bio	19	13	77.35	84.90	79.24	86.79	78.30	85.84	5.25	47.29
CylB	7	7	97.00	98.00	97.75	98.71	97.46	98.43	12.68	118.41
Ch	11	4	98.80	99.28	98.88	98.69	98.84	98.99	571.05	1047.25
IONO	22	9	96.03	97.61	96.88	97.77	96.58	97.72	15.88	130.28
Sck	22	16	96.53	97.40	97.65	99.40	97.58	99.28	510.05	947.25
HC	14	7	93.05	97.22	94.73	98.68	93.75	97.82	16.96	184.56
GC	17	17	89.05	90.87	90.77	92.28	90.30	91.90	136.02	335.58
LR	5	5	90.00	90.00	91.89	91.89	91.22	91.22	0.59	0.64
RWW	8	4	99.02	99.63	99.43	99.56	99.12	99.61	987.12	1809.23

Our FRST-DCA new approach performs much better than traditional RST on the whole; i.e., in terms of both feature selection and classification quality. Despite of these FRST-DCA satisfactory results, the algorithm is a bit "slow" in comparison to QR-DCA. Trying to focus on this limitation, we have developed a second fuzzy-rough DCA where its details are presented in the next Section.

4.4 FLA-DCA

This Section presents our fuzzy-rough FLA-DCA approach. FLA-DCA is based on the Fuzzy Lower Approximation and fuzzy similarity relations.

4.4.1 FRST Main Concepts Based on Similarity Relations

Redefining the Fuzzy Rough Set

The concept of crisp equivalence classes can be extended by the inclusion of a fuzzy similarity relation S on the universe, which determines the extent to which two elements are similar in S (Jensen & Shen, 2002). The fuzzy lower and fuzzy upper approximations become:

$$\mu_{\underline{R_P}X}(x) = inf_{y \in U} I(\mu_{R_P}(x, y), \mu_X(y)) \tag{4.1}$$

and

$$\mu_{\overline{R_P}X}(x) = sup_{y \in U} T(\mu_{R_P}(x, y), \mu_X(y)) \tag{4.2}$$

In the presented formulae, I is a fuzzy implicator and T is a t-norm. R_P is the fuzzy similarity relation induced by the subset of features P defined as:

$$\mu_{R_P}(x, y) = \bigcup_{a \in P} \{\mu_{R_a}(x, y)\} \tag{4.3}$$

where $\mu_{R_a}(x, y)$ is the degree to which objects x and y are similar for feature a. A fuzzy similarity relation can be constructed for this purpose, defined as (Jensen & Shen, 2002):

$$\mu_{R_a}(x, y) = max\left(min\left(\frac{(a(y) - (a(x) - \sigma_a))}{(a(x) - (a(x) - \sigma_a))}, \frac{((a(x) + \sigma_a) - a(y))}{((a(x) + \sigma_a) - a(x))}\right), 0\right) \tag{4.4}$$

where σ_a is the standard deviation of feature a. The fuzzy lower approximation contains information regarding the extent of certainty of object membership to a given concept. The fuzzy upper approximation contains information regarding the degree of uncertainty of objects and the tuple $< \underline{P}(X), \overline{P}(X) >$ is called a *fuzzy-rough set*.

Reduction Process

To search for the optimal subset of features, the fuzzy-rough reduct, the fuzzy positive region has to be calculated. Formally, in the traditional RST, the crisp positive region is defined as the union of the lower approximations. By the extension to the fuzzy principal, the membership of an object $x \in U$ belonging to the *fuzzy positive region* can be defined by:

$$\mu_{POS_{R_P(Q)}}(x) = sup_{X \in U/Q} \mu_{\underline{R_P}X}(x) \tag{4.5}$$

Object x will not belong to the fuzzy positive region only if the fuzzy equivalence class it belongs to is not a constituent of the fuzzy positive region. Using the definition of the fuzzy positive region, the *fuzzy-rough dependency function* can be defined as follows:

$$\gamma'_P(Q) = \frac{\sum_{x \in U} \mu_{POS_{R_P(Q)}(x)}}{|U|} \tag{4.6}$$

As with rough sets, the dependency of Q on P is the proportion of objects that are discernible out of the entire data set. In the present approach, this corresponds to determining the fuzzy cardinality of $\mu_{POS_{R_P(Q)}}(x)$ divided by the total number of objects in the universe. The Fuzzy-Rough QuickReduct algorithm, defined previously, is modified and is based on this measure. According to the new Fuzzy-Rough QuickReduct algorithm, the fuzzy dependency degree of the addition of each attribute to the reduct R (initially empty) is calculated, and the best candidate is chosen. This process continues until the fuzzy dependency of the subset equals the fuzzy dependency degree of the entire data set; i.e., $\gamma'_R(D) = \gamma'_C(D)$.

4.4.2 The FLA-DCA Feature Selection Process

For feature selection, FLA-DCA computes first of all the fuzzy lower approximations of the two decision concepts d_k, for all attributes c_i and for all objects x_j; denoted by $\mu_{\underline{R_{c_i}}\{d_k\}}(x_j)$. Using these results, FLA-DCA calculates the fuzzy positive regions for all c_i, for each object x_j; defined as $\mu_{POS_{R_{c_i}}(D)}(x_j)$. Based on these calculations and to find the fuzzy-rough reduct, FLA-DCA starts off with an empty set and moves to calculate the fuzzy dependency degrees of D on c_i, defined as $\gamma'_{c_i}(D)$. The attribute c_m having the greatest value of the fuzzy-rough dependency degree is added to the empty fuzzy-rough reduct set. Once the first attribute c_m is selected, FLA-DCA adds in turn one attribute to the selected first attribute and computes the fuzzy-rough dependency degree of each obtained attributes couple $\gamma'_{\{c_m, c_i\}}(D)$. The algorithm chooses the couple having the greatest fuzzy-rough dependency degree. The process of adding each time one attribute to the reduct continues until the fuzzy-rough dependency degree of the obtained set of features equals the fuzzy-rough dependency degree, $\gamma'_C(D)$, of the entire database.

4.4.3 The FLA-DCA Signal Categorization Process

Once features are selected, FLA-DCA has to assign for each selected attribute its right signal category. To achieve this task, FLA-DCA follows the same steps as FRST-DCA. In other words, FLA-DCA selects the first attribute c_m having the greatest fuzzy-rough dependency degree to form the SSs. To form the PAMP signals, FLA-DCA selects the second feature having the the greatest dependency degree among the calculated $\gamma'_{\{c_m,c_i\}}(D)$. Finally, the rest of the reduct attributes are combined and affected to represent the DSs.

Example 4.2 *To illustrate the operation of our FLA-DCA, we use the same database given in Table 2.5. The fuzzy connectives chosen for this example are the Lukasiewicz t-norm* ($max(x + y - 1, 0)$) *and the Lukasiewicz fuzzy implicator* ($min(1 - x + y, 1)$).

Using the fuzzy similarity measure defined in Equation 4.4, the resulting relations are as follows for each feature in the data set:

$$R_a(x,y) = \begin{pmatrix} 1.0 & 1.0 & 0.699 & 0.0 & 0.0 & 0.0 \\ 1.0 & 1.0 & 0.699 & 0.0 & 0.0 & 0.0 \\ 0.699 & 0.699 & 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 & 0.699 & 0.699 \\ 0.0 & 0.0 & 0.0 & 0.699 & 1.0 & 1.0 \\ 0.0 & 0.0 & 0.0 & 0.699 & 1.0 & 1.0 \end{pmatrix}$$

$$R_b(x,y) = \begin{pmatrix} 1.0 & 0.0 & 0.568 & 1.0 & 1.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.137 \\ 0.568 & 0.0 & 1.0 & 0.568 & 0.568 & 0.0 \\ 1.0 & 0.0 & 0.568 & 1.0 & 1.0 & 0.0 \\ 1.0 & 0.0 & 0.568 & 1.0 & 1.0 & 0.0 \\ 0.0 & 0.137 & 0.0 & 0.0 & 0.0 & 1.0 \end{pmatrix}$$

$$R_c(x,y) = \begin{pmatrix} 1.0 & 0.0 & 0.036 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.036 & 0.518 & 0.518 & 0.518 \\ 0.036 & 0.036 & 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.518 & 0.0 & 1.0 & 1.0 & 1.0 \\ 0.0 & 0.518 & 0.0 & 1.0 & 1.0 & 1.0 \\ 0.0 & 0.518 & 0.0 & 1.0 & 1.0 & 1.0 \end{pmatrix}$$

The first step is to compute the lower approximations of each concept for each feature. Considering feature a and the decision concept $\{x_1, x_3, x_6\}$ *in the example data set:*

$$\mu_{\underline{R_a}\{x_1,x_3,x_6\}}(x) = inf_{y \in U} I(\mu_{R_a}(x,y), \mu_{\{x_1,x_3,x_6\}}(y))$$

For antigen 3, this is:

$$\mu_{\underline{R_a}\{x_1,x_3,x_6\}}(x_3) = inf_{y \in U} I(\mu_{R_a}(x_3, y), \mu_{\{x_1,x_3,x_6\}}(y))$$

$$= inf\{I(0.699,1), I(0.699,0), I(1,1), I(0,0), I(0,0), I(0,1)\}$$

$$= 0.301$$

For the remaining antigens, this is:

$$\mu_{\underline{R_a}\{x_1,x_3,x_6\}}(x_1) = 0.0 \; ; \; \mu_{\underline{R_a}\{x_1,x_3,x_6\}}(x_2) = 0.0$$
$$\mu_{\underline{R_a}\{x_1,x_3,x_6\}}(x_3) = 0.301 \; ; \; \mu_{\underline{R_a}\{x_1,x_3,x_6\}}(x_4) = 0.0$$
$$\mu_{\underline{R_a}\{x_1,x_3,x_6\}}(x_5) = 0.0 \; ; \; \mu_{\underline{R_a}\{x_1,x_3,x_6\}}(x_6) = 0.0$$

For concept x_2, x_4, x_5, the lower approximations are:

$$\mu_{\underline{R_a}\{x_2,x_4,x_5\}}(x_1) = 0.0 \; ; \; \mu_{\underline{R_a}\{x_2,x_4,x_5\}}(x_2) = 0.0$$
$$\mu_{\underline{R_a}\{x_2,x_4,x_5\}}(x_3) = 0.301 \; ; \; \mu_{\underline{R_a}\{x_2,x_4,x_5\}}(x_4) = 0.0$$
$$\mu_{\underline{R_a}\{x_2,x_4,x_5\}}(x_5) = 0.0 \; ; \; \mu_{\underline{R_a}\{x_2,x_4,x_5\}}(x_6) = 0.0$$

Hence, the positive regions for each antigen are:

$$\mu_{POS_{R_a}}(Q)(x_1) = 0.0 \; ; \; \mu_{POS_{R_a}}(Q)(x_2) = 0.0$$
$$\mu_{POS_{R_a}}(Q)(x_3) = 0.301 \; ; \; \mu_{POS_{R_a}}(Q)(x_4) = 0.301$$
$$\mu_{POS_{R_a}}(Q)(x_5) = 0.0 \; ; \; \mu_{POS_{R_a}}(Q)(x_6) = 0.0$$

The resulting degree of dependency is, therefore:

$$\gamma'_{\{a\}}(Q) = \frac{\sum_{x \in U} \mu_{POS_{R_a(Q)}(x)}}{|U|} = \frac{0.602}{6} = 0.1003$$

Calculating the dependency degrees for the remaining features results in:

$$\gamma'_{\{b\}}(Q) = 0.3597 \; ; \; \gamma'_{\{c\}}(Q) = 0.4078$$

As feature c results in the largest increase in the fuzzy dependency degree, this feature is selected and added to the reduct candidate. The algorithm then evaluates the addition of all remaining features to this candidate. Fuzzy similarity relations are combined using Equation 4.3. This produces the following evaluations:

$$\gamma'_{\{a,b\}}(Q) = 0.5501 \; ; \; \gamma'_{\{b,c\}}(Q) = 1.0$$

Feature subset $\{b, c\}$ produces the maximum fuzzy dependency value for this data set, and the algorithm terminates. The data set can now be reduced to these features only.

For the signal categorization process and with respect to the signal categories ranking, FLA-DCA processes as follows:

The first attribute added to the fuzzy reduct, as it has the highest fuzzy dependency degree, is chosen to represent the SSs. The next attribute added to the fuzzy reduct is chosen to represent the PAMPs. The rest of the fuzzy reduct attributes are combined and the resulting value is assigned to the DSs. This is summarized in the following list:

- $SS = \{c\}$
- $PAMP = \{b\}$
- $DS = \emptyset$

4.4.4 Experimental Setup, Results and Discussion

Previously with FRST-DCA, we have shown that applying FRST instead of RST is more convenient for data preprocessing since the mentioned information loss is avoided. Despite of that the proposed FRST-DCA is a bit slow in terms of running time. In this Section, we will show that FLA-DCA can cover this FRST-DCA problem while generating satisfactory classification results.

Comparing the results in terms of number of selected features, from Table 4.2, we can notice that in most databases FLA-DCA is able to select fewer features than FRST-DCA. For some data sets, both algorithms select the same number of attributes. For instance, when applying FLA-DCA to the GC database the number of selected attributes is 11. However, when applying FRST-DCA to the same database the number of selected features is set to 17.

Table 4.2: Comparison Results of FRST-DCA and FLA-DCA

Database	♯ Attributes		Sensitivity (%)		Specificity (%)		Accuracy (%)		Time (s)	
	DCA		DCA		DCA		DCA		DCA	
	FRST	FLA	FRST	FLA	FRST	FLA	FRST	FLA	FRST	FLA
SN	16	10	94.60	95.49	95.88	96.90	95.19	96.15	95.91	9.41
Bio	13	9	84.90	86.79	86.79	92.45	85.84	89.62	47.29	8.47
CylB	7	5	98.00	97.00	98.71	98.39	98.43	97.85	118.41	18.96
Ch	4	4	99.28	99.34	98.69	98.75	98.99	99.06	1047.25	666.8
IONO	9	9	97.61	98.41	97.77	99.11	97.72	98.86	130.28	30.72
Sck	16	14	97.40	96.96	99.40	99.12	99.28	98.99	947.25	602.8
HC	7	6	97.22	96.75	98.68	98.02	97.82	97.28	184.56	23.84
GC	17	11	90.87	88.32	92.28	90.49	91.90	89.90	335.58	166.26
LR	5	5	90.00	90.00	91.89	91.89	91.22	91.22	0.64	0.61
RWW	4	3	99.63	99.18	99.56	99.37	99.61	99.23	1809.23	1265.1

From Table 4.2, we can notice that there is no huge difference in terms of classification accuracy of both FLA-DCA and FRST-DCA. In some data sets, both algorithms perform equally. In some others, FLA-DCA outperforms

FRST-DCA and vice-versa. For example, when applying the algorithms to the SN data set the classification accuracy of FLA-DCA is set to 96.15%. However, when applying FRST-DCA to the same database the accuracy is set to 95.19%. Concerning the CylB database, the FRST-DCA is outperforming FLA-DCA. As for the LR data set, both algorithms generate the same accuracy. Same remark is observed for the specificity and the sensitivity criteria.

Most importantly, when comparing the results in terms of running time we can notice that the time taken by our FLA-DCA to process is much less than the time taken by FRST-DCA to run. This is explained by the fact that FLA-DCA does not need to generate the membership functions of each data point like FRST-DCA. For example, when applying the algorithms to the Bio database, the amount of time taken by our FLA-DCA to process is 8.47(s) which is much less than the time taken by FRST-DCA which is set to 47.29(s). The major results of this work were published in (Chelly & Elouedi, 2013f).

Remarkably, FLA-DCA uses only the fuzzy lower approximation to select features. However, more uncertainty can be handled within the selected reduct set. This specific point was tackled by our next proposed fuzzy-rough algorithm.

4.5 FBR-DCA

In this Section, we present the FBR-DCA fuzzy-rough algorithm. FBR-DCA is based on the Fuzzy Boundary Region and fuzzy similarity relations.

4.5.1 FRST Concepts Based on the Fuzzy Boundary Region

The Transition to the Fuzzy Boundary Region

Most approaches to rough set feature selection and all approaches to fuzzy-rough feature selection use only the lower approximation or the fuzzy lower approximation for the evaluation of feature subsets (Jensen & Shen, 2002). The fuzzy lower approximation contains information regarding the extent of certainty of object membership to a given concept. However, the fuzzy upper approximation contains information regarding the degree of uncertainty of objects, and hence, this information can be used to discriminate between subsets. For example, two subsets may result in the same fuzzy lower approximation but one subset may produce a smaller fuzzy upper approximation. This subset will be more useful as there is less uncertainty concerning objects within the fuzzy boundary region. The fuzzy-rough boundary for a fuzzy concept X may be defined as:

$$\mu_{BND_{R_P}(X)}(x) = \mu_{\overline{R_P}X}(x) - \mu_{\underline{R_P}X}(x) \tag{4.7}$$

This subset which is defined as the difference between the fuzzy upper and the fuzzy lower approximations contains objects with less uncertainty.

Reduction Process

To search for the optimal subset of features the uncertainty for every concept has to be calculated. The uncertainty for a concept X using features in P can be calculated as follows:

$$U_P(X) = \frac{\sum_{x \in U} \mu_{BND_{R_P}(X)}(x)}{|U|} \tag{4.8}$$

This is the average extent to which objects belong to the fuzzy boundary region for the concept X. The total uncertainty degree for all concepts, given a feature subset P is defined as:

$$\gamma'_P(Q) = \frac{\sum_{X \in U/Q} U_P(X)}{|U/Q|} \tag{4.9}$$

The Fuzzy-Rough QuickReduct algorithm, defined previously, is modified and is based on this measure. According to the new Fuzzy-Rough QuickReduct algorithm, the task of the algorithm is to minimize the total uncertainty degree. When this reaches the minimum for the data set, a fuzzy-rough reduct has been found.

4.5.2 The FBR-DCA Feature Selection Process

For feature selection, FBR-DCA has to determine first of all the fuzzy boundary regions for both concepts; i.e., the two-class labels d_k. To do so, the fuzzy lower and the fuzzy upper approximations of each concept d_k for each feature c_i and for all objects x_j must be calculated. The fuzzy boundary region, the fuzzy lower and the fuzzy upper approximations are denoted by $\mu_{BND_{R_{c_i}}(d_k)}(x_j)$, $\mu_{\underline{R_{c_i}}(\{d_k\})}(x_j)$ and $\mu_{\overline{R_{c_i}}(\{d_k\})}(x_j)$, respectively. Once the fuzzy boundary regions are measured, FBR-DCA calculates the uncertainty degrees for each attribute c_i for each concept d_k, denoted by $U_{c_i}(d_k)$, as presented in the previous Section.

To find the fuzzy-rough reduct, FBR-DCA starts off with an empty set and moves to calculate the total uncertainty degrees for each feature c_i defined as $\gamma'_{c_i}(D)$. The attribute c_m having the smallest total uncertainty degree among all the calculated total uncertainty degrees of the remaining features is added to the empty fuzzy-rough reduct set. Once the first attribute c_m is selected, FBR-DCA adds in turn one attribute to the selected first attribute and computes the total uncertainty degrees of each obtained attributes couple $\gamma'_{\{c_m,c_i\}}(D)$. The algorithm chooses the couple having the smallest total uncertainty degree. The process of adding each time one attribute to the reduct continues until the total uncertainty degree of the obtained subset results in the minimal uncertainty for the data set.

4.5.3 The FBR-DCA Signal Categorization Process

For signal categorization, FBR-DCA selects the first attribute c_m having the smallest total uncertainty degree to form the SSs. Since FBR-DCA has already computed the total uncertainty degree of each attributes couple $\gamma'_{\{c_m,c_i\}}(D)$, FBR-DCA chooses the couple having the smallest total uncertainty degree. More precisely, FBR-DCA selects that second attribute c_r having the smallest $\gamma'_{\{c_m,c_r\}}(D)$ among the calculated $\gamma'_{\{c_m,c_i\}}(D)$. c_r is chosen to form the PAMPs. Finally, the rest of the fuzzy-rough reduct attributes are combined and affected to the DSs.

Example 4.3 *To illustrate the operation of our FBR-DCA, we use the same database given in Table 2.5. The fuzzy connectives chosen for this example are the Lukasiewicz t-norm* ($max(x + y - 1, 0)$) *and the Lukasiewicz fuzzy*

implicator ($min(1 - x + y, 1)$).

For feature selection and to determine the fuzzy boundary region, FBR-DCA calculates, first, the fuzzy lower and upper approximations of each concept for each feature. Considering feature a and concept $\{x_1, x_3, x_6\}$*, the process is as follows:*

$$\mu_{BND_{R_a}(\{x_1,x_3,x_6\})}(x) = \mu_{\overline{R_a}\{x_1,x_3,x_6\}}(x) - \mu_{\underline{R_a}\{x_1,x_3,x_6\}}(x)$$

For object x_4*, this is:*

$$\mu_{BND_{R_a}(\{x_1,x_3,x_6\})}(x_4) = sup_{y\in U} T(\mu_{R_a}(x_4, y), \mu_{\{x_1,x_3,x_6\}}(y)) - inf_{y\in U} I(\mu_{R_a}(x_4, y), \mu_{\{x_1,x_3,x_6\}}(y))$$

$$= 0.699 - 0.0 = 0.699$$

For the remaining antigens, this is:

$$\mu_{BND_{R_a}(\{x_1,x_3,x_6\})}(x_1) = 1.0 \; ; \; \mu_{BND_{R_a}(\{x_1,x_3,x_6\})}(x_2) = 1.0$$
$$\mu_{BND_{R_a}(\{x_1,x_3,x_6\})}(x_3) = 0.699 \; ; \; \mu_{BND_{R_a}(\{x_1,x_3,x_6\})}(x_4) = 0.699$$
$$\mu_{BND_{R_a}(\{x_1,x_3,x_6\})}(x_5) = 1.0 \; ; \; \mu_{BND_{R_a}(\{x_1,x_3,x_6\})}(x_6) = 1.0$$

Hence, the uncertainty for concept $\{x_1, x_3, x_6\}$ *is:*

$$U_a(\{x_1, x_3, x_6\}) = \frac{\sum_{x\in U} \mu_{BND_{R_a}(\{x_1,x_3,x_6\})}(x)}{|U|}$$

$$= \frac{1.0+1.0+0.699+0.699+1.0+1.0}{6} = 0.899$$

For concept $\{x_2, x_4, x_5\}$*, the uncertainty is:*

$$U_a(\{x_2, x_4, x_5\}) = \frac{\sum_{x\in U} \mu_{BND_{R_a}(\{x_2,x_4,x_5\})}(x)}{|U|}$$

$$= \frac{1.0+1.0+0.699+0.699+1.0+1.0}{6} = 0.899$$

From this, the total uncertainty for feature a is calculated as follows:

$$\gamma'_a(Q) = \frac{\sum_{X\in U/Q} U_a(X)}{|U/Q|}$$

$$= \frac{0.899+0.899}{2} = 0.899$$

The values of the total uncertainty for the remaining features are:

$$\gamma'_b(Q) = 0.640 \; ; \; \gamma'_c(Q) = 0.592$$

As feature c results in the smallest total uncertainty, it is chosen and added to the fuzzy reduct candidate. The algorithm then considers the addition of the remaining features to the subset:

$$\gamma'_{\{a,c\}}(Q) = 0.500 \; ; \; \gamma'_{\{b,c\}}(Q) = 0.0$$

The subset $\{b, c\}$ results in the minimal uncertainty for the data set, and the algorithm terminates.

For the signal categorization process and with respect to the signal categories ranking, FBR-DCA processes as follows:

The first attribute added to the fuzzy reduct, as it has the smallest uncertainty value, is chosen to represent the safe signals. The next attribute added to the fuzzy reduct is chosen to represent the PAMP signals. The rest of the fuzzy reduct attributes are combined and the resulting value is assigned to the danger signals. This is summarized in the following list:

- $SS = \{c\}$
- $PAMP = \{b\}$
- $DS = \emptyset$

Once the selected features are assigned to their suitable signal types, FRST-DCA, FLA-DCA and FBR-DCA calculate the values of each signal category using the same process as the standard DCA (Greensmith & Aickelin, 2006a). The output is, thus, a new information table which reflects the signal database. In fact, the universe U of the induced signal data set is $U = \{x'_1, x'_2, \ldots, x'_N\}$ a set of antigen identifiers and the conditional attribute set $C = \{SS, PAMP, DS\}$ contains the three signal types; i.e., SSs, PAMPs and DSs. Once data pre-processing is achieved, our fuzzy-rough DCAs process their next steps which are the Detection phase, the Context Assessment and the Classification phase as the standard DCA does and as described previously in the Chapter 1.

4.5.4 Experimental Setup, Results and Discussion

Let us remind that FLA-DCA uses only the fuzzy lower approximation to select features. However, more uncertainty can be handled within the FLA-DCA generated reduct set. This uncertainty is found within the fuzzy boundary region. Based on this idea, we try to show that our FBR-DCA based on the fuzzy boundary region can more discern the generated reduct set while generating satisfactory classification results. This is confirmed by the results presented in Table 4.3.

Comparing FBR-DCA and FLA-DCA in terms of number of selected features, we notice that FBR-DCA selects either fewer features than FLA-DCA or the same number of attributes. For instance, when applying both FBR-DCA

and FLA-DCA to the Bio database the number of selected attributes is the same and set to 9. However, when applying the algorithms to the SN data set we notice that FBR-DCA selects 9 features in comparison to FLA-DCA which selects 10 attributes. This is explained by the fact that FBR-DCA discerns more the generated reduct set since the algorithm uses both the fuzzy lower and the fuzzy upper approximations.

Furthermore, we notice that for both algorithms there is no significant difference in terms of classification accuracy. From Table 4.3, we can notice that in some data sets either FBR-DCA outperforms FLA-DCA or the opposite. In some databases, both algorithms performs equally in terms of classification accuracy. Same remark is observed for the specificity and the sensitivity criteria.

Table 4.3: Comparison Results of FLA-DCA and FBR-DCA

Database	♯ Attributes		Sensitivity (%)		Specificity (%)		Accuracy (%)		Time (s)	
	DCA		DCA		DCA		DCA		DCA	
	FLA	FBR	FLA	FBR	FLA	FBR	FLA	FBR	FLA	FBR
SN	10	9	95.49	97.29	96.90	97.93	96.15	97.59	9.41	14.87
Bio	9	9	86.79	86.79	92.45	92.45	89.62	89.62	8.47	13.58
CylB	5	5	97.00	97.00	98.39	98.39	97.85	97.85	18.96	29.06
Ch	4	4	99.34	99.40	98.75	98.82	99.06	99.12	666.8	714.95
IONO	9	9	98.41	98.41	99.11	99.11	98.86	98.86	30.72	41.12
Sck	14	14	96.96	96.53	99.12	99.09	98.99	98.93	602.8	704.95
HC	6	5	96.75	96.29	98.02	97.36	97.28	96.73	23.84	39.84
GC	11	10	89.05	87.95	90.77	90.35	89.90	89.70	166.26	196.9
LR	5	5	90.00	90.00	91.89	91.89	91.22	91.22	0.61	0.70
RWW	3	3	99.18	99.18	99.37	99.37	99.23	99.23	1265.1	1599.11

When comparing the results in terms of running time, we can notice that the time taken by our FBR-DCA to process is a bit longer than the time taken by FLA-DCA. This is explained by the fact that FBR-DCA calculates both the fuzzy lower and the fuzzy upper approximations. In contrast, FLA-DCA needs only to calculate the fuzzy lower approximation. For example, when applying the algorithms to the Bio database the amount of time taken by our FBR-DCA to process is 13.58(s) which is a bit longer than the time taken by FLA-DCA which is set to 8.47(s). This work was published in (Chelly & Elouedi, 2013b).

4.6 Comparison of the Proposed Fuzzy-Rough DCAs

In this Section, we will try to make a sort of a summary of the proposed fuzzy rough DCAs. We will compare the algorithms to each other while highlighting their positive aspects and their limitations.

4.6.1 Experimental Setup

Up to now, we have presented three different fuzzy-rough DCAs where each one is based on a different fuzzy-rough data pre-processing procedure. In this Section, we will compare the developed algorithms namely FRST-DCA, FLA-DCA and FBR-DCA. We will focus mainly on these algorithms as we have already shown in (Chelly & Elouedi, 2013c, 2013f, 2013b) that the proposed fuzzy-rough algorithms outperform the rough DCAs on the whole; i.e., in

terms of both feature selection and classification quality.

