Université de Tunis

Institut Supérieur de Gestion de Tunis

Handling Imprecision in Danger Theory using Fuzzy Set Theory: A Fuzzy Dendritic Cell Method (FDCM)

Mémoire en vue de l'obtention du Mastère

en Informatique de Gestion

Présenté par

Chelly Zeineb

Sous la direction de

Mr. Zied Elouedi

Session: Juin 2010

Abstract

The work presented in this master thesis describes a new classification method using the danger theory (DT) approach under imprecision. The DT is based on the behavior of dendritic cells (DCs). The hybridization of DCs behavior with fuzzy set theory leads to the development of the so-called fuzzy dendritic cell method (FDCM). FDCM proposes to introduce concepts of fuzzy set theory to deal with imprecision found in the definition of words such as "semi-mature" and "mature" (the two states of DCs) on which depends the classification procedure in the danger theory. In addition to this imprecision, our method handles with the crisp separation between these two states. It allows to smooth such separation using fuzzy set basics. Experimentations on real data sets show that by alleviating this crisp separation, our new approach improves the classification accuracy in comparison to the standard dendritic cell algorithm.

Key words: Artificial immune systems, Danger theory, Dendritic cells, Fuzzy set theory.

Résumé

Le travail présenté dans ce rapport décrit une nouvelle méthode de classification basée sur la théorie du danger (TD) dans un environnement imprécis. La TD est fondée sur le comportement des cellules dendritiques (CDs). L'hybridation du comportement des CDs avec la théorie des ensembles flous aboutit au développement de la méthode des cellules dendritiques avec des ensembles flous. Cette méthode propose d'utiliser la théorie des ensembles flous pour traiter l'imprécision présente dans la définition de certains termes comme "semi-mûr" et "mûr" (les deux états des cellules dendritiques) sur lesquels se base la procédure de classification dans la théorie du danger. De plus, notre méthode traite la séparation rigide entre ces deux états en la lissant, et ce en utilisant les notions de la théorie des ensembles flous. Des expérimentations ont été effectuées sur des bases de données réelles afin de voir l'impact du fait de lisser cette séparation rigide. Notre méthode améliore les résultats de classification par rapport à l'algorithme standard des cellules dendritiques.

Mots clés: Systèmes immunitaires artificiels, Théorie du danger, Cellules dendritiques, Théorie des ensembles flous.

Acknowledgments

I would like to thank my advisor 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 competence and his generosity have allowed me to work in ideal conditions. His comments and his observations have always been of great interest. I thank him for his support and for his constant encouragements.

Heartfelt thanks to all of my friends for their encouragement.

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. Lastly, and most importantly, I wish to thank my parents Fradj and Souad for bearing me, raising me, supporting me, teaching me, and loving me. To them I dedicate this master thesis.

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

Contents

List of Figures

List of Tables

Introduction

Although we are in permanent contact with innumerable germs in the environment of which some are pathogenic, the infections which we develop are relatively rare. The reason is that our organism has multiple means of defence which constitute the immune system (IS).

The main task of the IS is discriminating self (defined early in life) and nonself (anything that comes later, i.e infectious foreign cells and substances). This process is called "self-nonself discrimination".

Several artificial immune system (AIS) applications have traditionally performed self-nonself discrimination such as the negative selection algorithm (NSA). However, in (Stibor, 2006), criticisms of the NSA have been mentioned by emphasizing that NSA could not function appropriately because it was based on a simplified version of the immunological self-nonself theory. It suffers from issues such as false positives, problems with detector generation/holes, the need for an initial learning phase, etc. Hence, (Aickelin & McLeod, 2003) propose a relatively newer immunological discovery as a possible alternative known as the Danger Theory (DT).

The DT overlaps now the way aiming at designing more efficiently a new foundation of artificial immune systems. The most prominent players of the DT are the "dendritic cells" (DCs). An apprehension of dendritic cells behavior led to the development of an inspired immune system algorithm termed the "dendritic cell algorithm" (DCA).

The DT using the DCA is applied to a