Let us remind that FRST-DCA uses a pre-categorization step that generates associated fuzzy sets for a data set. For both FLA-DCA and FBR-DCA methods, the Lukasiewicz fuzzy connectives are used with a fuzzy similarity defined in Equation 4.4. After feature selection, the data sets are reduced according to the discovered fuzzy reducts. These reduced data sets are then classified using the dendritic cell classifier. The comparison between our fuzzy-rough algorithms is presented in Table 4.4.

4.6.2 Discussion of the Obtained Results

From Table 4.4, we can notice that FLA-DCA and FBR-DCA can find smaller subsets than FRST-DCA in general. For instance, when applying FLA-DCA and FBR-DCA to the GC database the number of selected features is set to 11 and 10, respectively. Nevertheless, when applying FRST-DCA to the same data set the number of selected features is set to 17. FBR-DCA finds smaller or equally sized subsets than the lower approximation method, FLA-DCA. This is to be expected, as FBR-DCA includes the fuzzy upper approximation information in addition to that of the fuzzy lower approximation. For instance, when applying FBR-DCA to the SN data set the number of selected feature is set to 9. However, when applying FLA-DCA to the same data set the number of selected features is set to 10. When applying both algorithms to the CylB data set the number of selected features is the same and set to 5 attributes.

From Table 4.4, we can also notice that FLA-DCA and FBR-DCA are quicker than FRST-DCA in terms of running time. This is explained by the fact that FRST-DCA based on fuzzy-rough sets has to generate, first, the fuzzy sets as well as the membership functions from the used database and, second, has to determine the fuzzy equivalence classes for each conditional attribute set. However, the complexity related to these calculations becomes prohibitively high for large feature subsets. On the other hand, both FLA-DCA and FBR-DCA avoid these extra calculations as they are both based on the use of fuzzy similarity relations.

Indeed, we have to point out that FBR-DCA needs more time to process than FLA-DCA. This is because FBR-DCA has to calculated, for each evaluation, both the fuzzy lower and upper approximations. Hence, the calculation of the fuzzy boundary region is more costly than that of the fuzzy lower approximation alone. For example, when applying FRST-DCA to the SN data set the time taken by the algorithm to process is 95.91(s) which is longer than the time needed by both FLA-DCA and FBR-DCA which is set to 9.41(s) and 14.87(s), respectively. For this example, we can also see that the time needed by FBR-DCA is longer than the time needed by FLA-DCA to process.

Furthermore, from Table 4.4 we can notice that the proposed fuzzy-rough DCAs perform similarly with classification accuracy improving or remaining the same for most data sets. FRST-DCA performs equally well. However, this is at the cost of extra features and extra time required to find the fuzzy reducts. For instance, when applying FRST-DCA, FLA-DCA and FBR-DCA to the Sck data set the classification accuracy is set to 99.28%, 98.99% and to 98.93%, respectively.

Table 4.4: Comparison Results of FRST-DCA, FLA-DCA and FBR-DCA

Database	♯ Attributes			Sensitivity (%)			Specificity (%)			Accuracy (%)			Time (s)		
	DCA			DCA			DCA			DCA			DCA		
	FRST	FLA	FBR	FRST	FLA	FBR	FRST	FLA	FBR	FRST	FLA	FBR	FRST	FLA	FBR
SN	16	10	9	94.60	95.49	97.29	95.88	96.90	97.93	95.19	96.15	97.59	95.91	9.41	14.87
Bio	13	9	9	84.90	86.79	86.79	86.79	92.45	92.45	85.84	89.62	89.62	47.29	8.47	13.58
CylB	7	5	5	98.00	97.00	97.00	98.71	98.39	98.39	98.43	97.85	97.85	118.41	18.96	29.06
Ch	4	4	4	99.28	99.34	99.40	98.69	98.75	98.82	98.99	99.06	99.12	1047.25	666.8	714.95
IONO	11	9	9	97.61	98.86	98.41	97.77	98.41	99.11	97.72	98.86	98.86	130.28	30.72	41.12
Sck	16	14	14	97.40	96.96	96.53	99.40	99.12	99.09	99.28	98.99	98.93	947.25	602.8	704.95
HC	7	6	5	97.22	96.75	96.29	98.68	98.02	97.36	97.82	97.28	96.73	184.56	23.84	39.84
GC	17	11	10	89.05	88.32	87.95	90.77	90.49	90.35	90.30	89.90	89.70	335.58	166.26	196.9
LR	5	5	5	90.00	90.00	90.00	91.89	91.89	91.89	91.22	91.22	91.22	0.64	0.61	0.70
RWW	4	3	3	99.63	99.18	99.18	99.56	99.37	99.37	99.61	99.23	99.23	1809.23	1265.1	1599.11

4.6.3 Comparison with the State-of-the-art Classifiers

In this Section, we have compared the performance of our proposed fuzzy-rough DCAs to other classifiers including the Support Vector Machine (SVM), Artificial Neural Network (ANN) and the Decision Tree (DTree). As we are focusing on the classification accuracy of the algorithms, the comparison made is in terms of the average of accuracies on the used databases. The parameters of SVM, ANN and DTree are set to the most adequate parameters to these algorithms using the Weka software.

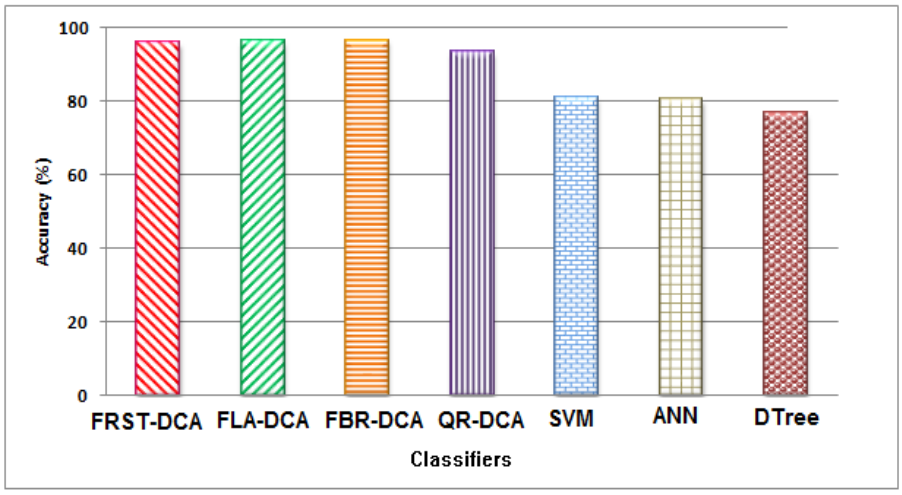

Figure 4.1: Comparison of Classifiers Average Accuracies

Figure 4.1 shows that our FRST-DCA, FLA-DCA and FBR-DCA have nearly the same classification performance. These algorithms outperform the crisp rough DCA; namely QR-DCA. This proves that the mentioned information is reduced as attribute values are kept unchanged; i.e., no quantization is performed on the original databases. Figure 4.1, also, shows that our fuzzy-rough DCAs outperform the rest of the classifiers namely SVM, ANN and DTree in terms of the overall classification accuracy.

4.7 Conclusion

Throughout this Chapter, we focused our work on using fuzzy rough set theory for the DCA data pre-processing phase. We have developed three fuzzy-rough DCAs based on different methodologies for the feature selection and signal categorization processes. These methods were compared and the appropriate observations were noted.

Up to now, we have made a deep investigation of the DCA data pre-processing phase. We have proposed adequate solutions to build a solid data pre-processing module. While focusing on the rest DCA algorithmic steps, more investigations have to de done. To begin with and in the next Chapter, we will focus on solving the DCA limitation as it is sensitive to the input class data order.

Chapter 5

An Automated Fuzzy Dendritic Cell Classification Method

5.1 Introduction

In this Chapter, we will investigate and try to solve another limitation of the DCA which is its sensitivity to the input class data order. As previously mentioned, the standard DCA had to be applied to ordered data sets where all class 1 are followed by all class 2. This is to ensure high and satisfactory classification results. While investigating this DCA limitation, we have noticed that this problem could be linked to the DCA context assessment phase. Therefore, we propose in this Chapter to develop an automated fuzzy dendritic cell algorithm characterized by its stability as a binary classifier.

This Chapter is structured as follows: In Section 5.2, we present the motivation behind the development of our proposed solution. In Section 5.3, fuzzy clustering techniques which are needed to build our new fuzzy DCA are detailed. Section 5.4 describes our new developed algorithm. Experiments are outlined in Section 5.5 and the final Section includes the discussion of the obtained results.

5.2 Problem Statement

A study focusing on the DCA behavior stated that the algorithm gives satisfactory and interesting classification results when it is, only, applied to databases having ordered classes (Greensmith & Aickelin, 2008b). In this Chapter, we try to study carefully the DCA algorithmic phases while trying to figure out the reasons of such a restriction.

First of all, we hypothesize that the first cause of such a sensitivity to the input class data order could be related to the DCA context assessment stage. More precisely, we hypothesize that such a restriction is the result of an environment characterized by a crisp separation between normality (semi-mature context) and abnormality (mature

context). In fact, if the difference value between these two DCs' contexts is small then the context of the DC will be hard to be defined. Thus, it could change the decision of the context affectation. Not considering this case, has a negative effect on classification accuracy when the class of data instances changes over time.

Still with the same hypothesis, we note that there is an imprecision in the used terms such as "semi-mature" and "mature". This critical point was partially handled in our first work named the Fuzzy Dendritic Cell Method (FDCM) (Chelly & Elouedi, 2010) by hybridizing the DCA with the fuzzy set theory component. Nevertheless, FDCM was still sensitive to the class data order. So, we had to resolve the shortcoming of the latter work.

While exploring FDCM, we noticed that all the algorithm input parameters were extracted automatically from data except the extents and midpoints of the fuzzy membership functions. Therefore, it seems crucial to develop an overall automated fuzzy DCA based on generating automatically the parameters of the algorithm. This can be achieved via the use of a fuzzy clustering technique. We believe that this automatic process can avoid false and uncertain values given by the user who can be a non-expert in the field. We, also, hypothesize that the effectiveness of our proposed solution may depend on the choice of our system's parameters. Thus, we should carefully select the right fuzzy clustering technique that will be hybridized with our proposed solution. This will be achieved through a study of several fuzzy clustering techniques.

5.3 Overview of the Fuzzy Clustering Techniques

Clustering of numerical data forms the basis of many classification and system modeling algorithms. The purpose of clustering is to identify natural groupings of data from a large data set to produce a concise representation of a system's behavior. As one of important clustering approaches, fuzzy clustering techniques allow to obtain the membership degree of samples belonging to each class and express the intermediate specificity of their memberships (Bezdek, 1981).

There are various fuzzy clustering algorithms proposed in literature and among the most applied and popular ones, we cite the following: Fuzzy c-means (FCM) (Bezdek, 1981), Gustafson-Kessel (GK) (Gustafson & Kessel, 1979) and Shell clustering (Hoppner, Klawonn, Kruse, & Runkler, 1999). These main algorithms have shown their potential to provide encouraging and competitive results among other fuzzy clustering techniques. Hence, we will briefly discuss them in what follows.

5.3.1 The Fuzzy c-means Algorithm

Fuzzy c-means (FCM) is a data clustering technique wherein each data point belongs to a cluster with some degree that is specified by a membership grade. This technique was, first, introduced by Dunn in (Dunn, 1973) and improved by Bezdek (Bezdek, 1981). FCM is based on the minimization of an objective function called c-means functional which is the following:

$$J_{FCM}(P, U, X, c, m) = \sum_{i=1}^{c} \sum_{k=1}^{N} (\mu_{ik})^m d_{ik}^2(x_k, p_i) \tag{5.1}$$

subject to the constraint

$$\sum_{i=1}^{c} \mu_{ik} = 1 \forall k \in \{1, \ldots, N\} \tag{5.2}$$

where P and U are the variables whose optimal values are being sought. X, c and m are input parameters of J_{FCM}, where: c denotes the number of fuzzy clusters assumed to exist in X; $m \geq 1$ is the fuzzification exponent that controls how fuzzy the result will be; μ_{ik} describes the degree of membership feature vector x_k with the cluster represented by p_i. $U = [\mu_{ik}]$ is the $c * N$ fuzzy partition matrix satisfying the constraint stated in Equation 5.2; N is the total number of feature vectors and d_{ik} is the distance between feature vector x_k and prototype p_i.

The FCM algorithm can, only, find clusters with the same shape and size because the distance used which is often the Euclidean is not adaptive. FCM is a sequence of iterations through the equations below, which are referred to as the update equations.

$$p_i* = \frac{\sum_{k=1}^{N} \mu_{ik}^m x_k}{\sum_{k=1}^{N} \mu_{ik}^m} \tag{5.3}$$

and

$$\mu_{ik}* = \frac{1}{\sum_{j=1}^{c} \left(\frac{d_{ik}^2}{d_{jk}^2}\right)^{(1/m-1)}} \tag{5.4}$$

When the iteration converges, a fuzzy c-partition matrix and the pattern prototypes are obtained. The algorithmic steps for FCM are the following (Bezdek, 1981):

Algorithm 5.1 The Fuzzy c-means Algorithm

1. Initialize the fuzzy membership matrix U;
2. Calculate c fuzzy cluster centers P using Equation 5.3;
3. Update memberships U using Equation 5.4;
4. Repeat step (2) and (3) until the minimum J_{FCM} value is achieved;

5.3.2 Gustafson-Kessel Algorithm

The Gustafson-Kessel (GK) algorithm is an extension of FCM. GK replaces the Euclidean distance used by FCM by the transformed Euclidean distance defined by Equation 5.5. This is to detect clusters with different shapes. Contrary to FCM, GK gives the clusters with the different sizes different dimensions (clusters are ellipsoids). In this case every cluster has its own norm matrix A_i.

$$d_{ik}^2(x_k, p_i) = \|x_k - p_i\|_{A_i}^2 = (x_k - p_i)^T A_i (x_k - p_i) \tag{5.5}$$

Thus, the objective function becomes the following:

$$J_{GK}(P, U, A, X, c, m) = \sum_{k=1}^{N} \sum_{i=1}^{c} (\mu_{ik})^m ((x_k - p_i)^T A_i (x_k - p_i)) \tag{5.6}$$

However, the objective function cannot be minimized directly because J_{GK} depends on A_i linearly. It means that J_{GK} can be made as small as the considered positive definite matrices A_i. Therefore A_i is constrained by:

$$||A_i|| = p_i = constant \tag{5.7}$$

The GK algorithmic steps are the following (Gustafson & Kessel, 1979):

Algorithm 5.2 The Gustafson-Kessel Algorithm

1. Initialize all p_i; Initialize all A_i;
2. Calculate fuzzy partition U using Equation 5.4;
3. Update prototypes P using Equation 5.3;
4. Calculate A's by:

$$A_i^{-1} = \frac{1}{p_i|C_i|}^{1/p} C_i \tag{5.8}$$

where C_i, the fuzzy convariance matrix, is given by:

$$C_i = \frac{\sum_{k=1}^{N} \mu_{ik}^m (x_k - p_i)(x_k - p_i)^T}{\sum_{k=1}^{N} \mu_{ik}^m} \tag{5.9}$$

5. If termination condition not achieved, return to step (2).

5.3.3 Shell Clustering Algorithm

The fuzzy clustering algorithms discussed up to now search for clusters that lie in linear subspaces. Besides, it is also possible to detect clusters that lie in nonlinear subspaces. These clusters can be detected using shell clustering algorithms. There are several variants of shell clustering algorithms such as the fuzzy c-shells algorithm, the fuzzy c-spherical shells algorithm, etc. A full review of them can be found in (Krishnapuram, Nasraoui, & Frigui, 1995). The main innovation behind every shell clustering algorithm is the distance measure it uses. Fuzzy shell clustering algorithms can detect several cluster's shapes such as ellipses, quadrics, polygons, ellipsoids, etc. One of the most used shell-clustering prototypes is the Adaptive Fuzzy C-Shell algorithm (AFCS) (Dave, 1993) which searches for ellipses. AFCS uses a hyperellipse for the ith prototype:

$$E_i(x_k, p_i) = \{||x_k - p_i||_{A_i}^2 = 1\} \tag{5.10}$$

where A_i is a positive definite symmetric matrix which determines the major and minor axes lengths as well as the orientation of the hyperellipse and p_i is its center. Consider the distance d_{ik}^2 defined by:

$$d_{ik}^2(x_k, (p_i, A)) = (||x_k - p_i||_{A_i} - 1)^2 = (\sqrt{(x_k - p_i)^T A (x_k - p_i)} - 1)^2 \tag{5.11}$$

Its minimization yields to the following equation for updating the parameters $B_i = (p_i, A_i)$ of ellipse E_i:

$$\sum_{k=1}^{N} \mu_{ik}^m \left(\frac{d_{ik}}{||x_k - p_i||A_i} \right) (x_k - p_i)(x_k - p_i)^T = 0 \tag{5.12}$$

This system must be solved numerically at each half step in the iteration. The AFCS algorithmic steps are the following (Hoppner et al., 1999):

Algorithm 5.3 The Fuzzy Adaptive C-Shell Algorithm

1. Initialize all p_i; Initialize all A_i;
2. Calculate fuzzy partition U;
3. Update prototypes B_i using Equation 5.12;
4. Calculate d using Equation 5.11;
5. Repeat step (2), (3) and (4) until the minimum B_i value is achieved;

The output of all these fuzzy clustering methods is a list of cluster centers and several membership grades for each data point. Let us remind that our objective is to generate automatically the parameters of our proposed fuzzy DCA solution. Thus, we will use and compare the information returned by these fuzzy clustering algorithms to build the fuzzy inference system for our fuzzy DCA. This will be achieved by creating membership functions to represent the fuzzy qualities of each cluster and consequently to determine automatically the extents and midpoints of the serached membership functions.

5.4 The Fuzzy Classification Dendritic Cell Method

In this Section, we propose a new fuzzy process consisting on the definition of a new version of the standard DCA, named the Fuzzy Classification Dendritic Cell Method (FCDCM). FCDCM takes into account the fact of alleviating the crisp assessment task by the use of fuzzy set theory. FCDCM is, also, based on an automatic generation of the algorithm parameters, specifically, the determination of the extent and midpoints of the membership functions. This will be achieved by the use of an appropriate fuzzy clustering technique.

Figure 5.1 shows a global view of our proposed FCDCM. As presented in Figure 5.1, FCDCM uses the same algorithmic steps as the standard DCA which are the pre-processing phase, the detection phase and the classification phase. However, instead of using the standard DCA context assessment phase, FCDCM applies a fuzzy process which incorporates a fuzzy context assessment step.

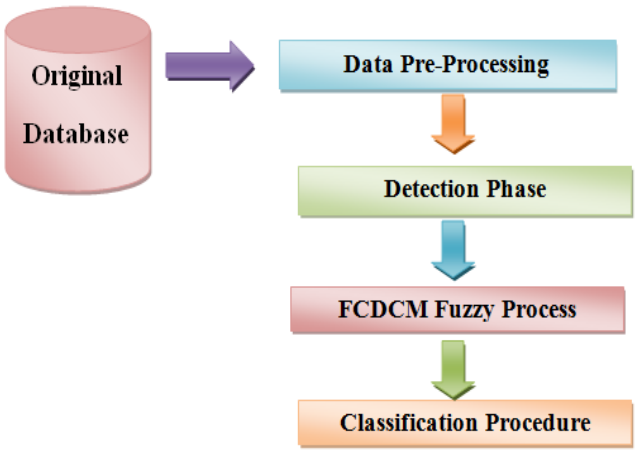

Figure 5.1: Global View of FCDCM

Implementing the fuzzy part of FCDCM falls into the definition of five main phases namely; the definition of the fuzzy system Inputs-Output variables, the definition of the linguistic variables, the construction of the fuzzy membership functions, the construction of the rule base and, finally, the fuzzy context assessment phase. Figure 5.2 shows the fuzzy procedure of FCDCM.

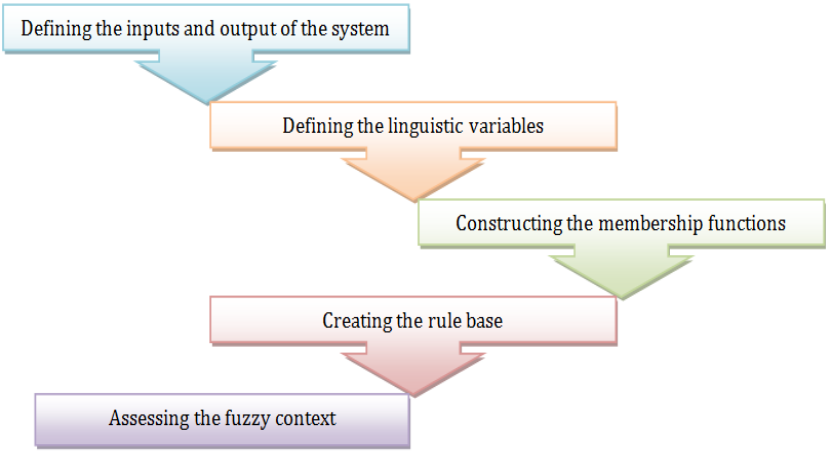

Figure 5.2: Five Steps of the Fuzzy Process

5.4.1 FCDCM Data Pre-Processing Phase

The FCDCM data pre-processing phase is performed the same as with the DCA. Technically, this phase can rely on experts knowledge which are supposed to select a subset of attributes from the initial data set and categorize each feature either as a PAMP signal, as a danger signal or as a safe signal. As stated in the first Chapter of this dissertation, a second option for data pre-processing can be the use of the principal component analysis.

5.4.2 FCDCM Detection Phase

The FCDCM detection phase includes the signal value derivation phase that is broadly divided into two processes. The first process consists of the calculation of the PAMPs and SS values and the calculation of the DS values. The second process consists of the calculation of the interim output signals detailed in Chapter 1.

Process for Calculating PAMPs and SSs

As stated in the signal definitions, in Chapter 1, both PAMPs and safe signals are positive indicators of an anomalous and normal signal. To achieve this, one attribute is used to form both PAMPs and SSs. In this way, we contrive the scenario where the algorithm is given a context of either PAMPs or safe signals. Using one attribute for these two signals requires a threshold level to be set: values greater than this can be classified as a safe signal, while values

below this level would be used as a PAMP signal. The exact procedure for calculating safe and PAMP signals is given in the following itemized list:

1. Select a suitable attribute;
2. Calculate the median of all the selected attributes values across both classes of data;
3. For each attribute value determine if it is a PAMP or a safe signal. If the attribute value is greater than the median then this value is used to form a safe signal. The absolute distance from the mean is calculated and attached to the safe signal value and the PAMP signal value takes 0 (and vise versa).

Process for Calculating DSs

A similar process is used to calculate the values for the danger signals. As stated previously, the danger signal is less than certain to be anomalous. This is interpreted as a combination of several attributes resulting in a value that may be used as anomalous. To do so, the mean value for each attribute set is required from the normal class alone. This process is explained in the following list:

1. Compute mean values using the values of class 1 for each attribute, not including class 2 as with the PAMP and safe signals;
2. Take each attribute value in turn and calculate the absolute distance between the attribute values and the calculated means;
3. Use the calculated distance values in a further calculation to form the single value for the DS. This value is the mean value of the absolute distances calculated, with the derivation shown in Equation 5.13:

$$DS = \frac{\sum \text{absolute distances}}{\text{number of attributes}} \tag{5.13}$$

4. Repeat this process for all entries of the selected attributes.

Once these signals are generated, the result is a set of feature vectors ready to be presented to the system.

5.4.3 FCDCM Fuzzy Process

As presented in Figure 5.2, the fuzzy process of FCDCM is composed of five main sub-steps. Each step is described in what follows.

Defining the System Inputs-Output Variables

Our objective is to smooth the crisp separation between the semi-mature context and the mature context using fuzzy concepts. This is because there is no clear boundary between these two DC contexts. To do so, we describe each context of each object using linguistic variables. Two inputs, one for each context, and one output are defined. The semi-mature context and the mature context, denoted respectively C_s and C_m, are considered as the input variables

of FCDCM. The final state "maturity" of a DC, S_{mat}, is chosen as the output variable. All the system's inputs and output are defined using fuzzy set theory.

$$C_s = \{\mu_{C_s}(c_{s_j})/c_{s_j} \in X_{C_s}\} \tag{5.14}$$

$$C_m = \{\mu_{C_m}(c_{m_j})/c_{m_j} \in X_{C_m}\} \tag{5.15}$$

$$S_{mat} = \{S_{mat}(s_{mat_j})/s_{mat_j} \in X_{S_{mat}}\} \tag{5.16}$$

where c_{s_j}, c_{m_j} and s_{mat_j} are, respectively, the elements of the discrete universe of discourse X_{C_s}, X_{C_m} and $X_{S_{mat}}$. μ_{C_s}, μ_{C_m} and $\mu_{S_{mat}}$ are, respectively, the corresponding membership functions.

Defining the Term Sets

The term set $T(S_{mat})$ interpreting S_{mat} which is a linguistic variable that constitutes the final state of maturity of a DC, could be:

$$T(S_{mat}) = \{Semi - mature, Mature\} \tag{5.17}$$

Each term in $T(S_{mat})$ is characterized by a fuzzy subset in a universe of discourse $X_{S_{mat}}$. Semi-mature might be interpreted as an object (data instance) collected under safe circumstances, reflecting a normal behavior and Mature as an object collected under dangerous circumstances, reflecting an anomalous behavior. Similarly, the input variables C_s and C_m are interpreted as linguistic variables with:

$$T(Q) = \{Low, Medium, High\} \tag{5.18}$$

where Q = C_s and C_m, respectively.

Membership Functions Construction

Each linguistic variable of our system is represented using triangular membership functions. In fact, this specific shape has the advantage of simplicity while giving minimum error and requiring minimum computation time. The choice of the triangular membership functions for our FCDCM approach is, also, based on the aim to ease the computational burden including the simplification of the defuzzification process.

In order to specify the range of each linguistic variable, we have run the DCA and we have recorded both semi-mature and mature values which reflect the (Semi) and (Mat) outputs generated by the DCA. Then, we picked up the minimum and maximum values of each of the two generated values to fix the borders of the range. The range of the output variable is determined as follows:

$$min(range(S_{mat})) = min(min(range[C_m]), min(range[C_s])) \tag{5.19}$$

$$max(range(S_{mat})) = max(max(range[C_m]), max(range[C_s])) \tag{5.20}$$

Once the range of the output S_{mat} is determined, it seems important now to fix the extents and midpoints of each membership function. We aim at generating these parameters automatically by the system to avoid any negative influence on the results if they will be given by any ordinary user. This is because the user may not be an expert

in the domain. As for experts, each one could propose his/her own parameters. As a result, we will have different values, which could give rise to some confusions about which ones are the most appropriate to our system. Hence, our choice can be focused on the use of an appropriate fuzzy clustering algorithm among the most applied and well known fuzzy techniques detailed in Section 3.

As previously stated, fuzzy clustering techniques are able to describe varying forms or shapes of clusters which has an effect on the determination of the coordinates of each data point on each cluster. These values which present the searched parameters for our system and which are generated differently from one fuzzy clustering algorithm to another influence the final results. In general, we can say that the more flexible one algorithm is in detecting different clusters forms, the more susceptible is this algorithm towards local optima. Hence, it is important to determine the right fuzzy clustering technique which will be capable of generating the most convenient extents and midpoints for our FCDCM algorithm. We, therefore, present FCDCM on different versions where each one applies a different fuzzy clustering method starting with a simple cluster form - the Euclidian distance - and changing regularly to more complex forms. The first FCDCM version, named $FCDCM_{FCM}$, uses the FCM algorithm. The second FCDCM version, $FCDCM_{GK}$, uses the GK algorithm and the last FCDCM version, $FCDCM_{AFCS}$, uses the AFCS algorithm.

To the recorded list of (Mat) and (Semi) values, we apply each time a different fuzzy clustering technique among the mentioned ones. This is to build a fuzzy inference system by creating membership functions to represent the fuzzy qualities of each cluster. Thus, we assume that each cluster reflects a membership function and the number of clusters is relative to the number of the membership functions of each variable (inputs and output). Our objective is to find which fuzzy clustering technique is the most appropriate for our FCDCM.

The output of this FCDCM third phase is a list of cluster centers and several membership grades μ_{ik} generated by the fuzzy clustering algorithms for each antigen. Thus, the extents and midpoints of the membership functions are automatically determined.

The Rule Base Construction

A knowledge base comprising rules is built to support the fuzzy inference. The different rules of the fuzzy system are extracted from the information listed below; reflecting the effect of each input signal on the state of a DC.

1. Safe signals: an increase in value is a probable indicator of normality. High values of the SSs can cancel out the effects of both PAMPs and DSs.
2. Danger signals: an increase in value is a probable indicator of damage but there is less certainty than with a PAMP signal.
3. PAMPs: an increase in value is a definite indicator of anomaly.
4. Inflammation: has the effect of amplifying the other three categories of input signals but is not sufficient to cause any effect on DCs when used in isolation.

From the list above, we can generate the set of rules presented in Table 5.1 where all the mentioned signals are taken into account implicitly in the fuzzy system.

Table 5.1: The Fuzzy Rule Base

Rule Number	Description
Rule(1):	If (C_m is Low) and (C_s is Low) then (S_{mat} is Mature)
Rule(2):	If (C_m is Low) and (C_s is Medium) then (S_{mat} is Semi-mature)
Rule(3):	If (C_m is Low) and (C_s is High) then (S_{mat} is Semi-mature)
Rule(4):	If (C_m is Medium) and (C_s is Low) then (S_{mat} is Mature)
Rule(5):	If (C_m is Medium) and (C_s is Medium) then (S_{mat} is Semi-mature)
Rule(6):	If (C_m is Medium) and (C_s is High) then (S_{mat} is Semi-mature)
Rule(7):	If (C_m is High) and (C_s is Low) then (S_{mat} is Mature)
Rule(8):	If (C_m is High) and (C_s is Medium) then (S_{mat} is Mature)
Rule(9):	If (C_m is High) and (C_s is High) then (S_{mat} is Mature)

Let us consider Rule (2) as an example: if the C_m input is set to its first membership function "Low" and the second input C_s to its second membership function "Medium", then the "Semi-mature" context of the output S_{mat} is assigned. This could be explained by the effect of the high values of SSs (which lead to the semi-mature context) that cancel out the effects of both PAMPs and DSs (which lead to the mature context). The same reasoning is affected to the rest of the rules.

We have, also, to mention the case when we have the same term set for both inputs C_s and C_m. In this case, we have not sufficient information about where the DC is going to migrate. In other words, we will not be sure about the final context of the S_{mat} output if it will be set to the "Mature" value or to the "Semi-mature" value. Thus, we have introduced the rule's weighting fuzzy concept. If we have the same term set for both inputs then the corresponding rule weight is set to 50%. This means that our fuzzy system will process that rule with 50% assigned to the "Semi-mature" term set and 50% assigned to the "Mature" term set. Consequently, we could give the same chances for the S_{mat} output to be processed with its two possible term sets.

Once the rule base is constructed, the fuzzy inference properties have to be selected. To do so, our FCDCM is based on the *"Min-Max"* inference method. Furthermore, the FCDCM is based on the *"centroid defuzzification"* method which is the most popular; presenting the center of gravity of the membership function. This method is the greediest in calculation but gives the most precise results which justifies our choice. The centroid defuzzification method is given by Equation 5.21.

$$\sum_{i=1}^{N}(\mu_{(i)} * output(i))/\sum_{i=1}^{N}(\mu_{(i)}) \tag{5.21}$$

where $\mu_{(i)}$ is the truth value of the result membership function for rule i, output(i) is the value for rule i where the result membership function is maximum over the output variable fuzzy set range and N is the number of rules.

The Fuzzy Context Assessment

Once the inputs are fuzzified and the output (centroid value) is generated, the cell context has to be fixed by comparing the output value to the middle of the S_{mat} range. This could be explained as follows:

- If the centroid value generated is greater than the middle of the output range then the surface of the "Maturity" output can be described as skewed to the right (positive asymmetry). Therefore, the final state/context of the object is "Mature". Thus, it indicates that the collected antigen may be anomalous.

- If the middle of the output range is greater than the centroid value generated then the surface of the "Maturity" output can be described as skewed to the left (negative asymmetry). Hence, the final state/context of the object is "Semi-Mature". This means that the antigen collected is likely to be normal.

5.4.4 FCDCM Classification Phase

The classification phase is the same procedure explained in Chapter 1. However, we will explain further how the anomaly threshold could be generated automatically from the data. To perform anomaly detection, a threshold must be applied to the MCAVs. The distribution of data between class one and class two is used and reflects the potential danger rate. The calculation displayed in Equation 5.22 shows this process.

$$at = \frac{an}{tn} \tag{5.22}$$

In this equation, *an* is the number of anomalous data items, *tn* is the total number of data items and *at* is the derived anomaly threshold. If the MCAV is greater than the anomaly threshold then the antigen is classified as anomalous else it is classified as normal.

5.4.5 Worked Example

In this Section, we give an example illustrating the application of FCDCM. The algorithm will be applied to the database presented in Table 5.2.

Table 5.2: Initial Data Set

Client	Age	Income	Number of credit cards	Duration of the loan	Credit
Client1	36	1300	3	8	yes
Client2	32	900	2	13	yes
Client3	33	1100	4	9	yes
Client4	20	600	1	20	no
Client5	24	650	1	30	no
Client6	30	1000	3	10	no

Table 5.2 is composed of 6 clients which are seen as antigens to be classified. Each antigen is characterized by 4 attributes which are the age, the income, the number of credit cards and the duration of the loan. FCDCM will classify each antigen to its specific binary class: "yes" or "no". The class of each antigen is defined by the attribute "Credit".

The Attribute Selection and Categorization Phase

As mentioned previously, the attribute selection and categorization phase can be based on the experts knowledge. Thus, we assume that experts select the attributes Age, Income and Duration of the loan for the FCDCM functioning. We, also, assume that they select the attribute Income to derive the PAMPs and the safe signals and the rest of the attributes (Age and Duration of the loan) to calculate the DS values. Each data item is mapped as an antigen. Hence, Table 5.3 represents the reduced data set.

Table 5.3: Reduced Data Set

Antigen (Client)	Age	Income	Duration of the loan
Ag1	36	1300	8
Ag2	32	900	13
Ag3	33	1100	9
Ag4	20	600	20
Ag5	24	650	30
Ag6	30	1000	10

The Signal Values Derivation Phase

Process for Calculating PAMP and Safe Signals The attribute Income is chosen to derive the PAMPs and safe signals and its median value is set to 950. For each attribute value, we determine if it is a PAMP or a safe signal. For instance, for Ag1 the first attribute value 1300 is higher than 950 then the resultant signals are PAMPs of value (1300-950=350) and SSs value of 0. For Ag4, the attribute value 650 is lower than 950 then the resultant signals are a SS of value (950-650=300) and a PAMP value of 0. The same process is used to calculate the values for the rest of the instances.

Process for Calculating DS In order to calculate the DS, first, the mean values are calculated across the values of class 1 for each attribute chosen. The two attributes selected for this experiment are:

* Age, mean = 33.6667
* Duration of the loan, mean = 10

Then, we take each attribute value in turn and calculate the absolute distance between the attribute values and the means shown in Table 5.4:

Table 5.4: Process of Calculating the Absolute Distance

	Age	Duration of the loan
attribute set (Ag1)	36	8
means	33.6667	10
absolute distance	2.3333	2

The calculated distance values are used in a further calculation to form the single value for the DS. This value is the mean value of the absolute distances calculated in the block above, with the derivation shown in Equation 5.13:

$$DS = \frac{2.3333 + 2}{2} = 2.1667$$

This process is repeated for all entries of the selected attributes. Following the generation of the signals, the result is a set of feature vectors shown in Table 5.5.

Table 5.5: Signal Feature Vectors

Antigen (Client)	SS	PAMP	DS
Ag1	0	350	2.1667
Ag2	50	0	2.3333
Ag3	0	150	0.8333
Ag4	350	0	11.8333
Ag5	300	0	14.8333
Ag6	0	50	1.8333

After the derivation of the different signals, we process by calculating the three output signals CSM, smDC and mDC for each object as detailed in Chapter 1.

The Fuzzy Process

Assume that the values of the two inputs (Semi-mature and Mature) are the following for the first object: x (mDC) = 8; and y (smDC) = 9.

Inputs and Output Descriptions Each input is represented by three triangular membership functions while the output is represented by two triangular membership functions. We assume that after the generation of all the smDC and mDC, the min and max values are set to 0 and 25. Thus, the range of the output variable is set to [0 25]. The extents of the different membership functions are parameters generated automatically by one of the fuzzy clustering algorithms discussed previously in Section 3. We assume that the inputs and the output of the system are as follows:

- The range of the first input named “Mature” is set to [0 25]. The “Mature” input is represented by three membership functions “Low”, “Medium” and “High” defined respectively by the following ranges [0 5 10], [7.5 12.5 17.5] and [15 20 25].
- The range of the second input named “SemiMature” is set to [0 25]. The “SemiMature” input is represented by three membership functions ”Low”, “Medium” and”High” defined respectively by the following ranges [0 5 10], [7.5 12.5 17.5] and [15 20 25].
- The range of the output named “Maturity” is set to [0 25]. The “Maturity” output is represented by two membership functions “Semi-Mature” and “Mature” defined respectively by the following ranges [0 8 15] and [10 18 25] .

The Fuzzy System Description FCDCM is composed of two inputs, one output and nine rules. It is based on the min-max aggregation method and the centroid deffuzification method.

The Inputs’ Membership Function Values Let us remind that the output of applying the fuzzy clustering algorithms is a set of membership functions degrees of each data point. That is how the membership function values of each input and output could be generated. For instance, when setting x = 8 and by checking the generated fuzzy clustering membership functions degrees list, we get the following vector [0.4 0.1 0]. The same process is applied to get the value of y and we obtain [0.2 0.3 0].

Rules’ Application Since the value of the membership function “High” is set to 0 for both inputs vectors, we look for the rules where these linguistic variables are not accorded. Hence, we apply the following rules:

1. If (Mature is Low) and (SemiMature is Low) then (Maturity is Mature)
2. If (Mature is Low) and (SemiMature is Medium) then (Maturity is Semi-Mature)
4. If (Mature is Medium) and (SemiMature is Low) then (Maturity is Mature)
5. If (Mature is Medium) and (SemiMature is Medium) then (Maturity is Semi-Mature)

After that, we apply the Mamdani method by taking the min between the condition of each rule and the max between the rules in order to generate one output value. This is achieved as follows:

1. min(0.4 , 0.2) = 0.2 (Mature)
2. min(0.4 , 0.3) = 0.3 (Semi-Mature)
4. min(0.1 , 0.2) = 0.1 (Mature)
5. min(0.1 , 0.3) = 0.1 (Semi-Mature)

Now we apply the max operator:

- max(0.2 , 0.1) = 0.2 (Mature)
- max(0.3 , 0.1) = 0.3 (Semi-mature)

Figure 5.3 shows the application phase of the rules and the generation of the centroid value which is set to 11.6.

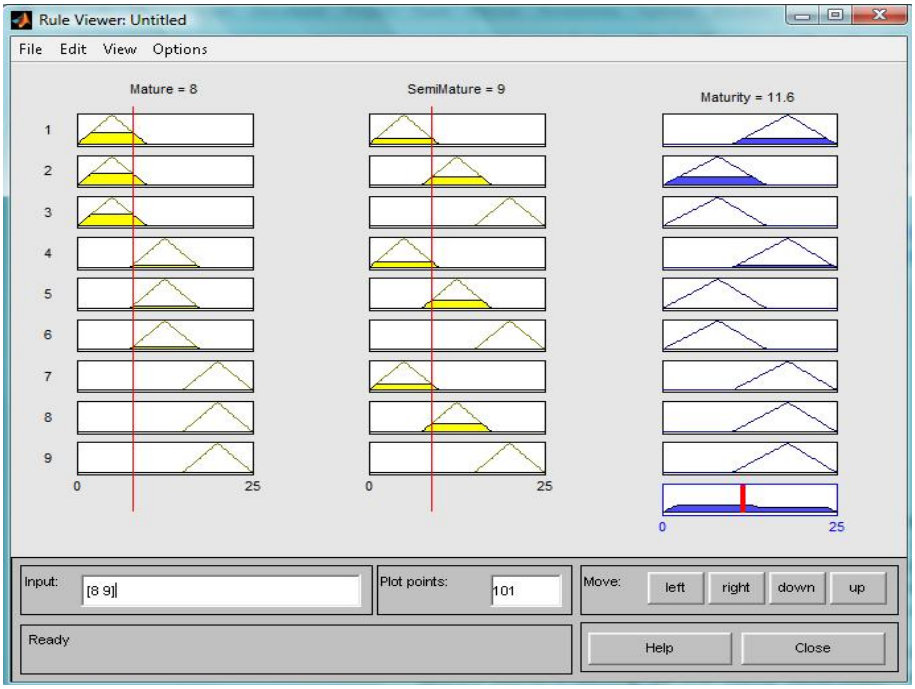

Figure 5.3: An Illustrative Schema of the Centroid Value Generation

Once the centroid value is generated, the final context of the object could be fixed as mature or semi-mature. This is achieved by the comparison of the middle of the output range (middle = 25/2 = 12.5) and the centroid value 11.6. As we remark, the middle of the output range is greater than the centroid value (12.5 > 11.6). This is, also, seen from Figure 5.3 where the surface of the "Maturity" output is skewed to the left (negative asymmetry). Hence, the final context of the object is "Semi-Mature".

The Classification Phase

Once the context is fixed for all the objects, the classification phase has to be executed. Thus, we have to calculate the anomaly threshold *at*:

$$at = \frac{3}{6} = 0.5$$

We suppose that each antigen (client) is repeated 10 times and the first client is detected in the mature state 3 times. Then, we can conclude that client1 is allowed to have a credit since (0.3=3/10<0.5). The same reasoning is applied to the rest of the instances.

5.5 Experimental Methodology, Results and Discussion

5.5.1 Parameters Settings and Evaluating Criteria

In this Section, we try to investigate the performance of the hybridization of our FCDCM proposed version with the various fuzzy clustering techniques previously seen. Different experiments are performed using the two-class data sets described in Table 3.2. We will compare first of all the FCDCM versions, including $FCDCM_{FCM}$, $FCDCM_{GK}$ and $FCDCM_{AFCS}$, to each other in order to select the best fuzzy clustering technique that should be hybridized with FCDCM. Then, we will show that FCDCM can solve the standard DCA limitation as it is sensitive to the input class data order. Concerning the algorithms parameters, they are the same presented in Chapter 3 with the PCA-DCA standard version. As for the parameter of the fuzzy clustering techniques, the value of the fuzzification exponent m is set to $m = 2$. This value is chosen since it has the advantage of simplifying the update equations and can therefore speed up computer implementations of the different algorithms (Nikhil, Bezdek, & James, 1995).

In order to choose the convenient fuzzy clustering technique, we have to compare them to each other. To do so, the following criteria may be used (Xie & Beni, 1991):

1. Partition Coefficient (PC): This validity measure rates the crispness of a classification. The more crisp the membership degree the better the classification. The PC has to be maximized. It is defined as follows:

$$PC(c) = \frac{1}{N}\sum_{i=1}^{c}\sum_{k=1}^{N}(\mu_{ik})^2 \tag{5.23}$$

2. Partition Index (PI): This index is the ratio of the sum of compactness and separation of the clusters. It is a sum of individual cluster validity measures normalized through division by the fuzzy cardinality of each cluster. PI is useful when comparing different partitions having equal number of clusters. A lower value of PI indicates a better partition.

$$PI(c) = \sum_{i=1}^{c}\frac{\sum_{k=1}^{N}(\mu_{ik})^m\|x_k - p_i\|^2}{N_i\sum_{j=1}^{c}\|p_j - p_i\|^2} \tag{5.24}$$

3. Xie and Beni's Index (XB): It aims to quantify the ratio of the total variation within clusters and the separation of clusters. The XB value has to be minimized.

$$XB(c) = \frac{\sum_{i=1}^{c}\sum_{k=1}^{N}(\mu_{ik})^m\|x_k - p_i\|^2}{N_{min_{i,k}}\|x_k - p_i\|^2} \tag{5.25}$$

4. Dunn's Index (DI): This index is originally proposed to use at the identification of compact and well separated clusters. So, the result of the clustering has to be recalculated as it was a hard partition algorithm. The DI value has to be minimized.

$$DI(c) = min_{i\in c}\{min_{k\in c,k\neq i}\{\frac{min_{x\in C_i,y\in C_k}d(x,y)}{max_{j\in c}\{max_{x,y\in C}d(x,y)\}}\}\} \tag{5.26}$$

These criteria, varying from 0% to 100%, help us to find the right fuzzy clustering technique which is capable of better detecting the needed parameters for our algorithm. It is, also, important to measure the performance of the fuzzy clustering techniques in terms of sensitivity, specificity, accuracy and execution time. This is because we

are looking for the right technique which performs better than the others in terms of detecting the most adapted parameters and the one taking less time to process. Yet, it is important to mention that in this part we are focusing on the DCA context assessment phase and not on the algorithm data pre-processing phase. Hence, the number of selected features is not used as a criterion for the sake of comparison.

5.5.2 Results and Discussion

Results about the Fuzzy Clustering Techniques

Table 5.6 presents a comparison between our FCDCM various versions in terms of validity measures; including PC, PI, XB and DI. From Table 5.6, we can notice that in most databases results provided by $FCDCM_{GK}$ are notably better than the ones provided by $FCDCM_{FCM}$ and $FCDCM_{AFCS}$. We can see that $FCDCM_{GK}$ takes the minimum values of PI, XB and DI and the highest value of PC comparing them to $FCDCM_{FCM}$ and $FCDCM_{AFCS}$ values. For instance, by applying $FCDCM_{GK}$ to the SN database, the PC takes the highest value of 84.30%. Whereas, with $FCDCM_{FCM}$ and $FCDCM_{AFCS}$ the PC values are 84.18% and 80.64%, respectively.

The same interesting results occur with the PI, XB and DI validity measures where $FCDCM_{GK}$ takes the minimum values on these measures, on the same database, in comparison to the $FCDCM_{FCM}$ and $FCDCM_{AFCS}$ values. These good $FCDCM_{GK}$ results are explained by the appropriate use of the Gustafson-Kessel algorithm (GK) which is more appropriate than FCM and AFCS at detecting and finding the elongated clusters. Hence, we can conclude that $FCDCM_{GK}$ is capable of fixing the extent and the midpoints of the membership functions better than $FCDCM_{FCM}$ and $FCDCM_{AFCS}$. This is, also, confirmed by the results presented in Table 5.7.

From Table 5.7, we can conclude again that $FCDCM_{GK}$ outperforms both $FCDCM_{FCM}$ and $FCDCM_{AFCS}$ in terms of sensitivity, specificity and classification accuracy, in most databases. For example, by applying $FCDCM_{GK}$ to the SN database the classification accuracy is set to 82.69%. However, when applying $FCDCM_{FCM}$ and $FCDCM_{AFCS}$ to the same database the accuracy is set to 79.32% and 80.28%, respectively.

Another advantage of $FCDCM_{GK}$ is that it takes less time to process than $FCDCM_{FCM}$ but a bit longer than $FCDCM_{AFCS}$. For example, by applying $FCDCM_{GK}$ to the Bio database the time taken by the algorithm to process is set to 3.06(s). However, when applying $FCDCM_{FCM}$ and $FCDCM_{AFCS}$ to the same database the time taken by the algorithms to process is set to 3.71(s) and 2.98(s), respectively.

To conclude, from the obtained results we notice that it is more appropriate to hybridize the GK algorithm with FCDCM than with FCM and $AFCS$ since the former hybridized algorithm produces more accurate results in terms of the mentioned evaluation criteria.

The choice of the $FCDCM_{GK}$ algorithm is mainly based on the classification accuracy criterion. This is explained by the fact that we are looking for an algorithm that will guarantee a stability in terms of classification accuracy.

Table 5.6: Numerical Values of Validity Measures (%)

Database	Method	PC	PI	XB	DI
SN	$FCDCM_{FCM}$	84.18	6.14	72.20	0.43
	$FCDCM_{AFCS}$	80.64	7.91	78.63	2.84
	$FCDCM_{GK}$	84.30	4.25	47.28	0.33
Bio	$FCDCM_{FCM}$	42.01	3.10	31.40	0.20
	$FCDCM_{AFCS}$	40.98	3.99	34.11	1.34
	$FCDCM_{GK}$	42.19	2.03	23.44	0.12
Cylb	$FCDCM_{FCM}$	50.53	3.60	26.23	4.16
	$FCDCM_{AFCS}$	40.15	25.22	27.73	3.61
	$FCDCM_{GK}$	54.96	1.20	18.10	4.18
Ch	$FCDCM_{FCM}$	70.28	16.94	4.55	2.20
	$FCDCM_{AFCS}$	59.90	27.17	6.44	2.28
	$FCDCM_{GK}$	74.71	14.54	2.54	1.63
IONO	$FCDCM_{FCM}$	99.98	0.93	93.16	20.78
	$FCDCM_{AFCS}$	90.01	0.34	62.04	21.09
	$FCDCM_{GK}$	99.99	0.11	23.09	20.91
Sck	$FCDCM_{FCM}$	76.04	22.70	10.31	7.96
	$FCDCM_{AFCS}$	65.66	32.93	12.20	8.04
	$FCDCM_{GK}$	80.47	20.30	8.30	7.39
HC	$FCDCM_{FCM}$	98.28	17.93	33.72	40.02
	$FCDCM_{AFCS}$	73.01	18.40	46.11	41.73
	$FCDCM_{GK}$	99.29	17.11	24.99	39.98
GC	$FCDCM_{FCM}$	23.42	05.64	1.54	0.73
	$FCDCM_{AFCS}$	19.96	9.05	2.14	0.76
	$FCDCM_{GK}$	24.90	4.84	0.84	0.54
LR	$FCDCM_{FCM}$	99.99	0.92	36.01	34.24
	$FCDCM_{AFCS}$	97.81	4.67	93.42	34.21
	$FCDCM_{GK}$	99.99	0.04	47.13	34.24
RWW	$FCDCM_{FCM}$	89.12	71.11	65.13	5.63
	$FCDCM_{AFCS}$	80.04	80.95	66.47	5.61
	$FCDCM_{GK}$	91.01	54.98	60.08	4.89

These encouraging results obtained from our $FCDCM_{GK}$ are explained by the most convenient parameters used by the algorithm to perform its classification task. These results were published in (Chelly & Elouedi, 2015).

Table 5.7: Experimental Measures

Database	Sensitivity (%)			Specificity (%)			Accuracy (%)			Time (s)		
	FCDCM			FCDCM			FCDCM			FCDCM		
	FCM	AFCS	GK	FCM	AFCS	GK	FCM	AFCS	GK	FCM	AFCS	GK
SN	82.88	83.78	85.58	75.25	76.28	79.38	79.32	80.28	82.69	4.45	3.75	4.01
Bio	45.28	47.16	50.94	43.39	45.28	49.05	44.33	46.22	50.00	3.71	2.98	3.06
Cylb	92.54	93.42	94.55	93.26	93.58	92.50	92.96	93.51	93.75	14.51	12.88	13.02
Ch	94.36	94.60	94.66	93.77	94.23	94.95	94.08	94.43	94.80	190.11	185.04	188.65
IONO	96.44	96.88	97.77	92.85	93.65	95.23	95.15	95.72	96.86	16.19	15.90	14.91
Sck	95.67	96.10	96.53	94.74	94.83	95.08	94.80	94.90	95.17	292.56	270.12	277.23
HC	92.12	93.05	92.59	78.94	86.84	85.52	86.68	90.48	89.67	13.61	12.03	12.98
GC	83.21	83.57	84.67	90.22	90.35	90.63	88.30	88.50	89.00	64.23	62.98	63.88
LR	80.00	75.00	70.00	83.78	81.08	78.37	82.45	78.94	75.43	0.72	0.58	0.63
RWW	97.99	98.01	98.18	98.12	98.24	98.74	98.02	98.07	98.32	526.99	521.01	521.32

The next step, now, is to check if $FCDCM_{GK}$ is capable to overcome the DCA limitation as it is sensitive to the input class data order. This will be discussed in what follows.

Results about the Algorithms Sensitivity

In (Greensmith & Aickelin, 2008b) it was shown that the DCA is sensitive to the input class data order. It was shown that the DCA misclassifications occur exclusively at the transition boundaries. Hence, DCA makes more mistakes when the context changes multiple times in a quick succession unlike when data are ordered between classes; class 1 and class 2. This problem was partially solved in (Chelly & Elouedi, 2010) with FDCM via the new fuzzy context assessment phase. However, FDCM gives unsatisfactory results in case of ordered contexts which could be explained by involving the ordinary user in the determination of the system's parameters. To handle the drawbacks of both DCA and FDCM, we developed our $FCDCM_{GK}$ which is based on generating automatically the parameters of the fuzzy inference system.

In this Section, we try to show that by the development of $FCDCM_{GK}$ we can obtain a stable DCA classifier. Thus, we will focus only on the classification accuracy criterion for the sake of comparison. To test the results, three different data orders are used. Experiment one uses a 1-step data order. Here, data are ordered continuously i.e. all class 1 items are processed followed by all class 2 items. In experiment two, data are partitioned into three sections resulting in a 2-step data order. The data comprising class 1 is split into two sections and the class 2 data is embedded between the class 1 partitions. Experiment three consists of data randomized between class 1 and class 2. The results of these experiments are displayed in Table 5.8.

From Table 5.8, it is clearly noticed that our $FCDCM_{GK}$ generates stable classification results among the three experiments. Hence, solving the DCA shortcoming as it is sensitive to the input class data order. For instance, by applying the $FCDCM_{GK}$ method to the IONO training data set the accuracy of the algorithm is stable and set to 96.86%. Nevertheless, when applying the DCA to the same data set, the accuracy of the algorithm decreases from 94.58% to 66.09%.

This high value of the DCA accuracy (94.58%) in case of an ordered training data set is explained by the appropriate use of this algorithm only in an ordered case. From the 2-Step experiment to the Random one, the DCA accuracy decreases from 78.63% to 66.09%. This behavior shows that the DCA is sensitive to the input class data order confirming the results obtained from literature. However, this problem is solved using the $FCDCM_{GK}$ since we notice a stability in the algorithm classification results through the different experiments realized. Indeed, by applying the FDCM to the same database, the accuracy is stable and set to 94.87% in case of disordered contexts and in case of an ordered context it takes a value of 88.31%. This latter low value is explained by the use of arbitrary midpoints and extents for the system's membership functions. Another important characteristic of our $FCDCM_{GK}$ algorithm is that it gives better classification results when compared to the DCA and the FDCM algorithms.

To sum up, in this Section we have shown that the problem of sensitivity is solved by developing our $FCDCM_{GK}$ algorithm. Our method aims at smoothing the crisp separation between the two contexts while generating automatically the parameters of the system leading to better results in terms of classification accuracy. These results were highlighted in (Chelly & Elouedi, 2011).

Table 5.8: Experimental Measure: Accuracy (%)

Database	DCA			FDCM			$FCDCM_{GK}$		
	1-Step	2-Step	Random	1-Step	2-Step	Random	1-Step	2-Step	Random
SN	77.88	74.51	69.71	69.23	79.32	79.32	82.69	82.69	82.69
Bio	41.50	40.56	39.62	40.56	43.39	43.39	50.00	50.00	50.00
Cylb	92.38	90.55	87.03	89.25	92.96	93.14	93.70	93.75	93.75
Ch	93.86	93.05	90.73	91.05	94.49	94.49	94.80	94.80	94.80
IONO	94.58	78.63	66.09	78.34	85.75	97.15	96.86	96.86	96.86
Sck	94.11	92.78	91.99	93.79	94.72	94.72	94.80	95.17	95.17
HC	83.96	81.52	80.16	81.52	87.50	87.50	89.67	89.67	89.67
GC	87.00	86.80	86.50	86.50	87.90	87.90	89.00	89.00	89.00
LR	84.21	82.45	78.94	77.19	78.94	80.70	75.43	75.43	75.43
RWW	97.87	96.96	95.42	98.01	98.19	98.19	98.32	98.32	98.32

5.5.3 Comparison with State-of-the-art Recent Methods

It this Section, we will compare the classification results of our $FCDCM_{GK}$ algorithm with well known classifiers which are the Support Vector Machine (SVM), the Artificial Neural Network (ANN) and the Decision Tree (DTree). We will, also, add the classical DCA to the comparison in order to show its performance in case of ordered data sets and in case of non-ordered databases. To do so, we have ordered the label classes of the used databases in order to get all class 1 followed by all class 2, and then we have applied the DCA to test its classification accuracy on these ordered data sets. We have named this case as DCA_O stating the application of DCA in case of Ordered input label classes. Secondly, we have applied DCA to Non-Ordered databases, case named DCA_{NO}, to show that DCA loses its performance leading the algorithm to be a sensitive binary classifier. On these non-ordered databases, we have applied the rest of the classifiers including our proposed $FCDCM_{GK}$, SVM, ANN and DTree. As we are focusing on the classification accuracy of the algorithms, the comparison made is in terms of the average of accuracies on the used databases and is presented in Figure 5.4.

First of all, we can notice that Figure 5.4 shows that the standard DCA when applied to ordered data sets (DCA_O) outperforms SVM, ANN and DTree in terms of overall classification accuracy. This interesting point is, unfortunately, not seen when the algorithm is applied to non-ordered databases (DCA_{NO}). This confirms the results shown in literature about the DCA sensitivity to the input class data order. However, if we look to the DCA classifier when applied to ordered databases we confirm that it is capable of producing satisfactory classification results in comparison to the state-of-the-art classifiers.

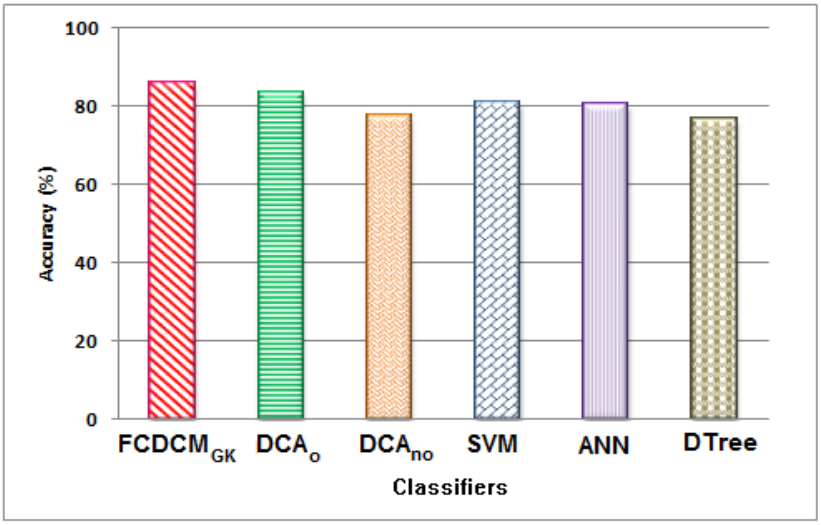

Figure 5.4: Comparison of Classifiers Average Accuracies

Secondly and most importantly, from Figure 5.4, we can notice that our developed $FCDCM_{GK}$ algorithm outperforms the mentioned classifiers including DCA_O, SVM, ANN and DTree in terms of overall accuracy. These encouraging $FCDCM_{GK}$ results are explained by the appropriate hybridization of our FCDCM with the GK algorithm in order to generate the most convenient parameters for our system. Consequently, we have obtained a novel immune-inspired fuzzy model making the standard DCA a better classifier by generating pertinent and more reliable and stable results.

5.6 Conclusion

In this Chapter, we have analyzed the behavior of the standard DCA while trying to overcome its shortcoming as it is sensitive to the class data order. This investigation led us to the development of the FCDCM algorithm. FCDCM presents a fuzzy version of the DCA and it aims at coping with the crisp separation between the two DCs contexts. In order to fix the parameters of the algorithm, we have used various fuzzy clustering techniques, compared them and chosen the most appropriate one. The experimental results demonstrate that our proposed method with the Gustafson-Kessel algorithm ($FCDCM_{GK}$) achieves better results compared to other methods.

In the next Chapter, we will try to investigate more the causes of the dendritic cell algorithm sensitivity to the input class data order. We will focus, mainly, on the dendritic cell algorithm detection phase while suggesting new automated and more adequate solutions for the DCA.

Chapter 6

The Fuzzy Maintained Dendritic Cell Classification Method

6.1 Introduction

Still focusing on the DCA sensitivity to the input class data order, we have noticed that this limitation cannot only be linked to the DCA context assessment phase, as presented in the previous Chapter, but also to the DCA detection phase. We hypothesize that there is a second possible cause related to the DCA sensitivity. It is possible that the induced signal base generated by the DCA detection phase contains disagreeable objects such as noisy, incoherent or redundant instances. Such objects may influence the DCA behavior and, thus, maintaining the signal base looks essential. To achieve this, we propose in this Chapter to expand a second fuzzy DCA version based on a maintenance technique.

This Chapter is structured as follows: In Section 6.2, we give an overview of the maintenance database policies while in Section 6.3, we present our new fuzzy maintained DCA. The experiments and the obtained results are outlined in Section 6.4.

6.2 Overview of the Maintenance Database Policies

The quality of the database is very important to generate accurate results and to have the possibility to learn models from the presented data. To achieve this, the disagreeable objects - especially noisy and redundant instances which affect negatively the quality of the results - have to be eliminated from the data set. To address this situation, data maintenance is a feasible way. The goal of database maintenance methods is to select the most representative information from a given data set.

Many works dealing with database maintenance have been proposed in literature (Leake & Wilson, 2001).

Most of them are based on updating a database by adding or deleting instances to optimize and reduce the initial database. These policies include different operations such as: the outdated, redundant or inconsistent instances may be deleted, groups of objects may be merged to eliminate redundancy and improve reasoning power, objects may be re-described to repair incoherencies, signs of corruption in the database have to be checked and any abnormalities in the database which might signal a problem has to be controlled. A database which is not maintained can become sluggish and without accurate data users will make uninformed decisions. The database maintenance policies may be categorized into two main heads; the selective reduction approaches and the deletion reduction approaches.

6.2.1 The Selective Reduction Approaches

The selective reduction approaches start with an empty set, select a subset of instances from the original set and add it into the new one. Actually, the selection methods aim to reduce a data set by selecting representatives from the training data set. Among the most efficient selection reduction data methods shown in (Leake & Wilson, 2001), we can mention the Condensed Nearest Neighbor Rule (CNN) (Chou, Kuo, & Chang, 2006). CNN is a redundancy technique that incrementally builds an edited database from scratch. Instances are added to a new database and removed from the original one; if and only if; they cannot be correctly classified by the edited base built so far. However, CNN suffers from some shortcomings as it is sensitive to noise. This means that CNN can view the noisy cases or instances as important exceptions and give an unsatisfying result.

Another selection reduction method is the Reduced Nearest Neighbor Rule (RNN) (Gates, 1972) which starts from using the whole training set as the initial reduced set: $S = T$, and removes each instance from S if such a removal does not cause any other instances in T to be misclassified by the instances remaining in S. The process is repeated until no further reduction can be achieved. Nevertheless, the iterative process is very time-consuming if the original training set is large and it is computationally more expensive than the CNN method. The Edited Nearest Neighbor Rules (ENN) (Wilson, 1972) is, also, a selective reduction method which removes all instances which have been misclassified by the k-NN from the training set. ENN keeps all the internal instances but deletes the border instances as well as the noisy instances, unlike the CNN algorithm.

6.2.2 The Deletion Reduction Approaches

Another type of maintenance methods is the deletion reduction methods which addressed the problem of maintenance optimization. In fact, from a given database these strategies are able to suppress “useless” instances and bring the base to a specific number of cases. Some researchers advocate a random deletion or addition policy (Markovich & Scott, 1988). In this policy, a random case is removed from the base once the data set size exceeds some predefined threshold. It is necessary to set the initial threshold capacity for the base. The random method may suffer from some limitations as it is based on the fact of removing randomly the base instances. Sometimes the key cases may be deleted. Some improvements of this policy were noticed in (Minton, 1990). Similar policies have been employed elsewhere with equally successful results in (Markovitch & Scott, 1993). In (Racine & Yang, 1996, 1997), other deletion methods based on eliminating any kind of redundancy and inconsistency from the database were described.

As mentioned previously, by maintaining the database we want to isolate the smallest set of instances which enable us to predict the class of a query instance with the same quality as the initial data set. Another maintenance method, not belonging to the mentioned approaches, has to be mentioned which is the Clustering, Outliers and In-

ternal cases Detection deletion method (COID) proposed in (Smiti & Elouedi, 2010). It uses a clustering technique to create small bases. Then, it applies outliers and internal cases detection methods, for each generated cluster, to reduce the size. This method aims at selecting cases which influence the quality of the initial base. Thus, in this method the clustering ensures that each base is small and it is easier to maintain each one individually. For each small cluster, the cases of type outliers and the cases which are near to the center of the cluster are kept and the rest of the cases is removed.

We intend to compare the mentioned maintenance policies and to apply the appropriate one to the DCA signal database. This will lead to the development of our new fuzzy maintained DCA. In fact, our proposed solution will be based on the COID maintenance technique (Smiti & Elouedi, 2010). COID has shown promising results in the Case Base Reasoning field (Smiti & Elouedi, 2011) while outperforming other maintenance methods. This encouraged us to use COID. Yet, this choice will be further argued and endorsed later in the experimental setup Section.

6.3 The Fuzzy Maintained COID-$FCDCM_{GK}$ Algorithm

Our goal is to overcome the DCA sensitivity to the input class data order. We hypothesize that the cause of such sensitivity can be related to the DCA detection phase where the used signal database may contain disagreeable objects such as noisy, incoherent or redundant instances. We, thus, propose in this Chapter a solution which is based on this hypothesis to overcome the mentioned DCA shortcoming. Our solution, named COID-Fuzzy Dendritic Cell Method (COID-$FCDCM_{GK}$), is based on the use of the COID maintenance policy and the previously developed FCDCM method based on the Gustafson-Kessel Algorithm algorithm (GK) algorithm ($FCDCM_{GK}$).

6.3.1 COID-$FCDCM_{GK}$ Layout

COID-$FCDCM_{GK}$ Architecture

COID-$FCDCM_{GK}$ is an extension of $FCDCM_{GK}$ which was described in the previous Chapter. COID-$FCDCM_{GK}$ takes into account the fact of alleviating the crisp assessment task by the use of fuzzy set theory and generating automatically the parameters of the algorithm. This will be achieved by the use of the GK fuzzy clustering technique. Adding to these, COID-$FCDCM_{GK}$ takes into account the fact of maintaining the DCA signal database by eliminating its "useless" objects. This will be achieved by the use of the COID technique. The choice of this specific method will be, later, argued in our experimental setup Section where we will make a comparison of different maintenance policies and select COID as the most interesting technique among others. Figure 6.1 shows our COID-$FCDCM_{GK}$ main steps.

From Figure 6.1, we can notice that COID-$FCDCM_{GK}$ is based on the same phases as $FCDCM_{GK}$ (see Figure 5.2); except for the detection phase. COID-$FCDCM_{GK}$ proposes a new detection phase to deal with the inconsistency of the signal data set. The architecture of the latter phase is described in what is next.

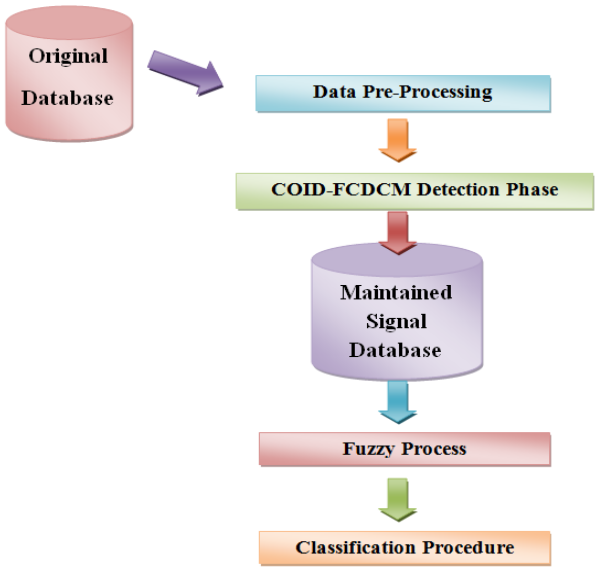

Figure 6.1: The COID-$FCDCM_{GK}$ Steps

COID-$FCDCM_{GK}$ Detection Phase Architecture

Throughout the new detection phase, COID-$FCDCM_{GK}$ incorporates the COID maintenance database method. COID is applied for the purpose of reducing the size of the initial input signal database while maintaining its quality. Formally, this is achieved by removing the noisy antigens and keeping the most important ones. COID works under two main levels which are the following:

- Clustering of antigens: the initial signal database is decomposed into groups of closely related antigens. Thus, we obtain independent small groups of sub signal databases.
- Detection of outliers and internal antigens: outliers and antigens which are close to the center of each cluster are selected and the other antigens are deleted.

The two steps of the COID method which are sub-steps of the COID-$FCDCM_{GK}$ detection phase are presented in Figure 6.2:

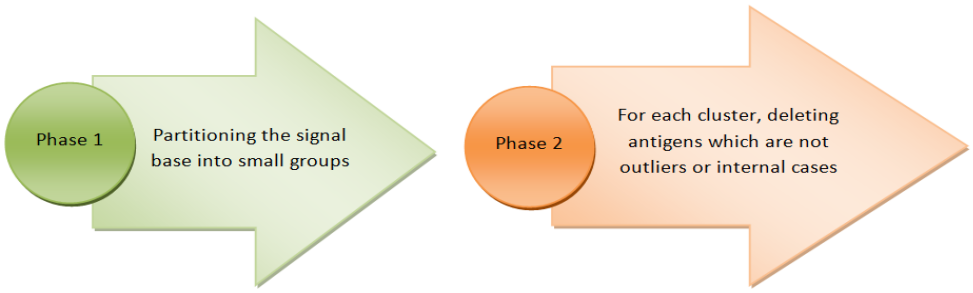

Figure 6.2: The COID-$FCDCM_{GK}$ Detection Phase Steps

6.3.2 COID-FCDCM Functioning

The COID-$FCDCM_{GK}$ algorithmic steps are performed the same as $FCDCM_{GK}$ except for the detection phase. Thus, in this Section, we will mainly focus on this phase by giving a detailed description of it.

COID-FCDCM Detection Phase Basic Concepts

Based on our hypothesis, the DCA signal database contains disagreeable objects especially noisy and redundant instances, in the sense that it affects negatively the behavior of the algorithm. To guarantee the stability of the DCA, the maintenance of the signal database becomes essential. Thus, we apply the COID approach in the detection phase of our proposed COID-$FCDCM_{GK}$ algorithm. As a result, a new smaller signal base is constructed to represent the original one. To achieve this, COID-$FCDCM_{GK}$ defines two important types of objects which should not be deleted from the original DCA signal database. These types of objects are described in Figure 6.3.

- Outlier: is an isolated instance, no other object can replace it or be similar to it. By eliminating outliers, the performance of the system will be negatively affected since there will not be any other objects that can solve the deleted outliers.
- Internal instance: is one object from a group of similar objects. Each instance from this group provides similar coverage to the other instances of the same group. Deleting any member of this group has no effect on the system's performance since the remaining objects offer the same value. However, deleting the entire group is tantamount as deleting an outlier as the system's performance is reduced. Thus, we have to keep one instance from each group of similar objects.

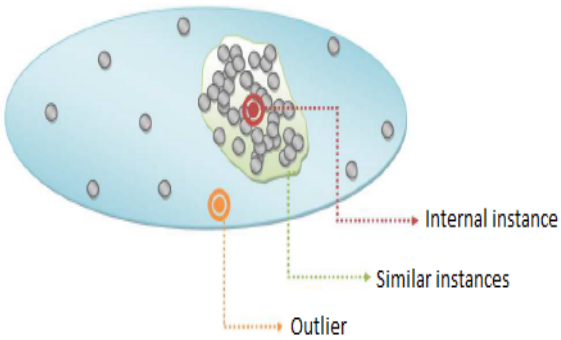

Figure 6.3: Instance Categories

Based on this idea, the COID method reduces the original signal database by keeping only these two types of instances and eliminates the rest. For that, it uses a clustering method to create groups of similar instances. Then, for each cluster, it applies specific methods to detect outliers and others to select the internal instances which are the nearest instances to the center of one cluster. The COID-$FCDCM_{GK}$ maintaining steps are itemized as follows:

- For the clustering method, COID adopts the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) (Sander, Ester, Kriegel, & Xu, 1998) since it offers minimal requirements of domain knowledge to determine the input parameters. Moreover, it can detect the noisy instances. Hence, the improvement of classification accuracy could be achieved.

- For the selection of the internal instances, COID calculates the Euclidean distance between the cluster's center and each instance from the same cluster and selects objects which have the smallest distance.
- For the outliers' detection, COID applies two outliers' detection methods to announce the univariate outliers and the multivariate ones. For univariate outliers detection, COID uses the InterQuartile Range (IQR) (Bussian & Härdle, 1984) as it is a robust statistical method and less sensitive to presence of outliers. For multivariate outliers detection, the Mahalanobis distance is used. The Mahalanobis distance takes into account the covariance among the variables in calculating distances. With this measure, the problems of scale and correlation are no longer an issue (Filzmoser, Garrett, & Reimann, 2005).

Consequently, the result of applying COID is the generation of a new reduced signal database lacking noisy and redundant objects. The generated base preserves nearly the same performance of the original DCA signal data set. Hence, our COID-$FCDCM_{GK}$ can treat the new signal database easily and it guarantees better classification results.

The Signal Base Maintenance Process

In this Section, we provide a detailed description of how to maintain the signal data set. The relative workflow is, thus, illustrated in Figure 6.4. First of all, throughout the COID-$FCDCM_{GK}$ detection phase the algorithm has to generate groups of similar antigens (Figure 6.4(b)). This step is handled by the use of the DBSCAN clustering method. Once the DBSCAN detects noisy objects and partitions the signal base, COID-$FCDCM_{GK}$ removes the detected noisy instances (Figure 6.4(c)). The result of this process is a set of independent small signal databases (Figure 6.4(d)).

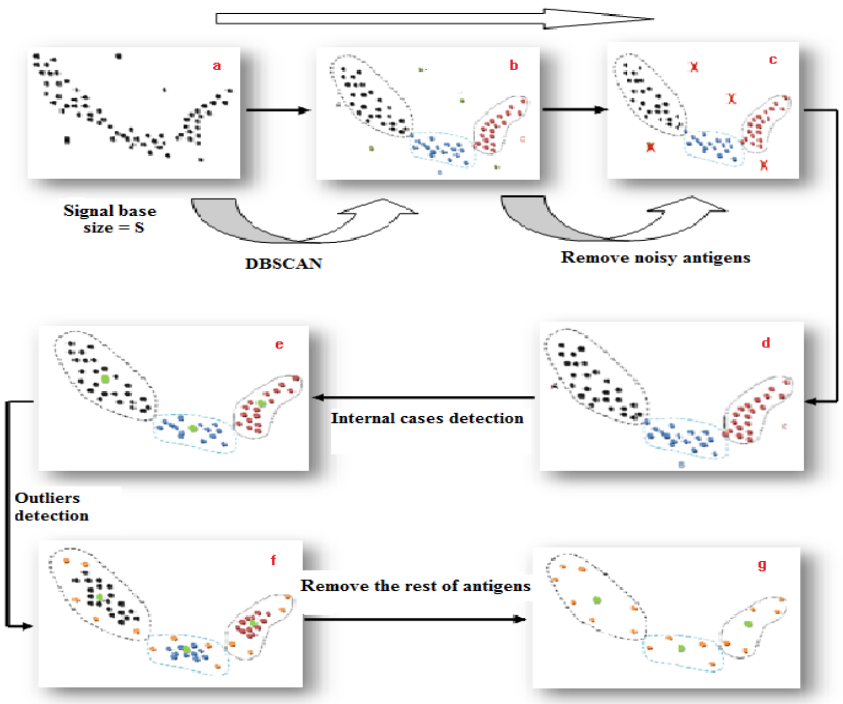

Figure 6.4: Overview of the COID-$FCDCM_{GK}$ Maintenance Process

The second step of the COID-$FCDCM_{GK}$ detection phase is to select antigens which are close to the center of each cluster (Figure 6.4(e)). After the detection of the internal antigens, COID-$FCDCM_{GK}$ applies outliers detection methods to announce outliers (Figure 6.4(f)). Finally, all unselected antigens, which are not outliers or internal cases, are deleted from the signal data set (Figure 6.4(g)).

Based on these steps, we obtain a reduced signal data set as shown in Figure 6.5.

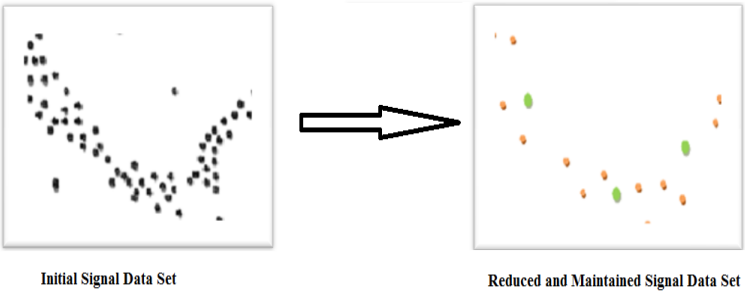

Figure 6.5: Reduced Signal Data Set

6.3.3 COID-$FCDCM_{GK}$: Worked Example

In this Section, an example of the COID-$FCDCM_{GK}$ detection phase and more precisely the maintenance phase is highlighted. Table 6.1 presents the signal database on which COID-$FCDCM_{GK}$ will apply the COID maintenance method in order to get a consistent new signal data set.

Table 6.1: Signal Data Set

Antigen (Client)	SS	PAMP	DS
Ag1	0	350	2.1667
Ag2	310	0	12.32
Ag3	0	295	0.8333
Ag4	350	0	11.8333
Ag5	0	309	3.546
Ag6	394	0	13.99
Ag7	320	0	23.3F3
Ag8	0	321	3.663
Ag9	0	90	9.S3123
Ag10	402	0	10.65

The first COID-$FCDCM_{GK}$ maintenance procedure consists of generating small signal databases. This is achieved by the use of the DBSCAN clustering method. Applying DBSCAN results on the following clusters:

- Cluster 1: Ag2, Ag4, Ag6, Ag10

- Cluster 2: Ag1, Ag3, Ag5, Ag8

DBSCAN is, also, capable of defining noisy instances among the antigens of the signal data set. By pointing out the noisy antigens, DBSCAN will look for those items which are seen as meaningless data; i.e. the data that cannot be understood and interpreted correctly by machines. The set of noisy antigens is the following:

- Noisy antigens: Ag7, Ag9

After defining antigens' clusters and the noisy antigens, COID-$FCDCM_{GK}$ deletes these latter antigens. Noisy antigens should be deleted from the signal data set since they are seen as distortions of a value or the addition of a spurious object. Therefore, Ag7 and Ag9 are discarded from the signal base.

The second COID-$FCDCM_{GK}$ maintenance procedure consists of selecting the important antigens among the generated clusters. By important antigens, COID-$FCDCM_{GK}$ selects both internal instances and the outliers. The process is described in the itemized list below:

1. Selecting internal instances: As explained, internal objects are the nearest antigens to the center of one group. COID-$FCDCM_{GK}$ calculates the distance between the cluster's center and each antigen for the same cluster. COID-$FCDCM_{GK}$ chooses the antigens which have the smallest distance.
   - For the Cluster 1: The internal instance is represented by Ag6;
   - For the Cluster 2: The internal instance is represented by Ag3;
2. Selecting outliers: As stated previously, outliers are essential in the signal base because they effect directly the quality of the system. They could be seen as isolated antigens. Adding to this, outliers have no other antigens that can replace them. For each cluster, COID-$FCDCM_{GK}$ applies the previously discussed outlier detection methods. This process leads to detect the following outliers:
   - For the Cluster 1: The outlier is represented by Ag10;
   - For the Cluster 2: The outliers are represented by Ag1 and Ag8;

Once COID-$FCDCM_{GK}$ selects these instances, the proposed method deletes all of the rest of the instances which have not been selected by COID-$FCDCM_{GK}$. Thus, the following antigens are deleted:

- For the Cluster 1: COID-$FCDCM_{GK}$ deletes both Ag4 and Ag2;
- For the Cluster 2: COID-$FCDCM_{GK}$ deletes Ag5;

Consequently, applying COID-$FCDCM_{GK}$ leads to build a new reduced signal base which contains antigens that do not reduce the performance of the initial signal data set. The new generated maintained signal base contains the following antigens:

- New Signal base: Ag1, Ag3, Ag6, Ag8, Ag10

Once the new signal base is generated, COID-$FCDCM_{GK}$ process its next steps which are the fuzzy process and the classification phase as the $FCDCM_{GK}$ algorithm does and as detailed in the previous Chapter.

6.4 Experimental Methodology, Results and Discussion

6.4.1 Experimental Hypotheses and Evaluating Criteria

As previously stated, the DCA suffers from being sensitive to the input class data order. In this Section, we try to investigate the reasons of such a limitation. The list below presents the hypotheses of the possible causes that lead the DCA to be non-stable while performing its classification task.

- **H1:** The DCA sensitivity is related to the crisp separation between normality (semi-mature context) and anomaly (mature context).
- **H2:** The DCA sensitivity is related to the DCA signal database.

We try to analyze the behavior of the DCA and investigate the sources of its sensitivity. Therefore, our experimentations are based on two hypotheses: H1 and H2. In (Chelly & Elouedi, 2011, 2015), we have confirmed the trueness of H1. Therefore, in what follows, we will focus our experimentations on testing the trueness of H2. In this case, we will compare the classification results of DCA when it is applied to the original signal database with the results obtained from the DCA when it uses the maintained reduced signal database free from noisy and redundant instances, case we dub COID-DCA. This comparison is based on the variation of the input class data orders between the two classes, class 1 and class 2, to test the stability of the classifiers in such cases. Let us remind that we apply the COID method to guarantee the quality of the DCA signal database and that the choice of the COID method is based on a comparative study of different database maintenance policies.

We have divided our experimental analysis and result Section into four main parts.

1. We will show that the selected maintenance COID method that is hybridized with our proposed COID-$FCDCM_{GK}$ version is an interesting technique among other maintenance techniques proposed in literature.
2. We will test the trueness of H2 as H1 was already tested and confirmed in (Chelly & Elouedi, 2011, 2015). In this part, we will compare DCA and our proposed COID-DCA which is a modified DCA based, only, on H2. To do so, we have based our experimentations on randomizing the data between the classes to notice the effect of this randomization on the trueness of H2. We will analyze the behavior of our new COID-DCA in such situation. We try to show that the performance of COID-DCA, does neither depend on such transitions nor on ordered data sets contrary to the DCA. Thus, we can conclude that H2 is true and that by maintaining the signal base using COID, the problem related to the sensitivity to the input class data order is solved. To achieve this, three different data orders are used. Experiment one uses the 1-Step data order. In experiment two, data are partitioned into three sections, resulting in the 2-Step data order. Experiment three consists of data randomized between class 1 and class 2.
3. Once H2 is confirmed, we will show the performance of our proposed COID-$FCDCM_{GK}$ which takes into account both H1 and H2. We will compare the results obtained from (Chelly & Elouedi, 2011, 2015) and COID-DCA with the COID-$FCDCM_{GK}$ generated results. This is to show that if we take both hypotheses into consideration, we can generate a more robust binary classifier.
4. We will show that our COID-$FCDCM_{GK}$ algorithm outperforms not only the standard DCA but also other well known state-of-the-art classifiers.

To measure the performance of the algorithms, we will use the following criteria:

1. Storage reduction (S): S denotes the average storage percentage of the number of instances which are kept among the whole set of instances presented initially in the signal base. S is computed using the following formula:

$$S = \frac{\text{number of final instances}}{\text{size of training dataset}} * 100 \tag{6.1}$$

   The S value has to be minimized constraint to remain a satisfactory accuracy.

2. Sensitivity, Specificity, Accuracy and the Execution Time.

6.4.2 Results and Analysis of Maintenance Policies

To measure the performance of the COID method and to discover its characteristics, we compare it to other well-known maintenance techniques. The selected algorithms are among the most efficient ones shown in (Leake & Wilson, 2001). The used algorithms will be the Condensed Nearest Neighbor algorithm CNN (Chou et al., 2006), the Reduced Nearest Neighbor RNN technique (Gates, 1972) and the Instance Based learning IBL schemes (IB2 and IB3) (Aha, Kibler, & Albert, 1991). To compare the mentioned maintenance policies, we will mainly focus on the storage reduction, accuracy and execution time as main criteria. The results of this comparison are presented in Table 6.2.

From Table 6.2, we can remark that the reduction rate (S%) obtained using the COID maintenance method is notably better than the one provided by the other maintenance policies in most data sets. For instance, for the Ch data set COID keeps about 52.79% of the data instances and that is a huge difference compared to the initial Ch database with 100% antigens. Comparing the COID reduction rate to the rest of the maintenance policies rates, on the same database, the rates are 62.79%, 67.46%, 77.00% and 85.07% for CNN, RNN, IB2 and IB3, respectively.

We can, also, notice that in some data sets the reduction rate of COID is less than the rates of the other techniques. While focusing on these data sets, we can notice that the number of antigens is not that much important. Consequently, we can conclude that if we apply a maintenance technique to a small size signal base then the classification accuracy of the algorithm will be negatively affected and that is seen from the results displayed in Table 6.2. For example, when applying the COID technique to the LR database (57 antigens) the accuracy of the COID-DCA algorithm is set to 82.45%. However, when applying DCA using the whole signal base the accuracy of DCA is set to 84.21%. Same remark is noticed when applying the rest of the maintenance polices to the DCA signal data set where we will obtain less classification results in comparison to the initial whole set of the signal database.

Conversely, for the rest of the databases we notice that the classification accuracy of COID-DCA is even better than the DCA initial results when using the 100% of the signal base. For instance, when applying the COID-DCA to the HC base the classification accuracy of the algorithm is set to 87.77% which is more important than the classification accuracy of the DCA which is set to 83.96%. Now, regarding the execution time of the hybrid algorithms we can notice that in most data sets COID takes less time to process in comparison to the rest of the DCA hybrid maintenance polices.

Table 6.2: Comparing Database Maintenance Schemes

Database	Evaluation criteria	DCA	COID	CNN	RNN	IB2	IB3
SN	S (%)	100	98.00	78.00	65.00	67.00	88.00
	Accuracy (%)	77.88	76.92	62.50	68.75	69.80	73.00
	Execution Time (s)	1.36	1.25	1.14	1.30	1.31	1.29
Bio	S (%)	100	90.00	71.10	71.70	89.00	89.00
	Accuracy (%)	41.50	40.56	37.95	36.32	38.00	38.00
	Execution Time (s)	0.27	0.24	0.22	0.25	0.26	0.26
CylB	S (%)	100	86.52	96.70	96.60	98.00	93.98
	Accuracy (%)	92.38	93.51	86.52	76.50	72.95	72.95
	Execution Time (s)	11.31	10.12	11.04	11.13	11.09	10.66
Ch	S (%)	100	52.79	62.79	67.46	77.00	85.07
	Accuracy (%)	93.86	94.61	91.52	90.04	89.23	89.59
	Execution Time (s)	183.96	104.88	168.71	166.08	170.24	166.21
IONO	S (%)	100	56.00	64.39	62.08	62.12	61.00
	Accuracy (%)	94.58	95.44	87.40	89.19	87.39	88.26
	Execution Time (s)	7.79	4.06	5.7	5.0	5.3	4.8
Sck	S (%)	100	68.28	79.26	77.50	75.40	71.20
	Accuracy (%)	94.11	94.83	81.29	81.66	89.00	83.21
	Execution Time (s)	256.98	162.08	246.34	229.12	208.62	198.77
HC	S (%)	100	67.20	68.04	79.62	67.00	76.00
	Accuracy (%)	83.96	87.77	84.97	83.74	86.66	84.54
	Execution Time (s)	10.04	7.14	7.66	8.89	7.08	8.03
GC	S (%)	100	90.40	98.00	96.00	94.40	91.20
	Accuracy (%)	87.00	88.10	73.68	63.16	73.26	68.42
	Execution Time (s)	61.32	43.15	45.24	45.23	48.97	45.07
LR	S (%)	100	87.00	67.00	77.04	73.70	72.96
	Accuracy (%)	84.21	82.45	61.02	60.24	79.78	78.09
	Execution Time (s)	0.50	0.33	0.25	0.19	0.12	0.18
RWW	S (%)	100	76.92	77.38	77.80	89.00	86.79
	Accuracy (%)	97.87	97.90	97.40	96.26	97.29	97.45
	Execution Time (s)	513.97	326.75	394.66	409.52	460.03	448.21

From the obtained results, we have shown that the COID method is an interesting maintenance technique among others proposed in literature. We focused on its efficiency in terms of shrinking the size of databases by removing their "useless" instances, its reasonable and acceptable execution time and its ability to generate satisfactory classification results among the other studied techniques. These important characteristics are the base for choosing COID as an appropriate technique to use in our proposed method, COID-$FCDCM_{GK}$, in order to maintain the DCA signal database.

6.4.3 Results and Discussion about H2: Comparing DCA and COID-DCA

H2 states that the DCA sensitivity to the input class data order is related to the quality of the signal database applied to it. To test the trueness of this hypothesis, we compare the DCA classification results when it uses the original signal database to the DCA classification results when it uses the COID maintained reduced signal database; case we dub COID-DCA. This comparison is based on the variation of the input class data orders between class 1 and class 2, and thus we will focus on the classification accuracy of the algorithms as main criterion. The results of this

experiment are highlighted in Table 6.3.

Table 6.3: Experimental Measures: Accuracy (%)

Database	DCA			COID-DCA		
	1-Step	2-Step	Random	1-Step	2-Step	Random
SN	77.88	74.51	69.71	76.44	76.92	76.92
Bio	41.50	40.56	39.62	40.56	40.56	40.56
CylB	92.38	90.55	87.03	93.51	93.51	93.51
Ch	93.86	93.05	90.73	94.58	94.61	94.61
IONO	94.58	78.63	66.09	95.44	95.15	95.44
Sck	94.11	92.78	91.99	94.83	94.83	94.83
HC	83.96	81.52	80.16	87.77	87.77	87.77
GC	87.00	86.80	86.50	88.10	87.90	87.10
LR	84.21	82.45	78.94	82.45	82.45	82.45
RWW	97.87	96.96	95.42	97.90	97.90	97.90

From Table 6.3, we can notice that by maintaining the DCA signal database using the COID technique the percentage of classification accuracy is nearly stable between the three data orders; i.e., 1-Step, 2-Step and the Random experiment. Thus, the limitation of the algorithm which consists of being sensitive to the input class data order is overcame. From this observation, we can affirm that the standard DCA drawback is related to the quality of the DCA signal database and by maintaining it using the COID method the problem is solved. Consequently, our second hypothesis (H2) is approved.

More precisely, from Table 6.3, it is clearly noticed that the classification accuracy of the COID-DCA algorithm are nearly stable through the three realized experiments with comparison to the accuracies of the standard DCA results which are decreasing from the ordered contexts (1-Step experiment) to the disordered contexts (2-Step and Random experiment). This phenomenon could be explained by the fact that the original signal database contains noisy and redundant instances which affect negatively the performance of the algorithm. Thus, by using an appropriate maintenance policy, the COID method, we can guarantee the quality of the database leading to a stable DCA classifier.

For instance, when applying the COID-DCA method to the SN database the accuracy of the algorithm is around 76.92% and 76.44%. Nevertheless, when applying the DCA to the same database the accuracy of the algorithm decreases from 77.88% to 69.71%. This high value of the DCA accuracy (77.88%) in case of an ordered database is explained by the appropriate use of this algorithm only in an ordered case. From the 2-Step experiment to the Random one, the DCA accuracy decreases from 74.51% to 69.71%. This behavior shows that the DCA is sensitive to the input class data order which confirms the results obtained from literature. However, this problem is solved when using the COID-DCA since we notice a stability in the algorithm classification results through the different experiments realized. Another example can be the RWW database where the accuracy of the algorithm is stable and set to 97.90%.

To sum up, in this Section, we could approve that the second cause of the DCA sensitivity to the input class data order is related to the quality of its signal database. We have developed COID-DCA which is seen as a stable DCA classifier generating stable classification results through the variation of the input class data orders. From the obtained results, we can confirm the trueness of H2.

6.4.4 Results and Discussion about COID-$FCDCM_{GK}$

Based on the COID-DCA obtained results and from the results obtained from our first work (Chelly & Elouedi, 2011, 2015), we can confirm that both hypotheses H1 and H2 are approved. Thus, we have developed a new classification algorithm named COID-$FCDCM_{GK}$ which is based on both hypotheses. We try to show that if we take into consideration both H1 and H2, the performance of our proposed COID-$FCDCM_{GK}$ algorithm can be improved in comparison to $FCDCM_{GK}$ and COID-DCA which are based on only one hypothesis, H1 and H2 respectively. We approach this by the development of COID-$FCDCM_{GK}$ which applies COID as a maintenance database policy, fuzzy sets to smooth the crisp separation between the DCs contexts and the GK fuzzy clustering technique to generate automatically the parameters of the system.

Since we have shown that the previously developed $FCDCM_{GK}$ and COID-DCA algorithms are no more sensitive to the input class data order, in this Section, we have run the algorithms to unordered data sets to test their classification performance in comparison with COID-$FCDCM_{GK}$. We will compare the three algorithms in terms of specificity, sensitivity, accuracy and execution time. From Table 6.4 and Table 6.5, and in most databases, it is clearly noticed that our COID-$FCDCM_{GK}$ has given good results in terms of the mentioned comparison criteria while outperforming both $FCDCM_{GK}$ and COID-DCA. Thus, we can conclude that it is more appropriate and reasonable to take into consideration both hypotheses rather than only one.

Based on the results presented in Table 6.4, we can notice that COID-$FCDCM_{GK}$ generates better classification results in comparison to COID-DCA and $FCDCM_{GK}$. For instance, by applying our COID-$FCDCM_{GK}$ to the HC database the accuracy of our algorithm is set to 90.76%. However, when applying the COID-DCA to the same database the accuracy of the algorithm is set to 87.77%. From these results, we can conclude that COID-$FCDCM_{GK}$ produces better classification results than COID-DCA which is only based on H2. Indeed, by applying $FCDCM_{GK}$ to the same database the accuracy of the algorithm is set to 89.67%. Again, we can remark that COID-$FCDCM_{GK}$ generates better results than those of the $FCDCM_{GK}$ which is only based on H1. Same remark is noticed for the specificity and the sensitivity criteria. These results confirm that our COID-$FCDCM_{GK}$, which is based on H1 and H2, outperforms the results generated by both COID-DCA and $FCDCM_{GK}$ which are based on only one hypothesis; either H1 or H2.

From Table 6.5, we can also notice that our proposed COID-$FCDCM_{GK}$ algorithm is characterized by its lightweight in terms of running time. In fact, COID-$FCDCM_{GK}$ is characterized by its short time of process in comparison to $FCDCM_{GK}$, but, it needs more time to process than COID-DCA. More precisely, the fact of reducing the size of the original DCA signal database decreases the execution time of our COID-$FCDCM_{GK}$ comparing it to the $FCDCM_{GK}$ one as this latter algorithm is applied to the entire non-maintained database. Nevertheless, COID-$FCDCM_{GK}$ needs more time to process than COID-DCA since it uses two extra components which are the fuzzy set concept and the GK clustering algorithm.

Table 6.4: Experimental Measures

Database	Sensitivity (%)			Specificity (%)			Accuracy (%)		
	$FCDCM_{GK}$	COID-DCA	COID-$FCDCM_{GK}$	$FCDCM_{GK}$	COID-DCA	COID-$FCDCM_{GK}$	$FCDCM_{GK}$	COID-DCA	COID-$FCDCM_{GK}$
SN	85.58	79.27	83.78	79.38	74.22	78.35	82.69	76.92	81.25
Bio	50.94	36.84	47.36	49.05	44.89	48.97	50.00	40.56	48.11
CylB	92.50	93.42	94.73	94.55	93.58	94.87	93.75	93.51	94.81
Ch	94.66	94.36	95.20	94.95	94.89	95.28	94.80	94.61	95.24
IONO	97.77	96.82	98.41	95.23	94.66	96.88	96.86	95.44	97.43
Sck	96.53	96.10	98.26	95.08	94.74	95.98	95.17	94.83	96.12
HC	92.59	91.66	93.98	85.52	82.23	86.18	89.67	87.77	90.76
GC	84.67	83.94	92.33	90.63	89.66	91.73	89.00	88.10	91.90
LR	70.00	80.00	65.00	78.37	83.78	75.67	75.43	82.45	71.92
RWW	98.18	98.04	98.50	98.74	97.56	98.99	98.32	97.90	98.63

For instance, applying the algorithms to the IONO database the amount of time taken by COID-$FCDCM_{GK}$ to process is set to 8.54(s) which is less than the time taken by $FCDCM_{GK}$ which is set to 14.91(s). Nevertheless, COID-$FCDCM_{GK}$ processes a bit longer than COID-DCA which takes 4.06(s) to process.

Table 6.5: Comparison in Terms of Execution Time (s)

Database	Time (s)		
	$FCDCM_{GK}$	COID-DCA	COID-$FCDCM_{GK}$
SN	4.01	1.25	3.11
Bio	3.06	0.24	2.51
CylB	13.02	10.12	11.49
Ch	188.65	104.88	120.72
IONO	14.91	4.06	8.54
Sck	277.23	162.08	231.19
HC	12.98	7.14	8.69
GC	63.88	43.15	49.31
LR	0.63	0.33	0.37
RWW	521.32	326.75	441.08

From these results, we can say that COID-$FCDCM_{GK}$ can be seen as an interesting binary classifier capable of generating satisfactory classification results. We will, also, show that COID-$FCDCM_{GK}$ outperforms other well known classifiers in term of classification accuracy. This will be discussed in what follows.

6.4.5 Comparison with State-of-the-art Recent Methods

It this Section, we will compare the classification results of our COID-$FCDCM_{GK}$ with well known classifiers which are the Support Vector Machine (SVM), the Artificial Neural Network (ANN), the Decision Tree (DTree), the COID-DCA and the FCDCM based GK algorithm. As we are focusing on the classification accuracy of the algorithms, the comparison made is in terms of the average of accuracies on the used databases and is presented in Figure 6.6.

We can notice that Figure 6.6 shows that our developed COID-$FCDCM_{GK}$ algorithm outperforms SVM, ANN, DTree, COID-DCA and FCDCM based GK algorithm in terms of overall accuracy. The major results of this work where published in (Chelly, Smiti, & Elouedi, 2012).

These encouraging COID-$FCDCM_{GK}$ results are explained by the appropriate use of the COID technique for the maintenance of the original DCA signal database. The fact of applying this policy induces a small signal database without redundant or noisy antigens. Eliminating this kind of instances avoids the use of “useless” data while guarantying a maintained signal database for our method. Furthermore, smoothing the crisp separation between the two contexts using fuzzy set theory as well as by generating automatically the parameters of our algorithm using the GK clustering technique avoids the negative influence on the final antigen context affectation. Consequently, we have obtained a novel immune-inspired fuzzy maintained model making the standard DCA a better classifier by generating more reliable classification results.

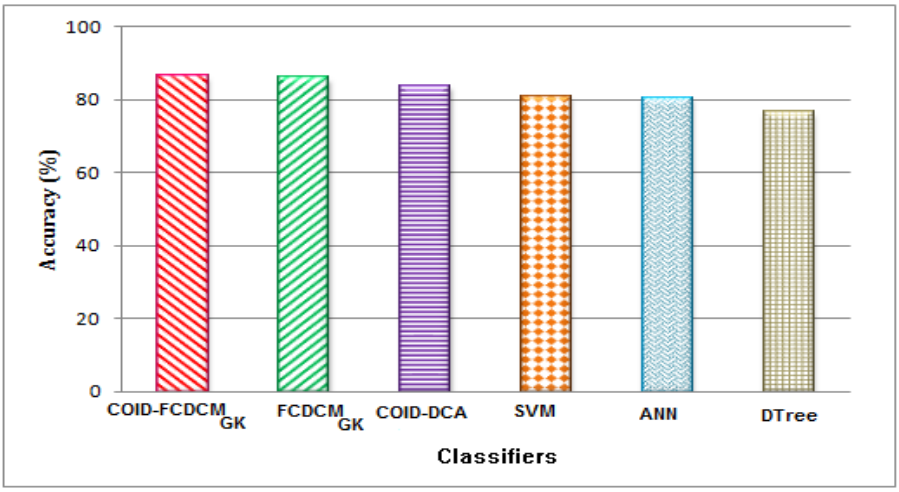

Figure 6.6: Comparison of Classifiers Average Accuracies

6.5 Conclusion

In this Chapter, we have presented a new fuzzy hybrid evolutionary algorithm, COID-$FCDCM_{GK}$, aiming at solving the DCA sensitivity to input class data order. COID-$FCDCM_{GK}$ is based on two hypotheses which were studied and checked. COID-$FCDCM_{GK}$ is a stable classifier which outperforms not only the state-of-the-art classifiers but also the standard version of the DCA as well as the DCA hybrid proposed versions.

Up to now, we have made several investigations on the dendritic cell functioning as well as its algorithmic steps. While investigating the DCA detection and context assessment phases, we have proposed solutions to overcome the DCA sensitivity to the input class data order. While investigating the DCA data pre-processing phase, we have proposed new rough and fuzzy-rough methods leading to a more robust DCA classifiers. Note that the proposed rough and fuzzy-rough DCAs seen in the previous Chapters are applied to ordered data sets only as they are sensitive to the input class data order. Therefore and at the end of this dissertation, we will propose a more general fuzzy-rough, maintained and non-sensitive dendritic cell algorithm. This will be dealt with in the next Chapter.

Chapter 7

The Hybrid Automated and Maintained Fuzzy Dendritic Cell Immune Classifier

7.1 Introduction

The developed non-sensitive COID-$FCDCM_{GK}$ algorithm has shown promising results in terms of classification accuracy. It is based on a deep investigation of its algorithmic steps. Yet, it is important to mention that the COID-$FCDCM_{GK}$ data pre-processing phase is based on the use of the principal component analysis technique. In (Chelly & Elouedi, 2012a, 2012b), we have criticized the fact of using PCA and proposed adequate solutions. Therefore, in this Chapter, we aim to develop an overall automated fuzzy DCA classifier characterized by its non-sensitivity to the input class data order and based on a robust data pre-processing phase.

This Chapter is organized as follows: Section 7.2 describes an analysis of the COID-$FCDCM_{GK}$ algorithm. Section 7.3 details the hybrid rough automated fuzzy DCAs. These classifiers are non-sensitive automated algorithms based on the use of the traditional rough set theory for data pre-processing. Section 7.4 presents the hybrid fuzzy-rough automated fuzzy DCAs. These algorithms are non-sensitive automated classifiers based on the use of the theory of fuzzy rough sets for data pre-processing. Section 7.5 proposes our newly algorithm. Finally, Section 7.6 gives a summary of the whole work.

7.2 Analysis of the COID-$FCDCM_{GK}$ Algorithm

7.2.1 Description of the Current State

To solve the DCA sensitivity to the input class data order, we have first developed $FCDCM_{FCM}$ (Chelly & Elouedi, 2011). $FCDCM_{FCM}$ is based on the use of the fuzzy c-means clustering technique to generate automatically the parameters of the algorithm. Then, we have hypothesized that the effectiveness of FCDCM may depend on the choice

of the right fuzzy clustering technique that should be hybridized with the algorithm. Based on this, we have made a comparative study of different fuzzy clustering techniques and we have shown that the use of the Gustafson-Kessel fuzzy clustering technique is more appropriate for the automatic detection of the FCDCM parameters. $FCDCM_{GK}$ takes into account the fact of alleviating the crisp separation between the two DCs contexts as well as generating automatically the parameters of the system guarantying an overall automated fuzzy algorithm (Chelly & Elouedi, 2015). $FCDCM_{GK}$ was further investigated and it was noticed that if we couple $FCDCM_{GK}$ with a maintenance technique then we will obtain better classification results of the algorithm. That led to the development of the COID-$FCDCM_{GK}$ algorithm (Chelly et al., 2012). However, the data pre-processing phase of the latter algorithm relies on the use of the principal component analysis statistical technique.

Meanwhile, in (Chelly & Elouedi, 2012a, 2012b), we have shown that using PCA is not the most appropriate data pre-processing technique for the DCA. This is because the PCA technique does not guarantee that the first selected principal components that capture most of the variance will be the most adequate features to retain for the DCA classification phase. Besides, applying PCA destroys the underling meaning behind the features present in the used data set which contradicts the specificity of the standard dendritic cell algorithm and as a consequence COID-$FCDCM_{GK}$. Adding to these issues, an informed guess has to be made as to how many variables should be kept for the PCA data reduction process leading to a non overall automated version. As for the signal categorization phase and since it is, also, based on the PCA attributes ranking in terms of variability it could not be considered as a coherent and a reliable categorization procedure. We have shown that using rough set theory for data pre-processing, instead of PCA, can guarantee more reliable classification results of the DCA.

On the other hand, we have shown in (Chelly & Elouedi, 2013c, 2013f, 2013b) that the rough DCA developed algorithms suffer from the quantization step that should be performed before the feature selection phase. This was seen as a kind of an information loss that may influence the algorithms data pre-processing procedure. To solve this issue, we have proposed new DCAs based on fuzzy-rough set theory as a pre-processor (Chelly & Elouedi, 2013c, 2013f, 2013b). In these works, we have shown that fuzzy rough set theory is more appropriate for the DCA data pre-processing phase; specifically for feature selection and signal categorization.

Taking all these facts into consideration, we try to develop an overall hybrid automated algorithm characterized by its non-sensitivity to the input class data order and based on a robust data pre-processing phase. To achieve this, a set of hypotheses have to be checked. This will be discussed in what follows.

7.2.2 Check Points

Our main goal is to mix COID-$FCDCM_{GK}$ with the right data pre-processing technique to ensure a global robust binary classifier. However before this, some works have to be done. In fact, we have proposed several hybridized DCAs based on rough set theory and fuzzy-rough set theory. So, it seems important now to choose the right one to select and to hybridize it with COID-$FCDCM_{GK}$. Yet, it is important to mention that it is not sufficient to draw conclusions from the works presented in (Chelly & Elouedi, 2012a, 2012b, 2013e), detailed in Chapter 3 and Chapter 4, and (Chelly & Elouedi, 2013c, 2013f, 2013b), detailed in Chapter 5 and Chapter 6. This is because in theory there is no specific rule or theorem that can guarantee that if we add the fuzzy component to DCA and hybridize the obtained fuzzy algorithm with rough set theory or fuzzy-rough set theory, the previously obtained results will remain; i.e., results obtained from (Chelly & Elouedi, 2012a, 2012b, 2013e) and (Chelly & Elouedi, 2013c, 2013f,

2013b). Therefore, some extra works have to be done at this level to endorse the results and observations obtained from (Chelly & Elouedi, 2012a, 2012b, 2013e, 2013c, 2013f, 2013b).

To handle the mentioned point and before adding the COID maintenance component, we will mainly propose two sorts of hybridizations. The first hybridization is based on the use of the traditional rough set theory in the $FCDCM_{GK}$ data pre-processing phase. The second hybridization is based on the use of fuzzy-rough set theory in the $FCDCM_{GK}$ data pre-processing phase. We will try to analyze the behavior of these proposed solutions while checking the following set of hypotheses:

- ***First Part of Hybridizations:***
  - **H1:** The use of RST, instead of PCA, for data pre-processing leads to an improvement of the $FCDCM_{GK}$ performance. We will show that RST as a feature selection technique is more adequate to be applied to $FCDCM_{GK}$ than the use of PCA. As for signal categorization, we will show that assigning the same attribute to both SS and PAMP using the RST reduct and core concepts is more adequate than when applying the PCA attributes ranking in terms of variability. We will name the developed method as RST-$FCDCM_{GK}$.
  - **H2:** The categorization of different features to different signals leads to a better performance of the $FCDCM_{GK}$ algorithm. This method is dubbed RC-$FCDCM_{GK}$. We will show that assigning for each selected feature a specific signal category, a process performed by RC-$FCDCM_{GK}$ leads to a better performance than assigning the same attribute to both SS and PAMP; a process performed by RST-$FCDCM_{GK}$.
  - **H3:** The use of a rough set heuristic, the proposed QR-$FCDCM_{GK}$ algorithm, can keep the $FCDCM_{GK}$ characteristic which is its lightweight in terms of processing time. This will be achieved by finding a trade-off between generating satisfactory classification results and preserving the lightweight of the algorithm.
- ***Second Part of Hybridizations:***
  - **H4:** The use of fuzzy-rough set theory instead of rough set theory can solve the limitation related to the information loss due to the data discretization. The developed fuzzy-rough algorithm is named FRST-$FCDCM_{GK}$.
  - **H5:** Instead of generating the associated fuzzy sets and membership functions of each feature, a task performed by FRST-$FCDCM_{GK}$, the use of the fuzzy lower approximation and fuzzy similarity relations would be sufficient. This will avoid all the extra calculations performed by FRST-$FCDCM_{GK}$. The developed method is called FLA-$FCDCM_{GK}$.
  - **H6:** Contrary to FLA-$FCDCM_{GK}$, we will show that if we use both of the fuzzy upper and the fuzzy lower approximations concepts, the selected subset will be more useful as there is less uncertainty concerning objects within the fuzzy boundary region. The developed fuzzy-rough algorithm is named FBR-$FCDCM_{GK}$.

Once conclusions are drawn from these hypotheses, it will be possible to add the COID maintenance technique in order to ensure a robust detection phase for the overall proposed algorithm.

7.3 The Rough $FCDCM_{GK}$ Classifiers

Let us remind that in the third Chapter of this dissertation, we have presented three rough DCA models; namely RST-DCA, RC-DCA and QR-DCA. So to check the validity of the first three hypotheses, H1, H2 and H3, we will hybridize our $FCDCM_{GK}$ fuzzy model, each time, with the data pre-processing basics adopted by these previously proposed rough DCAs. This leads us to develop three hybrid rough $FCDCM_{GK}$ versions which we dub RST-$FCDCM_{GK}$, RC-$FCDCM_{GK}$ and QR-$FCDCM_{GK}$. Details of these algorithms are given in this Section.

7.3.1 Description of the Hybrid Rough $FCDCM_{GK}$ Algorithms

The proposed rough $FCDCM_{GK}$ algorithms are based on two levels. In the top-level, they apply rough set theory to build a solid data pre-processing phase. In the second level, the algorithms apply fuzzy set theory to smooth the crisp separation between the DC's semi-mature and mature contexts. The fact of adding the fuzzy component aims at ensuring the stability of the algorithms.

Concerning the top level of the newly proposed RST-$FCDCM_{GK}$, the algorithm uses the same reasoning as RST-DCA for data pre-processing and as presented in (Chelly & Elouedi, 2012b). In other words and for feature selection, RST-$FCDCM_{GK}$ has to calculated, first, the positive region for the whole conditional attribute set of the antigen input database. Then, RST-$FCDCM_{GK}$ moves to find the reduced set of features, the reduct, which preserves the positive region as the whole conditional attribute set does. This was explained in the third Chapter and more precisely in Section 3.4.1. RST-$FCDCM_{GK}$, also, proposes solutions when the algorithm generates only one reduct or a family of reducts. Once feature selection is achieved, RST-$FCDCM_{GK}$ assigns for each selected feature its right signal category. Just like RST-DCA, RST-$FCDCM_{GK}$ assigns the same feature to both SS and PAMP and the rest of the reduct features to the DS. Based on the obtained reduct and the generated signal base, RST-$FCDCM_{GK}$ will classify each antigen either as being normal or anomalous. To do so, it applies the theory of fuzzy sets to smooth the crisp separation between the antigens two contexts. This was detailed in the fifth Chapter and more precisely in Section 5.4 and as presented in (Chelly & Elouedi, 2011).

The second developed rough $FCDCM_{GK}$ algorithm is named RC-$FCDCM_{GK}$. The algorithm uses the same reasoning as RC-DCA for data pre-processing and as presented in (Chelly & Elouedi, 2012a). More precisely, RC-$FCDCM_{GK}$ follows the same process of feature selection as RST-$FCDCM_{GK}$ but differs from the latter algorithm in the signal categorization phase. RC-$FCDCM_{GK}$ assigns for each selected feature a different signal category; i.e., either a SS, a DS or a PAMP signal. Specifically, RC-$FCDCM_{GK}$ assigns two different features for the safe and PAMP signals; i.e. one feature is assigned to the SS and another feature is used to represent the PAMP signal. The details of this process were highlighted in Section 3.5.2 of Chapter 3. The bottom level of RC-$FCDCM_{GK}$ is based on fuzzy set theory same as RST-$FCDCM_{GK}$.

Both RST-DCA and RC-DCA were criticized in (Chelly & Elouedi, 2013e) because of their extra calculations due to the generation of all the possible reduct sets. Thus, in (Chelly & Elouedi, 2013e), one possible alternative was developed based on the use of the crisp QuickReduct algorithm. Based on the same analogy, both RST-$FCDCM_{GK}$ and RC-$FCDCM_{GK}$ generate all the reduct sets and handle both cases; i.e., when the algorithms generate one reduct and when they generate a family of reducts. This is seen as a time consuming task and therefore we have developed the QR-$FCDCM_{GK}$ algorithm which is based on the crisp QuickReduct algorithm. QR-$FCDCM_{GK}$

follows the same reasoning as QR-DCA (Chelly & Elouedi, 2013e) for feature selection and signal categorization and as previously detailed in Section 3.6 relative to the third Chapter. The bottom level of QR-$FCDCM_{GK}$ is based on fuzzy set theory same as the previous rough $FCDCM_{GK}$ versions.

7.3.2 Results and Discussion

To test the validity of the proposed rough $FCDCM_{GK}$ algorithms, our experiments are performed on the non-ordered two-class databases previously presented in Table 3.2. In this Section, we will test the validity of H1, H2 and H3.

First, we will show that using the theory of rough sets as an information extraction technique, instead of PCA, is more convenient for the $FCDCM_{GK}$ data pre-processing phase as it improves the algorithm classification performance. To do so, we will compare RST-$FCDCM_{GK}$ with PCA'-$FCDCM_{GK}$ and RC-$FCDCM_{GK}$ with PCA-$FCDCM_{GK}$. Both PCA'-$FCDCM_{GK}$ and PCA-$FCDCM_{GK}$ are based on the use of PCA for feature selection. As for signal categorization, PCA'-$FCDCM_{GK}$ processes same as RST-$FCDCM_{GK}$; i.e., it assigns the same feature to SS and PAMP. PCA-$FCDCM_{GK}$ processes same as RC-$FCDCM_{GK}$; i.e., it assigns different features to represent the safe and the PAMP signals.

Secondly, we will show that assigning for each selected feature a specific signal category leads to a better performance than assigning the same attribute to both SS and PAMP. To achieve this, we will compare RC-$FCDCM_{GK}$ with RST-$FCDCM_{GK}$ and PCA-$FCDCM_{GK}$ with PCA'-$FCDCM_{GK}$. This comparison is based on the number of extracted features, specificity, sensitivity and accuracy criteria.

Lastly, we will show that the developed QR- $FCDCM_{GK}$ version can find a trade-off between generating satisfactory classification results and processing in less time than RST-$FCDCM_{GK}$ and RC-$FCDCM_{GK}$. To achieved this, the execution time is added to the previous criteria for the sake of comparison. These notes are confirmed by the results displayed in Table 7.1 and Table 7.2.

Checking the Validity of H1

First of all, from Table 7.1, we can notice that both RC-$FCDCM_{GK}$ and RST-$FCDCM_{GK}$ have the same number of selected features. This is explained by the fact that both models are based on the same feature selection phase. They generate all the possible reducts and choose the one having the smallest set in terms of number of extracted features.

Indeed, from Table 7.1, we can notice that the number of features selected by the rough $FCDCM_{GK}$ approaches is less than the one generated by the standard $FCDCM_{GK}$ when applying PCA; i.e., PCA-$FCDCM_{GK}$ and PCA'-$FCDCM_{GK}$. This can be explained by the appropriate use of RST as an information extraction technique. In fact, RC-$FCDCM_{GK}$ and RST-$FCDCM_{GK}$ by using the reduct concept keep only the most informative features from the whole set of attributes. For instance, when applying RC-$FCDCM_{GK}$ to the CylB data set the number of selected features is only 7 attributes. However, when applying the PCA-$FCDCM_{GK}$ to the same database the number of the retained features is set to 16. We can notice that PCA preserves additional features than necessary which influences the PCA-$FCDCM_{GK}$ classification task by producing less accuracy in comparison to the RC-$FCDCM_{GK}$ results.

Table 7.1: Comparison Results of the $FCDCM_{GK}$ PCA and RST Based Approaches

Database	♯ Attributes				Sensitivity (%)				Specificity (%)				Accuracy (%)			
	$FCDCM_{GK}$				$FCDCM_{GK}$				$FCDCM_{GK}$				$FCDCM_{GK}$			
	PCA	RC	PCA'	RST	PCA	RC	PCA'	RST	PCA	RC	PCA'	RST	PCA	RC	PCA'	RST
SN	28	20	28	20	89.18	91.89	85.58	90.09	83.50	95.87	79.38	85.56	86.53	93.75	82.69	87.98
Bio	24	19	24	19	56.60	83.01	50.94	66.03	54.71	84.90	49.05	58.49	55.66	83.96	50.00	62.26
CylB	16	7	16	7	96.00	98.00	92.50	97.00	96.79	98.39	94.55	97.43	96.48	98.24	93.75	97.26
Ch	14	11	14	11	95.26	98.92	94.66	98.20	95.61	99.60	94.95	99.21	95.43	99.24	94.80	98.68
IONO	24	19	24	19	95.23	96.82	94.44	96.03	97.77	98.66	95.55	98.22	96.86	98.00	95.15	97.43
Sck	22	20	22	20	97.83	98.26	96.53	97.40	97.03	97.99	95.08	96.89	97.08	98.01	95.17	96.92
HC	19	14	19	14	94.90	98.14	92.59	96.75	88.81	96.05	85.52	92.10	92.39	97.28	89.67	94.83
GC	17	17	17	17	90.87	92.33	84.67	87.95	92.28	92.69	90.63	90.35	91.90	92.60	89.00	89.70
LR	10	5	10	5	90.00	95.00	70.00	90.00	91.89	97.29	78.37	94.59	91.22	96.49	75.43	92.98
RWW	10	6	10	6	99.18	99.63	98.18	99.26	99.12	99.56	98.74	99.31	99.16	99.61	98.32	99.27

On the other hand, RC-$FCDCM_{GK}$ based on the reduct concept selects the minimal set of features from the original database and can guarantee that the reduct attributes will be the most relevant for the algorithm classification task.

From Table 7.1, we notice that the classification accuracy of RC-$FCDCM_{GK}$ is notably better than the one given by PCA-$FCDCM_{GK}$ for almost all the data sets. We can, also, notice that the RST-$FCDCM_{GK}$ classification results are better than the ones given by PCA'-$FCDCM_{GK}$. Same remark is noticed for both the sensitivity and the specificity criteria. For instance, when applying the RST-$FCDCM_{GK}$ to the CylB database the classification accuracy is set to 97.26%. However, when applying the PCA'-$FCDCM_{GK}$ to the same database the accuracy is set to 93.75%. When applying the RC-$FCDCM_{GK}$ algorithm to the same data set, the classification accuracy of the algorithm is set to 98.42%. However, when applying the PCA-$FCDCM_{GK}$ to the same database the accuracy is set to 96.48%. Thus, from this part, we can conclude that the rough information extraction technique is an interesting pre-processor that would be better applied to $FCDCM_{GK}$, instead of applying PCA. From these results, H1 is proved to be true.

Checking the Validity of H2

From Table 7.1, we can clearly see that in all databases RC-$FCDCM_{GK}$ and PCA-$FCDCM_{GK}$ outperform the classification accuracy generated by RST-$FCDCM_{GK}$ and PCA'-$FCDCM_{GK}$. For instance, the classification accuracies of RST-$FCDCM_{GK}$ and PCA'-$FCDCM_{GK}$ when applied to the IONO database are set to 97.43% and 95.15%, respectively, which are both less than 98.00% and 96.86% generated respectively by RC-$FCDCM_{GK}$ and PCA-$FCDCM_{GK}$. This is explained by the fact that RST-$FCDCM_{GK}$ and PCA'-$FCDCM_{GK}$ differ from both RC-$FCDCM_{GK}$ and PCA-$FCDCM_{GK}$ in the signal categorization phase. More precisely, both RC-$FCDCM_{GK}$ and PCA-$FCDCM_{GK}$ assign different features to different signal categories; i.e., a specific feature is assigned for each of the SS and the PAMP signals. However, RST-$FCDCM_{GK}$ and PCA'-$FCDCM_{GK}$ use the same attribute to assign it for both SS and PAMP. From these results, we can conclude that it is crucial to assign for each signal category a specific and different feature and thus, H2 is proved to be true.

Checking the Validity of H3

Now, in order to check the validity of H3 we will focus on the results obtained from QR-$FCDCM_{GK}$. From Table 7.2, we can notice that QR-$FCDCM_{GK}$ can find a trade-off between generating satisfactory classification results and processing in less time than both RST-$FCDCM_{GK}$ and RST-$FCDCM_{GK}$. QR-$FCDCM_{GK}$ has either the same number of features as RST-$FCDCM_{GK}$ and RST-$FCDCM_{GK}$ or more features. This is explained by the fact that QR-$FCDCM_{GK}$, by applying the QuickReduct algorithm, follows the features path that generates the highest dependency degree. Consequently, the taken path may either lead to a final reduct including the smallest number of features or to a path including more selected features. For instance, applying QR-$FCDCM_{GK}$ to the IONO database the number of selected attributes is set to 22. However, when applying RST-$FCDCM_{GK}$ or RC-$FCDCM_{GK}$ the number of selected features is set to 19. Applying the rough $FCDCM_{GK}$ models to the Ch database the number of the selected features is the same: 11.

Table 7.2: Comparison Results of the Rough $FCDCM_{GK}$ Versions

Database	♯ Attributes			Sensitivity (%)			Specificity (%)			Accuracy (%)			Time (s)		
	$FCDCM_{GK}$			$FCDCM_{GK}$			$FCDCM_{GK}$			$FCDCM_{GK}$			$FCDCM_{GK}$		
	RC	QR	RST	RC	QR	RST	RC	QR	RST	RC	QR	RST	RC	QR	RST
SN	20	22	20	91.89	90.99	90.09	95.87	93.81	85.56	93.75	92.30	87.98	1717.21	10.03	1695.96
Bio	19	19	19	83.01	83.01	66.03	84.90	84.90	58.49	83.96	83.96	62.26	1699.34	9.56	1617.02
CylB	7	7	7	98.00	98.00	97.00	98.39	98.39	97.43	98.24	98.24	97.26	1458.06	14.61	1426.73
Ch	11	11	11	98.92	98.92	98.20	99.60	99.60	99.21	99.24	99.24	98.68	1791.44	586.93	1700.10
IONO	19	22	19	96.82	97.61	96.03	98.66	97.77	98.22	98.00	97.72	97.43	679.68	19.81	599.42
Sck	20	22	20	98.26	97.83	97.40	97.99	97.96	96.89	98.01	97.95	96.92	1432.81	529.17	1401.70
HC	14	14	14	98.14	98.14	96.75	96.05	96.05	92.10	97.28	97.28	94.83	268.67	23.75	94.29
GC	17	17	17	92.33	92.33	87.95	92.69	92.69	90.35	92.60	92.60	89.70	540.45	141.09	507.87
LR	5	5	5	95.00	95.00	90.00	97.29	97.29	94.59	96.49	96.49	92.98	0.84	0.64	0.81
RWW	6	8	6	99.63	99.40	99.26	99.56	99.62	99.31	99.61	99.46	99.27	2270.16	1032.01	2251.47

In terms of classification accuracy, we can notice that the difference between the classification accuracies generated by both QR-$FCDCM_{GK}$ and RC-$FCDCM_{GK}$ is not significant. Thus, we can say that QR-$FCDCM_{GK}$ has nearly the same classification performance as RC-$FCDCM_{GK}$. In some databases, the QR-$FCDCM_{GK}$ classification accuracy is a bit less than the one generated by RC-$FCDCM_{GK}$. We also remark, that in all databases, QR-$FCDCM_{GK}$ and RC-$FCDCM_{GK}$ outperform the classification accuracy generated by RST-$FCDCM_{GK}$. This is explained by the fact that RST-$FCDCM_{GK}$ differs from QR-$FCDCM_{GK}$ and RC-$FCDCM_{GK}$ in the signal categorization phase.

The main advantage of QR-$FCDCM_{GK}$ is that it takes less time to process than both RST-$FCDCM_{GK}$ and RC-$FCDCM_{GK}$. For example, when applying the algorithms to the CylB database the amount of time taken by QR-$FCDCM_{GK}$ to process is 14.61(s) which is less than the time taken by both RC-$FCDCM_{GK}$ and RST-$FCDCM_{GK}$ which is 1458.06(s) and 1426.73(s), respectively. The QR-$FCDCM_{GK}$ lightweight in terms of running time is explained by the advantage of using the QuickReduct algorithm as it attempts to calculate a reduct without exhaustively generating all possible subsets.

In contrast, RST-$FCDCM_{GK}$ and RC-$FCDCM_{GK}$ generate all possible subsets and retrieve those with a maximum rough set dependency degree. This is an expensive solution and all the calculations involved in discovering all the reduct sets are pointless. These results endorse the results obtained in (Chelly & Elouedi, 2012b, 2012a, 2013e) and prove that H3 is true. The major results of this part were published in (Chelly & Elouedi, 2013d, 2014a).

To summarize, QR-$FCDCM_{GK}$ can be seen as an interesting classification technique proposed as an alternative to RST-$FCDCM_{GK}$ and RC-$FCDCM_{GK}$. QR-$FCDCM_{GK}$ is capable of generating better classification results than RST-$FCDCM_{GK}$ and RC-$FCDCM_{GK}$ while having a lightweight in terms of running time.

Since we have endorsed the results obtained from (Chelly & Elouedi, 2012b, 2012a, 2013e), it seems necessary now to check if we will have the same observations noted in (Chelly & Elouedi, 2013c, 2013f, 2013b) when we use the theory of fuzzy-rough set with $FCDCM_{GK}$. This will be discussed in the next Section.

7.4 The Fuzzy-Rough $FCDCM_{GK}$ Classifiers

Previously in the fourth Chapter of this dissertation we have presented three fuzzy-rough DCAs; namely FRST-DCA, FLA-DCA and FBR-DCA. So to check the validity of the rest of the hypotheses, H4, H5 and H6, we will hybridize our $FCDCM_{GK}$ algorithm, each time, with the data pre-processing basics adopted by these previously proposed fuzzy-rough DCAs. This leads us to develop the FRST-$FCDCM_{GK}$, the FLA-$FCDCM_{GK}$ and the FBR-$FCDCM_{GK}$ algorithms. Details of these algorithms are given in this Section.

7.4.1 Description of the Hybrid Fuzzy-Rough $FCDCM_{GK}$ Algorithms

Same as the rough $FCDCM_{GK}$ models, the fuzzy-rough $FCDCM_{GK}$ algorithms are based on two levels as well. In the top-level, the proposed algorithms apply fuzzy-rough set theory to build a solid data pre-processing phase. This aims at coping with the information loss caused by the data quantization step performed by the rough $FCDCM_{GK}$

models before feature selection. In the second level, the algorithms apply fuzzy set theory to ensure the stability of the obtained classifiers.

FRST-$FCDCM_{GK}$ uses the same reasoning as FRST-DCA for data pre-processing and as presented in (Chelly & Elouedi, 2013c). In other words and for feature selection, FRST-$FCDCM_{GK}$ applies the Fuzzy-Rough QuickReduct algorithm. As for signal categorization, the algorithm is based on the fuzzy dependency degree concept. This was explained in the fourth Chapter and more precisely in Section 4.3.

The second fuzzy-rough $FCDCM_{GK}$ developed algorithm is named FLA-$FCDCM_{GK}$. The algorithm uses the same reasoning as FLA-DCA for data pre-processing (Chelly & Elouedi, 2013f). More precisely, FLA-$FCDCM_{GK}$ uses the fuzzy lower approximation concept to select features and to categorize them into their right signal types. The details of this process were highlighted in Section 4.4 of Chapter 4.

Now based on both the fuzzy lower and fuzzy upper approximations, the FBR-$FCDCM_{GK}$ was developed. Same as FBR-DCA (Chelly & Elouedi, 2013b), FBR-$FCDCM_{GK}$ uses the concept of the boundary region to select features and then to assign for each attribute its most convenient signal category. Both of these processes were detailed in Section 4.5 relative to the fourth Chapter. The bottom level of the fuzzy-rough $FCDCM_{GK}$ algorithms is based on fuzzy set theory same as the previous rough $FCDCM_{GK}$ versions.

7.4.2 Results and Discussion

In this Section, we will test the validity of H4, H5 and H6. First, we will show that using the theory of fuzzy-rough sets, instead of RST, is more convenient for the $FCDCM_{GK}$ data pre-processing phase as it avoids the information loss due to data discretization. To do so, we will compare the fuzzy-rough $FCDCM_{GK}$ algorithms with QR-$FCDCM_{GK}$.

Secondly, we will show that the fact of using the fuzzy lower approximation instead of generating the membership functions of each feature, a task performed by FRST-$FCDCM_{GK}$, is more appropriate as all these extra calculations may be avoided. To achieve this, we will compare FLA-$FCDCM_{GK}$ with FRST-$FCDCM_{GK}$.

Lastly, we will show that we may use the boundary region instead of the fuzzy lower approximation to perform feature selection. The fact of using both of the fuzzy lower and fuzzy upper approximations may be more adequate as the selected subset will be more useful. This is because this set has less uncertainty within the fuzzy boundary region. In this part will we compare the FBR-$FCDCM_{GK}$ with the FLA-$FCDCM_{GK}$ algorithm. These notes are confirmed by the results displayed in Table 7.3 and Table 7.4.

Checking the Validity of H4

From Table 7.3 and in terms of number of selected features, we can notice that the fuzzy-rough $FCDCM_{GK}$ algorithms have fewer features than the rough QR-$FCDCM_{GK}$. This is explained by the fact that the $FCDCM_{GK}$ algorithms incorporates the information usually lost in crisp discretization.

Table 7.3: Comparison Results of the Fuzzy-Rough $FCDCM_{GK}$ Algorithms

Database	♯ Attributes				Sensitivity (%)				Specificity (%)				Accuracy (%)			
	$FCDCM_{GK}$				$FCDCM_{GK}$				$FCDCM_{GK}$				$FCDCM_{GK}$			
	FRST	FLA	FBR	QR	FRST	FLA	FBR	QR	FRST	FLA	FBR	QR	FRST	FLA	FBR	QR
SN	16	10	9	22	96.39	97.29	98.19	90.99	96.90	97.93	98.96	93.81	96.63	97.59	98.55	92.30
Bio	13	9	9	19	86.79	90.56	90.56	83.01	90.56	94.33	94.33	84.90	88.67	92.45	92.45	83.96
CylB	7	5	5	7	99.00	98.50	98.50	98.00	99.36	98.71	98.71	98.39	99.22	98.63	98.63	98.24
Ch	4	4	4	11	99.52	99.65	99.78	98.92	99.54	99.67	99.80	99.60	99.53	99.65	99.76	99.24
IONO	9	9	9	22	97.62	99.20	99.20	97.61	99.11	99.55	99.55	97.77	98.57	99.43	99.43	97.72
Sck	16	14	14	22	98.26	97.40	96.96	97.83	99.68	99.46	99.40	97.96	99.60	99.33	99.25	97.95
HC	7	6	5	14	99.53	99.07	98.61	98.14	99.34	98.68	98.02	96.05	99.45	98.91	98.36	97.28
GC	17	11	10	17	92.33	84.67	90.87	92.33	92.69	90.63	92.28	92.69	92.60	89.00	91.90	92.60
LR	5	5	5	5	95.00	95.00	95.00	95.00	97.29	97.29	97.29	97.29	96.49	96.49	96.49	96.49
RWW	4	3	3	8	99.81	99.61	99.61	99.40	99.62	99.18	99.18	99.62	99.76	99.50	99.50	99.46

The fuzzy-rough $FCDCM_{GK}$ algorithms perform much better than the rough version on the whole; i.e., in terms of both feature selection and classification quality. For instance, applying FRST-$FCDCM_{GK}$, FLA-$FCDCM_{GK}$ and FBR-$FCDCM_{GK}$ to the Bio database the number of selected attributes is set to 13, 9 and 9, respectively. However, when applying QR-$FCDCM_{GK}$ to the same database the number of selected features is set to 19.

Furthermore, from Table 7.3, we can notice that the fuzzy-rough $FCDCM_{GK}$ algorithms outperform QR-$FCDCM_{GK}$ in terms of classification accuracy. For instance, when applying the algorithms to the Sck data set the classification accuracy of FRST-$FCDCM_{GK}$ FLA-$FCDCM_{GK}$ and FBR-$FCDCM_{GK}$ are set to 99.60%, 99.33%, and 99.25% . However, when applying QR-$FCDCM_{GK}$ to the same database the accuracy is set to 97.95%. Same remark is observed for the specificity and the sensitivity criteria.

From Table 7.4 and when comparing the results in terms of running time, we can notice that the fuzzy-rough $FCDCM_{GK}$ algorithms need more time to process than QR-$FCDCM_{GK}$. This is due to the attachment of the fuzzy component.

Table 7.4: Comparison Results in Terms of Running Time (s)

Database	Time (s)			
	$FCDCM_{GK}$			
	FRST	FLA	FBR	QR
SN	111.33	11.61	28.41	10.03
Bio	53.11	10.99	17.09	9.56
CylB	124.11	21.99	32.69	14.61
Ch	1061.02	692.90	722.13	586.93
IONO	138.71	33.98	47.15	19.81
Sck	968.38	612.67	725.99	529.17
HC	199.70	25.99	45.06	23.75
GC	348.13	169.76	210.04	141.09
LR	0.79	0.691	0.72	0.64
RWW	1846.66	1301.45	1653.87	1032.01

From these results we can confirm that the fact of applying FRST, instead of RST, to the $FCDCM_{GK}$ data pre-processing phase is more convenient leading to a better performance. Consequently, H4 is proved to be true.

Checking the Validity of H5 and H6

From Table 7.3, we can notice that FLA-$FCDCM_{GK}$ and FBR-$FCDCM_{GK}$ can find smaller subsets than FRST-$FCDCM_{GK}$ in general. For instance, applying FLA-$FCDCM_{GK}$ and FBR-$FCDCM_{GK}$ to the GC database the number of selected features is set to 11 and 10, respectively. Nevertheless, applying FRST-$FCDCM_{GK}$ to the same data set the number of selected features is set to 17. The fuzzy boundary-region-based method, FBR-$FCDCM_{GK}$, finds smaller or equally sized subsets than the lower approximation method, FLA-$FCDCM_{GK}$. This is to be expected, as FBR-$FCDCM_{GK}$ includes fuzzy upper approximation information in addition to that of the fuzzy lower approximation. For instance, applying FBR-$FCDCM_{GK}$ to the SN data set the number of selected feature is set

to 9. However, when applying FLA-$FCDCM_{GK}$ to the same data set the number of selected features is set to 10. Applying both algorithms to the CylB data set the number of selected features is the same and set to 5 attributes.

From Table 7.4, we can notice that FLA-$FCDCM_{GK}$ and FBR-$FCDCM_{GK}$ are quicker than FRST-$FCDCM_{GK}$ in terms of running time. This is explained by the fact that FRST-$FCDCM_{GK}$ based on fuzzy-rough sets has to generate, first, the fuzzy sets as well as the membership functions from the used database and, second, has to determine the fuzzy equivalence classes for each conditional attribute set. However, the complexity related to these calculations becomes prohibitively high for large feature subsets.

On the other hand, both FLA-$FCDCM_{GK}$ and FBR-$FCDCM_{GK}$ avoid the extra calculations performed by FRST-$FCDCM_{GK}$ as they are both based on the use of fuzzy similarity relations. This ensures that for one subset, only one fuzzy similarity relation is used to compute the fuzzy lower approximation. Indeed, from Table 7.4, we can remark that FBR-$FCDCM_{GK}$ needs more time to process than FLA-$FCDCM_{GK}$. This is because FBR-$FCDCM_{GK}$ has to calculate, for each evaluation, both the fuzzy lower and upper approximations, and hence, the calculation of the fuzzy boundary region is more costly than that of the fuzzy lower approximation alone. Besides, we can notice that all of the fuzzy-rough-$FCDCM_{GK}$ algorithms perform similarly with classification accuracy; improving or remaining the same for most data sets. FRST-DCA performs equally well; however, this is at the cost of extra features and extra time required to find the fuzzy reducts. These observations and notes endorse the results obtained in (Chelly & Elouedi, 2013c, 2013f, 2013b) and prove that H5 and H6 are true. The major results of this part were published in (Chelly & Elouedi, 2013a, 2014c).

At the end of this part, we could affirm the trueness of all hypotheses. Accordingly, we can say that if we add an extra fuzzy component to the rough or to the fuzzy-rough DCAs then the previously drown results will remain; i.e., as if without that extra component. Now, it is possible to focus on the algorithms detection phase by maintaining the generated signal base. This will be achieved by applying the COID maintenance technique.

7.5 The COID-FLA-$FCDCM_{GK}$ Immune Classifier

In this Section, we will detail the overall automated and maintained developed algorithm. We will give its architecture, the details of its algorithmic steps, finally we will highlight its characteristics as a binary classifier.

7.5.1 COID-FLA-$FCDCM_{GK}$ Description

Which Data Pre-processing Methodology Should We Choose?

Up to now, we have shown that the fuzzy-rough $FCDCM_{GK}$ algorithms perform well in terms of classification accuracy in comparison to the rough $FCDCM_{GK}$ versions. We have, also, shown that the fuzzy-rough $FCDCM_{GK}$ algorithms perform similarly in terms of classification accuracy. However, they differ at the cost of extra features and extra time required to process. Besides, we have shown that FLA-$FCDCM_{GK}$ needs less time to process in comparison to FBR-$FCDCM_{GK}$ and FRST-$FCDCM_{GK}$. Therefore, we will choose FLA-$FCDCM_{GK}$ as a binary classifier for further investigations.

COID-FLA-$FCDCM_{GK}$ Architecture

The FLA-$FCDCM_{GK}$ algorithm is characterized by its robust data pre-processing step and being stable as a binary classifier. Yet, it is important to shed light on the algorithm context assessment phase. Let us remind that in (Chelly et al., 2012), we have shown that it is important to maintain the DCA generated signal base. To achieve this, the standard DCA version was hybridized with the COID maintenance technique leading to an improvement of the DCA classification results. Therefore, in what is next we will focus on the FLA-$FCDCM_{GK}$ detection phase and couple the latter algorithm with the COID maintenance technique. The developed algorithm is dubbed COID-FLA-$FCDCM_{GK}$. A global view of the COID-FLA-$FCDCM_{GK}$ architecture is presented in Figure 7.1.

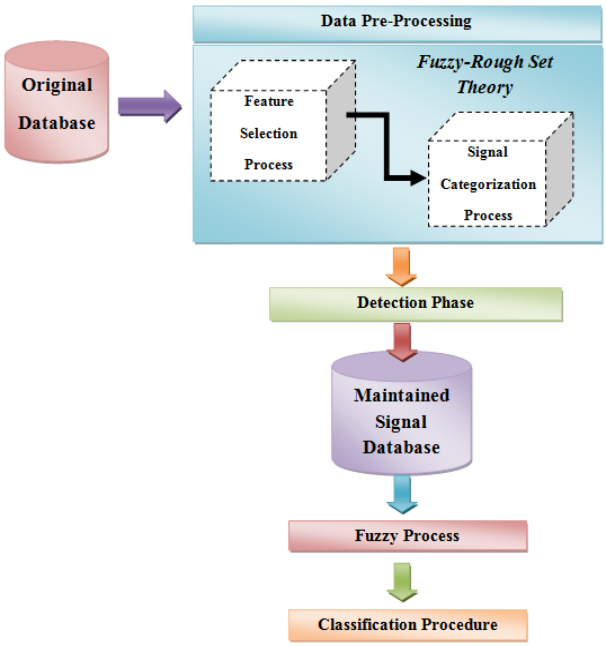

Figure 7.1: The COID-FLA-$FCDCM_{GK}$ Architecture

7.5.2 COID-FLA-$FCDCM_{GK}$ Algorithmic Details

COID-FLA-$FCDCM_{GK}$ attempts to classify each data item, seen as an antigen, either as normal or as anomalous. The COID-FLA-$FCDCM_{GK}$ is not only a classification algorithm. It, also, allows to know how anomalous an antigen may be. This is achieved by the generation of an anomaly coefficient value; i.e., the Mean Mature Context Antigen Value (MCAV). A generic form of the COID-FLA-$FCDCM_{GK}$ is described by Algorithm 7.1 and is given in this Section. For the ease of analysis, the algorithm is divided into four main phases as presented in the numerated list and which will be discussed separately in what follows.

1. *Fuzzy-Rough data Pre-processing* - Line 1 to Line 5;
2. *COID-Detection phase* - Line 6 to Line 12;
3. *Fuzzy Context Assessment phase* - Line 13 to Line 18;
4. *Classification phase* - Line 19 to Line 26;

Algorithm 7.1 The COID-FLA-$FCDCM_{GK}$ Algorithm

```
input: signals from all categories and antigens;
output: antigens plus context values;
for each DC do                                  /* Fuzzy-Rough data Pre-processing */
   initialize-DC();
end for
while CSM output signal < migration threshold do        /* COID-Detection phase */
   get-antigens();
   get-signals();
   calculate-inter();
   update-cumul();
end while
signal-base-maintenance();
Fuzzy-System-Construction();                    /* Fuzzy Context Assessment phase */
if middle-of-Maturity-output > centroid-value then
   cell-context = 0;
else
   cell-context = 1;
end if
for each antigen do                             /* Classification phase */
   if cell-context == 1 then
      Nb-mature ++;
   end if
end for
for each antigen do
   MCAV = Nb-mature / Nb-antigen ;
end for
```

COID-FLA-$FCDCM_{GK}$ Fuzzy-Rough Data Pre-Processing Phase

Same as the standard DCA version, COID-FLA-$FCDCM_{GK}$ has two types of input data which are signals from all categories and antigens. Signals are represented as vectors of real-valued numbers while antigens are the data item IDs. The COID-FLA-$FCDCM_{GK}$, as a binary classifier, has to classify each antigen either as normal or as anomalous. Thus, the algorithm output is the antigen context which is represented by the binary value 0 to mean that the antigen is likely to be normal, or 1 meaning that the antigen is likely to be anomalous.

Formally and through the *Fuzzy-Rough data pre-processing phase*, COID-FLA-$FCDCM_{GK}$ performs data pre-processing where feature selection and signal categorization are achieved. COID-FLA-$FCDCM_{GK}$ selects the most important features from the input training data set and assigns each selected attribute to its specific signal category; i.e., either as SS, as DS or as PAMP. To achieve this, COID-FLA-$FCDCM_{GK}$ applies fuzzy-rough set theory and more precisely the algorithm uses the fuzzy lower approximation concept as previously explained in (Chelly & Elouedi, 2013f). This task is achieved via the ***initialize-DC()*** function.

COID-FLA-$FCDCM_{GK}$ COID-Detection Phase

Throughout the *COID-Detection phase*, COID-FLA-$FCDCM_{GK}$ has to generate a signal database by combining the input signals with the antigens. This combination is achieved via the use of both ***get-antigens()*** and ***get-signals()*** functions. The induced signal database rows represent the antigens to classify and the attributes represent SS, PAMP and DS. The attribute values, for each antigen, are calculated based on specific equations (Greensmith, 2007). Formally, this combination is achieved through using a population of artificial DCs. Using multiple DCs means that multiple antigens are sampled multiple times.

Based on the induced signal database, the algorithm processes its input signals to get three cumulative output signal values known as the costimulatory molecule signal value (CSM), the semi-mature signal value (*smDC*) and the mature signal value (*mDC*). This task is performed through the use of the ***calculate-inter()*** function where these cumulative output signal values are calculated using a signal processing equation (Greensmith, 2007) and a set of weights. These three DC output signals perform two roles which are, first, to determine if an antigen type is anomalous and, second, to limit the time spent sampling data. In fact, each DC in the population is assigned a migration threshold value (*mt*) upon its creation. So, if the value of CSM exceeds *mt* then the DC stops sampling antigens and signals; else the algorithm continues sampling and, also, keeps calculating and updating the values of CSM, *smDC* and *mDC* via the ***update-cumul()*** function.

Once the signal base is ready for the algorithm next steps, the base has to be maintained. To do so, the COID maintenance technique is applied through the use of the ***signal-base-maintenance()*** function. A worked example of this process was previously presented in Chapter 6 and more precisely in Section 6.3.3. The output of this phase is, thus, a new maintained and cleaned signal base lacking redundant and misleading antigens.

COID-FLA-$FCDCM_{GK}$ Fuzzy Context Assessment Phase

Now, through the *fuzzy context assessment phase* the fuzzy part has to be constructed using the ***Fuzzy-System-Construction()*** function. This function performs three main roles. First, it aims at defining the linguistic variables, their ranges and at automatically generating the extents and midpoints of the variables using the Gustafson-Kessel Algorithm. Secondly, the function aims at constructing the rule base. Lastly, it aims at generating the centroid value for each antigen. At this stage, the DC forms a cell context that is used to perform antigen classification. In fact, a comparison is made between the middle-of-Maturity-output and the centroid-value and the one that has a greater/higher output is the one that becomes the cell context; i.e., either 1 or 0. This information is ultimately used in the generation of an anomaly coefficient which will be dealt with in the final step; the *classification phase*.

COID-FLA-$FCDCM_{GK}$ Classification Phase

The calculted value for the cell context is used to derive the nature of the response by measuring the number of DCs that are fully mature. This generated number is represented by the Mature Context Antigen Value ($MCAV$). The $MCAV$ is calculated by dividing the number of times an antigen appears in the mature context, *Nb-mature*, by the total number of presentation of that antigen, *Nb-antigen*. Once the $MCAV$ is calculated for each antigen, the algorithm can perform its classification task. This is done by comparing the $MCAV$ of each antigen to an anomalous threshold. More precisely, those antigens whose $MCAVs$ are greater than the anomalous threshold are classified

into the anomalous category while the others are classified into the normal one.

7.5.3 Experimental Setup, Results and Analysis

To test the performance of our proposed hybrid automated and maintained COID-FLA-$FCDCM_{GK}$ immune classifier, we will compare it classification results to its basic versions. This is to note the worth values added to the obtained general algorithm. In this Section, we will compare COID-FLA-$FCDCM_{GK}$ with PCA-$FCDCM_{GK}$, COID-PCA-$FCDCM_{GK}$ and FLA-$FCDCM_{GK}$. Results of these comparisons are highlighted in Table 7.5 and Table 7.6. Once the advantages of COID-FLA-$FCDCM_{GK}$ are discerned, we will further compare its classification results to the standard DCA version, to the SVM, the ANN and the DTree classifiers.

Comparison of COID-FLA-$FCDCM_{GK}$ with Its Basic Versions

Starting the comparison between PCA-$FCDCM_{GK}$ and COID-PCA-$FCDCM_{GK}$, from Table 7.5 and Table 7.6 we can notice that when maintaining the signal base, in most data sets, the classification accuracy of COID-PCA-$FCDCM_{GK}$ is better than the one generated by PCA-$FCDCM_{GK}$. Indeed, in terms of running time, COID-PCA-$FCDCM_{GK}$ has the advantage of running in less time than PCA-$FCDCM_{GK}$. This is because the algorithm uses the reduced cleaned signal base instead of the initial one.

From Table 7.5 and Table 7.6 and when comparing both PCA-$FCDCM_{GK}$ and FLA-$FCDCM_{GK}$, we can note that applying fuzzy-rough set theory instead of PCA is more convenient for the $FCDCM_{GK}$ data pre-processing phase. However, this is with an extra cost in terms of running time. This is due to the attachment of the fuzzy rough set component for feature selection and signal categorization.

Most importantly and from Table 7.5, we can notice that the highest classification results are observed when applying the COID-FLA-$FCDCM_{GK}$ algorithm. For instance, when applying the different algorithms to the GC data set the classification accuracy of COID-FLA-$FCDCM_{GK}$ is set to 96.10% which is higher than the classification accuracies generated by PCA-$FCDCM_{GK}$, COID-PCA-$FCDCM_{GK}$ and FLA-$FCDCM_{GK}$ which are set to 89.00%, 91.90% and 89.00%, respectively. Same remark is noticed for both the sensitivity and the specificity criteria.

Besides and based on the results obtained from Table 7.6, it is important to mention that COID-FLA-$FCDCM_{GK}$ needs more time to process than both PCA-$FCDCM_{GK}$ and COID-PCA-$FCDCM_{GK}$. COID-FLA-$FCDCM_{GK}$ processes longer than PCA-$FCDCM_{GK}$, first, because the data pre-processing phase of COID-FLA-$FCDCM_{GK}$ is based on the FRST concepts and, second, due the use of the COID maintenance technique. On the other hand, COID-FLA-$FCDCM_{GK}$ processes longer than COID-PCA-$FCDCM_{GK}$ because the data pre-processing phase of COID-FLA-$FCDCM_{GK}$ is based on the FRST concepts instead of PCA.

Table 7.5: Comparison Results of the COID-FLA-$FCDCM_{GK}$ and Its Basic Approaches

Database	Sensitivity (%)				Specificity (%)				Accuracy (%)			
	$FCDCM_{GK}$				$FCDCM_{GK}$				$FCDCM_{GK}$			
	PCA	COID-PCA	FLA	COID-FLA	PCA	COID-PCA	FLA	COID-FLA	PCA	COID-PCA	FLA	COID-FLA
SN	85.58	83.78	97.29	96.39	79.38	78.35	97.93	96.90	82.69	81.25	97.59	96.63
Bio	50.94	47.36	90.56	89.47	49.05	48.97	94.33	93.87	50.00	48.11	92.45	91.50
CylB	92.50	94.73	98.50	98.68	94.55	94.87	98.71	99.35	93.75	94.81	98.63	99.07
Ch	94.66	95.20	99.65	99.76	94.95	95.28	99.67	99.93	94.80	95.24	99.65	99.84
IONO	94.44	98.41	99.20	98.41	95.55	96.88	99.55	99.11	95.15	97.43	99.43	98.86
Sck	96.53	98.26	97.40	98.26	95.08	95.98	99.46	99.88	95.17	96.12	99.33	99.78
HC	92.59	93.98	99.07	99.34	85.52	86.18	98.68	99.53	89.67	90.76	98.91	99.45
GC	84.67	92.33	84.67	94.89	90.63	91.73	90.63	96.55	89.00	91.90	89.00	96.10
LR	70.00	65.00	95.00	90.00	78.37	75.67	97.29	94.59	75.43	71.92	96.49	92.98
RWW	98.18	98.50	99.61	99.85	98.74	98.99	99.18	99.68	98.32	98.63	99.50	99.81

Meanwhile, COID-FLA-$FCDCM_{GK}$ needs less time to process than FLA-$FCDCM_{GK}$. This is explained by the fact that COID-FLA-$FCDCM_{GK}$ is applied to a reduced signal base which is generated via the use of the COID technique.

Table 7.6: Comparison in Terms of Running Time (s)

Database	Time (s)			
	$FCDCM_{GK}$			
	PCA	COID-PCA	FLA	COID-FLA
SN	4.01	3.11	11.61	10.15
Bio	3.06	2.51	10.99	10.04
CylB	13.02	11.49	21.99	17.24
Ch	188.65	120.72	692.9	361.03
IONO	11.76	8.54	33.98	19.62
Sck	277.23	231.19	612.67	403.98
HC	12.98	8.69	25.99	16.28
GC	63.88	49.31	169.76	132.83
LR	0.63	0.37	0.69	0.42
RWW	521.32	441.08	1301.45	881.23

Comparison of COID-FLA-$FCDCM_{GK}$ with State-of-the-art Classifiers

Figure 7.2 shows a comparison between the COID-FLA-$FCDCM_{GK}$ algorithm and the standard DCA version, SVM, ANN and the DTree classifiers. Focusing on the classification accuracy of the algorithms, the comparison made is in terms of the average of accuracies on the used databases. Figure 7.2 shows that the standard PCA-DCA version when applied to the ordered data sets has better classification accuracy than SVM, ANN and DTree confirming the results obtained from literature.

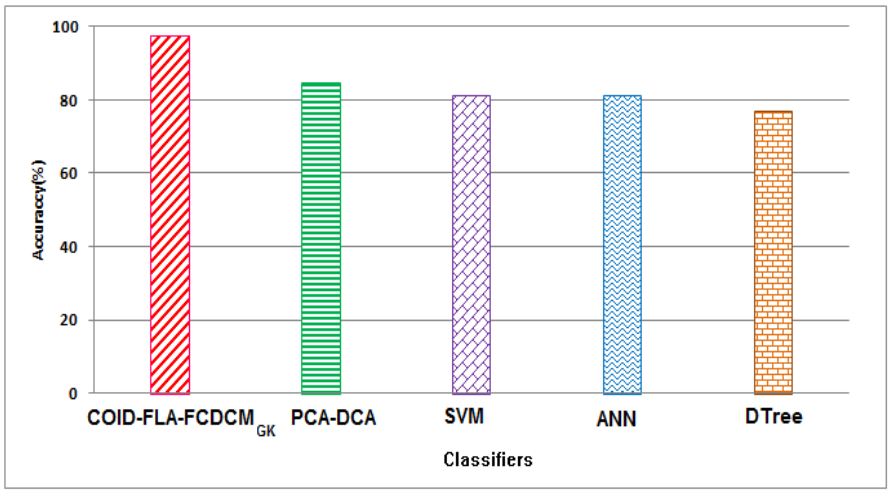

Figure 7.2: Comparative Results

Indeed, Figure 7.2 shows that the highest peak is for our proposed COID-FLA-$FCDCM_{GK}$ algorithm. From Figure 7.2, we can clearly see that the difference between the standard PCA-DCA version and COID-FLA-$FCDCM_{GK}$ is outstanding. This can clearly highlight the advantages of COID-FLA-$FCDCM_{GK}$ over its first basic version; i.e., the standard DCA.

These encouraging COID-FLA-$FCDCM_{GK}$ results are explained by the robust algorithmic steps of our proposed algorithm. First, the algorithm is characterized by its well-studied data pre-processing phase. The algorithm selects the most appropriate set of features and categorizes them to their right and specific signal types. This was handled by the application of fuzzy-rough set theory. Secondly, these good results are explained by the robust detection phase which was deeply investigated. This phase was handled by the use of the COID technique in order to ensure a good quality of the generated signal base.

Adding to these interesting notes, COID-FLA-$FCDCM_{GK}$ is coping with the crisp notions and concepts of the basic algorithm as it relies within an imprecise framework. This was handled by the use of fuzzy sets. Such hybridization led to the generation of a more stable classifier. These components added to the standard version of the dendritic cell algorithm lead us to obtain an overall smoothed automated and maintained DCA version.

7.6 Summary

In this Section, a summary of the whole work is presented by Figure 7.3 which shows the DCA different steps and the studies performed on each phase. Starting with the standard version of the dendritic cell algorithm, the input, several investigations have been conducted to more clarify the functioning of the DCA as well as its algorithmic properties and behavior. The output is, thus, the COID-FLA-$FCDCM_{GK}$ method.

<table>
<tr><td colspan="10">Input : DCA</td></tr>
<tr><td rowspan="4">Data pre-processing Phase</td><td rowspan="2">Feature Selection</td><td colspan="3">Rough Set Theory</td><td colspan="3">Fuzzy Rough Set Theory</td><td rowspan="2">PCA</td></tr>
<tr><td>RST</td><td>RC</td><td>QR</td><td>FRST</td><td>FLA</td><td>FBR</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>✓</td><td></td><td></td></tr>
<tr><td rowspan="2">Signal Categorization</td><td colspan="3">Same Features</td><td colspan="4">Different Features</td></tr>
<tr><td colspan="3"></td><td colspan="4">✓</td></tr>
<tr><td rowspan="3">Detection Phase</td><td rowspan="3">Quality of the Signal Base</td><td colspan="5">Maintenance policy</td><td colspan="3">No Maintenance policy</td></tr>
<tr><td>COID</td><td>CNN</td><td>RNN</td><td>IB2</td><td>IB3</td><td colspan="3" rowspan="2"></td></tr>
<tr><td>✓</td><td></td><td></td><td></td><td></td></tr>
<tr><td rowspan="4">Context Assessment Phase</td><td rowspan="2">Smoothness of the algorithm</td><td colspan="5">Fuzzy Set Theory</td><td colspan="3">Crisp Set Theory</td></tr>
<tr><td colspan="5">✓</td><td colspan="3"></td></tr>
<tr><td rowspan="2">Parameters Tuning</td><td colspan="5">Fuzzy Clustering Technique</td><td colspan="3">Manually</td></tr>
<tr><td>GK</td><td colspan="2">FCM</td><td colspan="2">AFCS</td><td colspan="3" rowspan="2"></td></tr>
<tr><td></td><td>✓</td><td></td><td></td><td></td></tr>
<tr><td>Classification Phase</td><td colspan="9"></td></tr>
<tr><td colspan="10">Output : COID-FLA-FCDCM$_{GK}$</td></tr>
</table>

Figure 7.3: A Progress Summary: from DCA to COID-FLA-$FCDCM_{GK}$

COID-FLA-$FCDCM_{GK}$ is the fruit of several hypotheses, experimentations and observations. This output which is an extension of the DCA can handle some of the DCA limitations; mainly related to the algorithm data pre-processing phase and to its sensitivity to the input class data order.

7.7 Conclusion

In this Chapter, we have proposed a hybrid automated and maintained fuzzy dendritic cell immune classifier. COID-FLA-$FCDCM_{GK}$ can be seen as the output of several investigations conducted on the standard DCA version. A set of hypotheses has been checked and based on the results drawn for the six hypotheses, the COID-FLA-$FCDCM_{GK}$ was developed. The worthy characteristics of the latter algorithm were highlighted in this Chapter as well.

Conclusion

Throughout this dissertation, we have investigated a series of algorithmic properties of the standard dendritic cell algorithm. The objective is to improve the applicability and the accessibility of the DCA to future users. The fruit of these investigations was an overall automated and maintained dendritic cell classifier characterized by its robustness and stability. Yet, to achieve the latter algorithm several developments were proposed in order to conduct conclusions and observations.

More precisely, we have tended to solve two main limitations of the standard version of the dendritic cell algorithm. The first limitation is linked to the algorithm non-robust data pre-processing phase. The second limitation is linked to the algorithm sensitivity to the input class data order. By taking into consideration these two points and by proposing the adequate solutions, the output was a refined DCA version lacking the mentioned limitations. We can describe the process of developing this output as follows:

First of all and throughout this dissertation, we have studied the DCA data pre-processing phase. This phase was seen as inconsistent since it is based on the use of the Principal Component Analysis statistical technique (PCA). Thus, we have tended to build a robust data pre-processing step based on more adequate data pre-processing techniques. Both of the Rough Set Theory (RST) and Fuzzy-Rough Set Theory (FRST) were involved at this stage.

Focusing on this aspect, we have proposed three different rough DCAs where each algorithm is based on a different feature selection process. The algorithms are named RST-DCA, RC-DCA and QR-DCA. These RST based methods could produce significantly better classification performance than the PCA statistical technique. This shows that it is better to use a feature selection technique than to apply a combination of dimensionality reduction and statistical inference techniques. In addition, among the three automated rough pre-processing methods, the QR based method appears to produce satisfactory classification performance while keeping the algorithm lightweight in terms of running time. Despite of the advantages of the developed rough DCAs, these methods have to perform data discretization before feature selection. To handle this limitation, three automated fuzzy-rough DCAs have been proposed; namely FRST-DCA, FLA-DCA and FBR-DCA. Each of these algorithms is based on a specific feature selection process based on FRST. These algorithms could minimize the information loss during data pre-processing and which is due to the data quantization process. As a result, these FRST based methods are advantageous over the other data pre-processing methods; i.e., using RST and feature reduction techniques.

While comparing the fuzzy-rough DCAs, we have noticed that there is no significant difference between the classification performance of the algorithms; but, that was with an extra cost in terms of running time. Most importantly, the FLA based approach was seen as the most interesting technique as it generates satisfactory classification results while processing in less time in comparison to the other two DCAs based fuzzy-rough sets; i.e., FRST-DCA and FBR-DCA.

In summary, when focusing on the standard DCA data pre-processing phase we could develop an automated data pre-processing module that replaces both the manual one, based on expert knowledge, and if it is based on dimensionality reduction techniques as well. This is to interface the input space of the DCA with a given problem domain. The rough and the fuzzy-rough proposed DCAs demonstrate the possibility of combining feature selection techniques that are commonly used in machine learning and mathematical modeling, to automate the data pre-processing phase of the algorithm. This could potentially make the algorithm more applicable to a larger set of problems in which expert knowledge is difficult or infeasible to extract.

After that and once we have investigated the DCA data pre-processing phase, we have focused on the DCA limitation as the algorithm is known to be sensitive to the input class data order. We have set two possible hypotheses. The first hypothesis states that the DCA sensitivity to the input class data order is tight to the algorithm detection phase. The second hypothesis states that the DCA sensitivity to the input class data order is tight to the algorithm context assessment phase. Both hypotheses were checked and proved to be true.

Our empirical investigations contacted in this part show that the mandatory respect of the antigens class order is related to two main causes. The first cause is linked to the DCA environment which is characterized by a crisp separation between the DC semi-mature context and the DC mature context. The second cause is linked to the quality of the algorithm generated signal base. However, the reality is connected to imprecision by nature. Such imperfection may affect the DCA classification performance leading the algorithm being sensitive to the input class data order. Therefore, we have combined fuzzy set theory with the DCA to deal with imprecise contexts. As for the quality of the signal base, we have to handle this point by the application of the COID maintenance method. The results of these hybridizations led to the fuzzy DCA versions which are stable classifiers.

In summary, our work conducted in this part demonstrates the possibility of combining theories managing imprecision with the dendritic cell algorithm. This could potentially make the algorithm more applicable to any data set order with no need to check or respect the data items class order.

Once we have solved the DCA main limitations, we tended to develop a more general algorithm based on the conducted observations and conclusions. We have built a more robust danger theory classifier based dendritic cells within an imprecise framework. The developed algorithm is based on robust and well-studied algorithmic steps. It is also based on the fuzzy lower approximation FRST concept to ensure a most convenient data pre-processing phase. Indeed, it uses fuzzy set theory to ensure the stability of the algorithm. The proposed new DCA version is a user-independent algorithm as its parameters are automatically generated by the use of adequate techniques. Adding to these, the algorithm applies a maintenance technique to ensure a good quality of its input parameters. The fruit of all these works is named the COID-FLA-$FCDCM_{GK}$ algorithm.

Ultimately, the investigations conducted in this dissertation led to a new version of the dendritic cell algorithm. COID-FLA-$FCDCM_{GK}$ tended to solve and overcome the standard DCA previously discussed limitations and shortcomings. COID-FLA-$FCDCM_{GK}$ is a hybrid automated and maintained fuzzy dendritic cell immune classifier.

The developed DCA hybrid versions, proposed in this dissertation, involve integrating techniques from the machine learning field with the dendritic cell algorithm. Experimentations show that the integrated systems become more applicable, more automated and more adaptable to the problem at hand. At the end, the mentioned evaluation aims are particularly satisfied, but through attempting to achieve them, the limitations of our proposed algorithms are explored. In fact, it is important to note that there is always a trade-off between the effectiveness of our proposed algorithms and the extra complexity they bring to the system. As a result, in terms of future work, it might be more effective to slightly modify the algorithms internal components to improve their detection capability while proposing a lightweight complexity for the algorithms.

Indeed, a number of interesting future works has to be mentioned. Firstly and while focusing on the algorithms data pre-processing phase, handling imperfection in attribute values seems to be an important issue. Furthermore, imperfection in attribute values could be handled in both building and classification procedures. Moreover, it would be interesting to build a new data pre-processing phase capable of handling missing data while performing the feature selection process. Still focusing on the data pre-processing phase, the fact of handling noisy data with the used data sets would be another motivating point to explore.

Secondly and while emphasizing on the algorithms detection phase, the weights for signal transformation are either statically assigned or are user defined parameters. This could extensively constrain the algorithm's ability to fully adapt to a given problem domain. Therefore, a special focus on this phase would be a great of interest. The aim is to determine the optimum weight values for maximum detection accuracy.

Thirdly, it is yet important to mention that inflammation signals have occasionally featured throughout this Thesis. Consequently, a formal analysis of these signals may be performed to understand their effects on the proposed algorithms. As a consequence, new signal categorization methodologies could be proposed.

These new investigations will involve integrating techniques from other fields, e.g. machine learning, signal processing and statistical inference, with the dendritic cell algorithm. Yet, researchers in machine learning often adopt Occam's Razor when designing learning models, and simplicity is the key to successful applications. Vapnik suggested that when trying to solve some problems, one should not solve a more difficult problem as an intermediate step (Vapnik, 1999). As a consequence, one should bear in mind the trade-off between computational complexity and classification performance, when modifying the current system based on proposed solutions.

Another line of research could be the applicability of the standard dendritic cell algorithm as well as the proposed derived DCA versions. Firstly, it would be very interesting to extend the application area of the DCA to problems with multi classes. It would be encouraging to focus on the application of the algorithms to different domains such as the intrusion detection problem or the image registration field and to discern the algorithms' characteristics, advantages and limitations. Finally, the incorporation of the DCA and the proposed algorithms into a larger framework may prove fruitful and the resultant system would be able to both perform antigens classification and respond appropriately.

Appendix

In this Appendix, we will present Fuzzy Set Theory (FST). FST is used as one of the main techniques to build our several proposed algorithms throughout this dissertation.

Appendix A

Fuzzy Set Theory

A.1 Introduction

Fuzzy set theory was introduced, in 1965, by Zadeh (Zadeh, 1965). It is considered as a useful theory for modeling and reasoning with imprecise knowledge. Fuzzy set theory is a mathematical theory where the fuzziness is the ambiguity that can be found in the definition of a concept or the meaning of a word (Zimmermann, 1996). Imprecision in expressions like "low frequency", "high demand" or "small number" can be called fuzziness. In this Appendix, the basics of fuzzy sets will be introduced.

In Section A.2, the basics of fuzzy set theory will be given. In Section A.3, the main notions of the fuzzy set membership functions will be introduced while in Section A.4, the fundamental operations of fuzzy sets will be highlighted. Finally, in Section A.5, the process of fuzzy logic will be described.

A.2 Fuzzy sets: Definition and Notations

Fuzzy sets were introduced as an extension of the classical notion of a set. In the classical (crisp) set theory, a very precise and clear boundary exists to show if an element either belongs or does not belong to the set. Hence, an element is not allowed to be in the set and not in the set at the same time. In contrast, a fuzzy set is a set without a clearly defined boundary. It permits the gradual assessment of the membership of elements in a set; this is described with the aid of a *membership function*.

Let X denote the universe of discourse and its elements are denoted by x, then a fuzzy set A in X is defined as a set of pairs:

$$A = \{(x, \mu_A(x)) | x \in X\}$$

where $\mu_A(x)$ is called the membership function of x in A. The membership function maps each element of X to a membership value between 0 and 1.

Fuzzy sets are based on *linguistic variables* (Zadeh, 1975; Tong & Bonissone, 1980). A linguistic variable is a variable whose values are not numbers but words or sentences. The set of values that it can take is called *term set*. Each term set constitutes a fuzzy set in the *universe of discourse* which contains all elements that can come into consideration.

Example A.1 *Let us consider an example dealing with the grade of maturity of a fruit. The universe of discourse related to the grade of maturity is the scale from 0 to 50. The linguistic variable "maturity" takes three term sets which are fuzzy sets labeled as "verdant", "half-mature" and "mature".*

A.3 Membership Functions

A membership function is a curve that defines how each point in the universe of discourse is mapped to a membership value (or degree of membership) between 0 and 1 (Goguen, 1967; Dubois & Prade, 1997).

Let X denote a universe of discourse. Then, the membership function μ_A, by which a fuzzy set A is defined, has the form:

$$\mu_A : X \rightarrow [0, 1]$$

where [0, 1] is the interval of real numbers from 0 to 1, inclusive.

The membership function $\mu_A(x)$ quantifies the grade of membership of the elements x to the fundamental set X. An element mapping to the value 0 means that the member is not included in the given set, 1 describes a fully included member. Values strictly between 0 and 1 characterize the fuzzy members.

A.4 Operations on Fuzzy Sets

The basic connective operations in classical set theory are those of intersection, union and complement. These operations can be generalized to fuzzy sets (Zimmermann, 1996; Dubois & Prade, 1997).

Let A and B be two fuzzy sets within a universe of discourse X with membership functions μ_A and μ_B, respectively. The following fuzzy set operations can be defined.

• **Fuzzy complement:**

The complement of a fuzzy set A is denoted by the fuzzy set $\bar{A}$. It corresponds to the *Boolean NOT* function and is given by Equation A.1:

$$c : [0, 1] \rightarrow [0, 1]$$

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \tag{A.1}$$

• **Fuzzy intersection:**

The intersection of two fuzzy sets A and B corresponds to the *Boolean AND* function and is given by Equation A.2:

$$i : [0, 1] \times [0, 1] \rightarrow [0, 1]$$

$$\mu_{A \cap B}(x) = min[\mu_A(x), \mu_B(x)] \tag{A.2}$$

The fuzzy intersection operator (fuzzy AND connective) can also be represented as the *algebraic product* of two fuzzy sets A and B, which is defined as the multiplication of their membership functions:

$$\mu_{A \cap B}(x) = \mu_A(x) \cdot \mu_B(x), x \in X \tag{A.3}$$

• **Fuzzy union:**

The union of two fuzzy sets A and B corresponds to *Boolean OR* function and is given by Equation A.4:

$$u : [0, 1] \times [0, 1] \rightarrow [0, 1]$$

$$\mu_{A \cup B}(x) = max[\mu_A(x), \mu_B(x)] \tag{A.4}$$

The fuzzy union operator (fuzzy OR connective) can also be represented as the *algebraic sum* of two fuzzy sets A and B, which is defined by Equation A.5:

$$\mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x) \tag{A.5}$$

• **Fuzzy equality:**

Two fuzzy sets A and B are equal if they have the same membership function within a universe of discourse X:

$$\text{A} = \text{B}, \text{ iif } \mu_A(x) = \mu_B(x) \tag{A.6}$$

• **Other fuzzy operations:**

Other operations on fuzzy sets could be mentioned in (Silvert, 1979). They are the same as for crisp set including the following list.

- Commutativity: $A \cup B = B \cup A$
- Associativity: $A \cup (B \cup C) = (A \cup B) \cup C$
- Distributivity: $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
- Idempotency: $A \cap A = A$
- Identity: $A \cap \emptyset = \emptyset$
- Involution: $\neg(\neg A) = A$
- Transitivity: $(A \subseteq B \subseteq C)$ then $(A \subseteq C)$
- DeMorgan's Laws
  * $\neg(A \cap B) = \neg A \cup \neg B$
  * $\neg(A \cup B) = \neg A \cap \neg B$

A.5 Fuzzy Logic

Fuzzy logic is derived from fuzzy set theory (Zadeh, 1990). It underlines modes of reasoning which are approximate rather than exact (Zadeh, 1989). That is, it handles the concept of partial truth - truth values between "completely true" and "completely false".

Fuzzy logic is based on the *fuzzy logic controller* (FLC). The structure of the FLC is shown in Figure A.1.

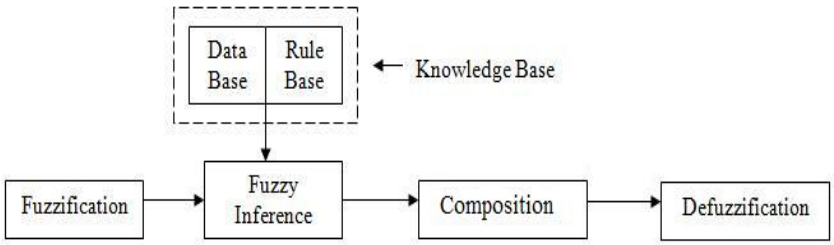

Figure A.1: Fuzzy Logic Controller

The FLC is composed of five boxes which are explained as follows:

• **Fuzzification:**

Fuzzification is the process of identifying the input and output of the system, defining appropriate IF-THEN rules as well as the membership function.

• **Fuzzy rule base:**

The fuzzy rule base consists of a set of antecedent - consequent linguistic rules of the form

IF antecedent THEN consequent

These rules express the relations between the input and output.

• **Fuzzy inference:**

In order to draw conclusions from a rule base we need a mechanism that can produce an output from a collection of *if-then* rules. This is done using the *compositional rule of inference* (CROI). This process evaluates all the rules and determines their truth values.

There are many methods dealing with the inference process such as *max-min* known as the *MAMDANI* method (Mamdani & Assilian, 1975), *max-prod* (Kyosev & Reinbach, 2006) and *sum-prod* method (Mizumoto, 1990).

• **Composition:**

It is the fact of combining all fuzzy conclusions obtained by the inference process into a single conclusion. Since different fuzzy rules might have different conclusions, we should consider all rules.

• **Defuzzification:**

This step is concerned with converting the fuzzy value obtained from composition into a “crisp” value. There are many defuzzification methods such as the *centroid* method (the center of gravity of the membership function) (Broekhoven & Baets, 2006) and the *maximum* method (the maximum truth value) (Lee, 1990).

A.6 Worked Example

Consider an example with two inputs and one output. The different steps are presented as follows:

- The fuzzification process:
  - Assume that we have two inputs (x, y) and one output (z).
  - The membership functions are represented in Figure A.2.

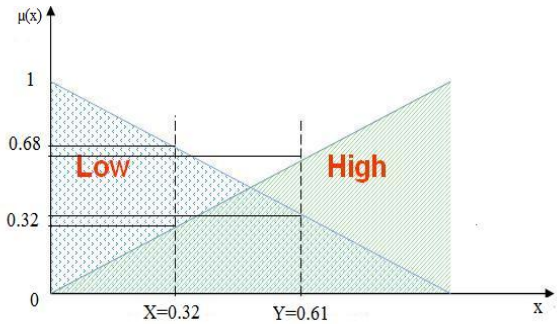

Figure A.2: Defining the Membership Functions Value's of Each Input

  - Suppose that the crisp inputs are x = 0.32 and y = 0.61. We project both of the inputs on each of the membership functions. Thus we get:
    Low(x) = 0.68, High(x) = 0.32
    Low(y) = 0.39, High(y) = 0.61
  - The rule base is as follows:
    * Rule 1: If x is low AND y is low Then z is high
    * Rule 2: If x is low AND y is high Then z is low
    * Rule 3: If x is high AND y is low Then z is low
    * Rule 4: If x is high AND y is high Then z is high
- The inference process:
  - Rule1: low(x) = 0.68, low(y) = 0.39 → high(z) = MIN(0.68,0.39) = 0.39
  - Rule2: low(x) = 0.68, high(y) = 0.61 → low(z) = MIN(0.68,0.61) = 0.61
  - Rule3: high(x) = 0.32, low(y) = 0.39 → low(z) = MIN(0.32,0.39) = 0.32
  - Rule4: high(x) = 0.32, high(y) = 0.61 → high(z) = MIN(0.32,0.61) = 0.32
- The composition process:
  - Low(z) = MAX(rule2, rule3) = MAX(0.61, 0.32) = 0.61
  - High(z) = MAX(rule1, rule4) = MAX(0.39, 0.32) = 0.39
  - We project each of these values on the two membership functions "Low" and "Hight" respectively (Figure A.3(a)) to get a new membership function (Figure A.3(b)).

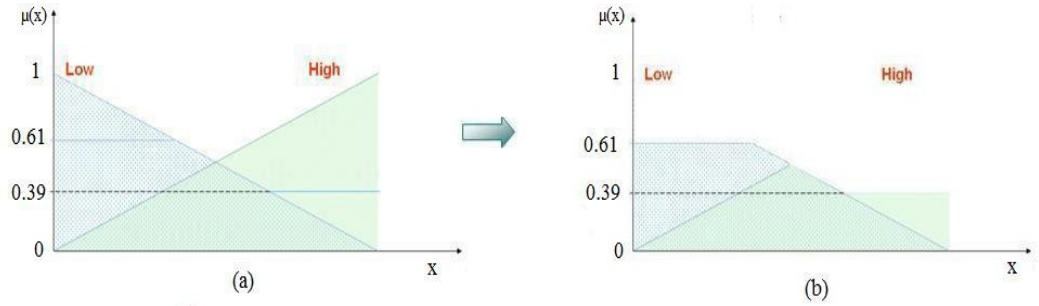

Figure A.3: Composition Process

- The defuzzification process:
  By applying the centroid method we get the following crisp output (see Figure A.4):

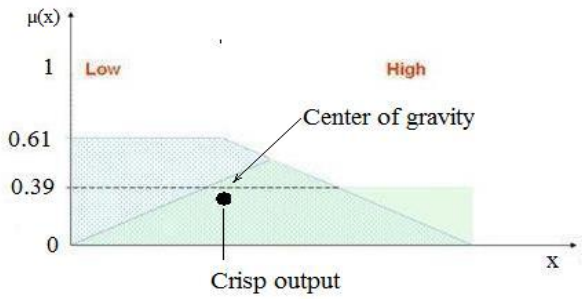

Figure A.4: Centroid Method for Defuzzification

A.7 Conclusion

In this Appendix, we have elucidated the basics of fuzzy set theory which is a generalization of the classical set theory. Offering a natural model, fuzzy sets are used to handle imprecise information.

References

Aha, D., Kibler, G., & Albert, M. K. (1991). Instance-based learning algorithms. *IEEE Transactions on Fuzzy Systems*, *1*, 37-66.

Asuncion, A., & Newman, D. (2007). Uci machine learning repository. *http://archive.ics.uci.edu/ml/*.

Bezdek, C. (1981). *Pattern recognition with fuzzy objective function algorithms*. Plenum Press.

Broekhoven, E., & Baets, B. (2006). Fast and accurate center of gravity defuzzification of fuzzy system outputs defined on trapezoidal fuzzy partitions. *Fuzzy Sets and Systems*, *157*, 904-918.

Brownlee, J. (2007). *Clonal selection algorithms*. Technical Report, Faculty of Information Communication Technology, Swinburne University of Technology Melbourne, Australia.

Bussian, B., & Härdle, M. (1984). Robust smoothing applied to white noise and single outlier contaminated raman spectra. *Applied Spectroscopy*, *383*, 309-313.

Cantú-Paz, E. (2004). Feature subset selection, class separability, and genetic algorithms. *Proceedings of the Genetic and Evolutionary Computation Conference, ACM, GECCO'2004*, 959-970.

Castro, L., & Zuben, V. (2001). Learning and optimization using clonal selection principle. *IEEE Transaction on Evolutionary Computation*, *6*, 239-251.

Cayzer, S., & Aickelin, U. (2002). The danger theory and its application to artificial systems. *Proceedings of the 1st International Conference on Artificial Immune Systems, Springer, ICARIS'2002*, *Lecture Notes in Computer Science, 2787*, 141-148.

Chelly, Z., & Elouedi, Z. (2010). Fdcm: A fuzzy dendritic cell method. *Proceedings of the 9th International Conference on Artificial Immune Systems, Springer, ICARIS'2010*, *Lecture Notes in Computer Science, 6209*, 102-115.

Chelly, Z., & Elouedi, Z. (2011). Further exploration of the fuzzy dendritic cell method. *Proceedings of the 10th International Conference of Artificial Immune Systems, Springer, ICARIS'2011*, *Lecture Notes in Computer Science, 6825*, 419-432.

Chelly, Z., & Elouedi, Z. (2012a). Rc-dca:a new feature selection and signal categorization technique for the dendritic cell algorithm based on rough set theory. *Proceedings of the 11th International Conference of Artificial Immune Systems, Springer, ICARIS'2012*, *Lecture Notes in Computer Science, 7597*, 152-165.

Chelly, Z., & Elouedi, Z. (2012b). Rst-dca: A dendritic cell algorithm based on rough set theory. *Proceedings of the 19th International Conference on Neural Information Processing, Springer, ICONIP'2012*, *Lecture Notes in Computer Science, 7665*, 480-487.

Chelly, Z., & Elouedi, Z. (2013a). Further exploration of the hybrid fuzzy-rough dendritic cell immune classifier. *Proceedings of the 4th International Conference on E-Health and Bioengineering, IEEE, EHB'2013*, 1-4.

Chelly, Z., & Elouedi, Z. (2013b). A fuzzy-rough data pre-processing approach for the dendritic cell classifier. *Proceedings of the 12th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, Springer, ECSQARU'2013*, *Lecture Notes in Computer Science, 7958*, 109-120.

Chelly, Z., & Elouedi, Z. (2013c). New data pre-processing approach for the dendritic cell algorithm based on fuzzy rough set theory. *Proceedings of the Genetic and Evolutionary Computation Conference, ACM, GECCO'2013*, 163-164.

Chelly, Z., & Elouedi, Z. (2013d). A new hybrid fuzzy-rough dendritic cell immune classifier. *Proceedings of the 4th International Conference on Swarm Intelligence, Springer, IC-SI'2013*, *Lecture Notes in Computer Science, 7928*, 514-521.

Chelly, Z., & Elouedi, Z. (2013e). Qr-dca : A new rough data pre-processing approach for the dendritic cell algorithm. *Proceedings of the 11th International Conference on Adaptive and Natural Computing Algorithms, Springer, ICANNGA'2013*, *Lecture Notes in Computer Science, 7824*, 140-150.

Chelly, Z., & Elouedi, Z. (2013f). Supporting fuzzy-rough sets in the dendritic cell algorithm data pre-processing phase. *Proceedings of the 20th International Conference on Neural Information Processing, Springer, ICONIP'2013*, *Lecture Notes in Computer Science, 8227*, 164-171.

Chelly, Z., & Elouedi, Z. (2014a). Further exploration of the hybrid fuzzy-rough dendritic cell immune classifier. *Proceedings of the 8th Hellenic Conference on Artificial Intelligence, Springer, SETN'2014*, *Lecture Notes in Computer Science, 8445*, 43-56.

Chelly, Z., & Elouedi, Z. (2014b). Improving the dendritic cell algorithm performance using fuzzy-rough set theory as a pattern discovery technique. *Proceedings of the 5th International Conference on Innovations in Bio-Inspired Computing and Applications, Springer, IBICA'2014*, *Advances in Intelligent Systems and Computing, 303*, 23-32.

Chelly, Z., & Elouedi, Z. (2014c). A two-leveled hybrid dendritic cell algorithm under imprecise reasoning. *Proceedings of the Genetic and Evolutionary Computation Conference, ACM, GECCO'2014*, 97-104.

Chelly, Z., & Elouedi, Z. (2015). Hybridization schemes of the fuzzy dendritic cell immune bi-

nary classifier based on different fuzzy clustering techniques. *New Generation Computing, Springer (To Appear).*

Chelly, Z., Smiti, A., & Elouedi, Z. (2012). Coid-fdcm: The fuzzy maintained dendritic cell classification method. *Proceedings of the 11th International Conference on Artificial Intelligence and Soft Computing, Springer, ICAISC'2012*, *Lecture Notes in Computer Science, 7268*, 233-241.

Chou, C., Kuo, B., & Chang, F. (2006). The generalized condensed nearest neighbor rule as a data reduction method. *Proceedings of the 18th International Conference on Pattern Recognition, IEEE, ICPR'2006*, 556-559.

Chouchoulas, A., & Shen, Q. (2001). Rough set-aided keyword reduction for text categorization. *Applied Artificial Intelligence Journal*, *15*, 843-873.

Dave, R. (1993). Robust fuzzy clustering algorithms. *Proceedings of the 2nd IEEE International Conference on Fuzzy Systems, IEEE, FUZZ-IEEE'1993*, 1281-1286.

Dubois, D., & Prade, H. (1992). Putting rough sets and fuzzy sets together. *Intelligent decision support : Handbook of applications and advances of the rough sets theory, Springer*, *11*, 203-232.

Dubois, D., & Prade, H. (1997). The three semantics of fuzzy sets. *Fuzzy Sets and Systems*, *90*, 141-150.

Dunn, C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, *3*, 32-57.

Eiben, A., & Smith., J. (2007). *Introduction to evolutionary computing*. Natural Computing.

Filzmoser, P., Garrett, R., & Reimann, C. (2005). Multivariate outlier detection in exploration geochemistry. *Computers and Geosciences*, *31*, 579-587.

Fu, H., & Li, G. (2008). Design of an immune-inspired danger theory model based on fuzzy set. *Proceedings of the International Symposium on Computational Intelligence and Design, IEEE, ISCID'2008*, 133-136.

Fu, H., & Zhang, C. (2009). Design of a danger signal detecting model based on fuzzy-set. *Proceedings of 5th International Conference on Wireless communications, networking and mobile computing, IEEE, WiCOM'2009*, 4566-4568.

Gates, W. (1972). The reduced nearest neighbor rule. *IEEE Transactions on Information Theory*, *18*, 431-433.

Goguen, J. (1967). L-fuzzy sets. *Journal of Mathematical Analysis and Applications*, *18*, 145-174.

Goldsby, R., Kindt, T., Osborne, B., & Kuby, J. (2003). *Immunology*. W. H. Freeman and Company.

Gonzalez, F. (2003). *A study of artificial immune systems applied to anomaly detection*. Doctoral Dissertation, The University of Memphis.

Greensmith, J. (2007). *The dendritic cell algorithm*. Doctoral Dissertation, University of Nottingham.

Greensmith, J., & Aickelin, U. (2005). Introducing dendritic cells as a novel immune-inspired algorithm for anomaly detection. *Proceedings of the 4th International Conference on Artificial Immune Systems, Springer, ICARIS'2005*, *Lecture Notes in Computer Science, 3627*, 153-167.

Greensmith, J., & Aickelin, U. (2006a). Articulation and clarification of the dendritic cell algorithm. *Proceedings of the 5th International Conference on Artificial Immune Systems, Springer, ICARIS'2005*, *Lecture Notes in Computer Science, 4163*, 404-417.

Greensmith, J., & Aickelin, U. (2006b). Dendritic cells for anomaly detection. *Proceedings of the IEEE Congress on Evolutionary Computation, IEEE, CEC'2006*, 664-671.

Greensmith, J., & Aickelin, U. (2007). Dendritic cells for syn scan detection. *Proceedings of the Genetic and Evolutionary Computation Conference, ACM, GECCO'2007*, 49-56.

Greensmith, J., & Aickelin, U. (2008a). The dca for bot detection. *Proceedings of the World Congress on Computational Intelligence, IEEE, WCCI'2008*, 1807-1816.

Greensmith, J., & Aickelin, U. (2008b). The deterministic dendritic cell algorithm. *Proceedings of the 7th International Conference on Artificial Immune Systems, Springer, ICARIS'2008*, *Lecture Notes in Computer Science, 5132*, 291-302.

Greensmith, J., & Aickelin, U. (2009). Intergrating real-time analysis with the dendritic cell algorithm through segmentation. *Proceedings of the Genetic and Evolutionary Computation Conference, ACM, GECCO'2009*, 1203-1262.

Greensmith, J., Aickelin, U., & Tedesco, G. (2010). Information fusion for anomaly detection with the dendritic cell algorithm. *Information Fusion*, *11*, 21-34.

GU, F., Oates, R., Greensmith, J., & Aickelin, U. (2009). Pca 4 dca: The application of principal component analysis to the dendritic cell algorithm. *Proceedings of the 9th Annual Workshop on Computational Intelligence, IEEE, UKCI'2009*, 1-6.

Gustafson, D., & Kessel, W. (1979). Fuzzy clustering with a fuzzy covariance matrix. *Proceedings of the IEEE Conference on Decision and Control, IEEE, CDC'1979*, 761-766.

Hoppner, F., Klawonn, F., Kruse, R., & Runkler, T. (1999). *Fuzzy cluster analysis*. Wiley.

Janeway, C. (1992). The immune system evolved to discriminate infectious nonself from noninfectious self immunol. *Immunology Today*, *13*, 11-16.

Jensen, R. (2005). *Combining rough and fuzzy sets for feature selection*. Doctoral Dissertation, School of Informatics University of Edinburgh.

Jensen, R., & Shen, Q. (2002). Fuzzy-rough sets for descriptive dimensionality reduction. *Proceedings of the 11th International Conference on Fuzzy Systems, IEEE, FUZZ-IEEE'2002*, 29-34.

Jensen, R., & Shen, Q. (2007). Fuzzy-rough sets assisted attribute selection. *IEEE Transactions on Fuzzy Systems*, *15*, 73-89.

Krishnapuram, R., Nasraoui, O., & Frigui, H. (1995). Fuzzy and possibilistic shell clustering algorithms and their application to boundary detection and surface approximation. *IEEE Transactions on Neural Networks*, *3*, 29-60.

Kyosev, Y., & Reinbach, I. (2006). Max-product fuzzy relational equations as inference engine for prediction of textile yarn properties. *Proceedings of the 9th International Conference on Computational Intelligence, Springer, CCI'2006*, *Lecture Notes in Artificial Intelligence, 38*, 18-20.

Leake, D., & Wilson, D. (2001). Maintaining case-based reasoners: Dimensions and directions. *Computational Intelligence*, *17*, 196-213.

Lee, C. (1990). Fuzzy logic in control systems: Fuzzy logic controller - parts 1 and 2. *IEEE Transactions on Systems, Man and Cybernetics*, *20*, 404-435.

Lutz, M., & Schuler, G. (2002). Immature, semi-mature and fully mature dendritic cells: which signals induce tolerance or immunity? *TRENDS in Immunology*, *23*, 445-449.

Mamdani, E., & Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, *51*, 1-13.

Markovich, S., & Scott, P. (1988). The role of forgetting in learning. *Proceedings of the 5th International Conference on Machine Learning, IEEE, ICML'1988*, 459-465.

Markovitch, S., & Scott, P. (1993). Information filtering: Selection mechanisms in learning systems. *Machine Learning*, *10*, 113-151.

Matzinger, P. (2001). The danger model in it's historical context. *Scandinavian Journal of Immunology*, *54*, 4-9.

Minton, S. (1990). Qualitative results concerning the utility of explanation-based learning. *Artifical Intelligence*, *42*, 363-369.

Mizumoto, M. (1990). Fuzzy controls by product-sum gravity-method. *Fuzzy Sets and Systems*, *2*, 461-466.

Musselle, C. (2010). Insights into the antigen sampling component of the dendritic cell algorithm. *Proceedings of the 9th International Conference on Artificial Immune Systems, Springer, ICARIS'2010*, *Lecture Notes in Computer Science, 6209*, 88-101.

Nikhil, R., Bezdek, C., & James, C. (1995). On cluster validity for the fuzzy c-means model. *IEEE Transactions on Fuzzy Systems*, *3*, 370-379.

Racine, K., & Yang, Q. (1996). On the consistency management of large case bases: the case for validation. *Proceedings of the AAAI workshop on knowledge base validation, IEEE, AAAI'1996*, 84-90.

Racine, K., & Yang, Q. (1997). Maintaining unstructured case bases. *Proceedings of the 2nd International Conference on Case-Based Reasoning, Springer, ICCBR'1997*, *Lecture Notes in Computer Science, 1266*, 553-564.

Sander, J., Ester, M., Kriegel, P., & Xu, X. (1998). Density-based clustering in spatial databases the algorithm gdbscan and its applications. *Data Mining, Knowledge Discovery*, *2*, 169-194.

Shafer, G. (1976). *A mathematical theory of evidence*. Princeton University Press.

Silvert, W. (1979). Symmetric summation: a class of operations on fuzzy sets. *IEEE Transactions on Systems, Man, and Cybernetics*, *9*, 657-659.

Slezak, D. (1996). Approximate reducts in decision tables. *Proceedings of the 6th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Springer, IPMU'1996*, *Lecture Notes in Computer Science, 6209*, 1159-1164.

Smiti, A., & Elouedi, Z. (2010). Coid : Maintaining case method based on clustering, outliers and internal detection. *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, *295*, 39-52.

Smiti, A., & Elouedi, Z. (2011). Overview of maintenance for case based reasoning systems. *International Journal of Computer Applications*, *8*, 49-56.

Swiniarski, R., & Skowron, A. (2003). Rough set methods in feature selection and recognition. *Pattern Recognition Letters*, *24*, 833-849.

Tong, R., & Bonissone, P. (1980). Linguistic approach to decision making with fuzzy sets. *IEEE Transactions on Systems, Man and Cybernetics*, *10*, 716-723.

Vapnik, V. (1999). *The nature of statistical learning theory*. Springer.

Whitbrook, A., Aickelin, U., & Garibaldi, J. (2010). Two-timescale learning using idiotypic behaviour mediation for a navigating mobile robot. *Journal of Applied Soft Computing*, *10*, 876-887.

Wilson, D. (1972). Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems Man and Cybernetics*, *2*, 408-421.

Xie, X. L., & Beni, G. A. (1991). Validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *3*, 841-846.

Zadeh, L. (1965). Fuzzy sets. *Information and Control*, *8*, 338-353.

Zadeh, L. (1975). The concept of a linguistic variable and its application to approximate reasoning. *Information Sciences*, *9*, 43-80.

Zadeh, L. (1989). Knowledge representation in fuzzy logic. *IEEE Transactions on Knowledge and Data Engineering*, *1*, 89-100.

Zadeh, L. (1990). The birth and evolution of fuzzy logic. *International Journal of General Systems*, *17*, 95-105.

Zhong, N., Dong, J., & Ohsuga, S. (2001). Using rough sets with heuristics for feature selection. *Journal of Intelligent Information Systems*, *16*, 199-214.

Zimmermann, H. (1996). *Fuzzy set theory and its applications*. Kluwer Academic.

Zuben, V., & Timmis, J. (2003). Artificial immune systems as a novel soft computing paradigm. *Soft Computing Journal*, *7*, 526-544.

Résumé:

Le travail présenté dans cette thèse décrit des nouvelles méthodes de classification inspirées de la théorie du danger dans un cadre imprécis. Ces nouvelles méthodes sont plus précisément basées sur l'algorithme des cellules dendritiques (DCA). L'objectif de cette thèse est de résoudre les problèmes de DCA tout en étudiant ses caractéristiques algorithmiques. Dans ce contexte, nous avons développé de nouvelles versions de DCA, plus robustes, basées sur des outils d'apprentissage automatique. Ces méthodes proposées visent, en premier lieu, à résoudre la non robustesse de la phase de prétraitement des données du DCA en développant des modules automatisés basés sur la théorie des ensembles approximatifs et celle des ensembles approximatifs flous. Nos méthodes visent, aussi, à résoudre la sensibilité du DCA face à l'ordre des classes des données. Ceci est traité par le développement des versions de DCA basées sur la théorie des ensembles flous, une technique de partitionnement de données floue et sur une méthode de maintenance. Enfin et en tenant en considération de ces parties, un DCA global est proposé.

Mots-clés: Théorie du Danger, Cellules Dendritiques, Théorie des Ensembles Flous, Techniques de Partitionnement de Données Floues, Théorie des Ensembles Approximatifs, Techniques de Maintenance.

Abstract:

The work presented in this Thesis focuses on developing new danger theory classification methods in an imprecise framework. The danger theory is based on the Dendritic Cell Algorithm (DCA). We aim to address and resolve the DCA issues by investigating its algorithmic characteristics. New DCA versions are developed leading to more robust DCA binary classifiers based on machine learning tools. These newly developed methods rely, first of all, on solving the DCA non-robust data pre-processing phase by proposing some automated modules based on Rough Set Theory and Fuzzy-Rough Set Theory. Secondly, the developed methods aim at investigating the reasons behind the sensitivity of the DCA to the input class data. This was handled by incorporating fuzzy set theory, a fuzzy clustering technique and a maintenance technique. Lastly, an overall hybrid DCA was proposed taking into account all the mentioned parts.

Keywords: Danger Theory, Dendritic Cells, Fuzzy Set Theory, Fuzzy Clustering, Rough Set Theory, Maintenance Techniques.

Année Universitaire 2013/2014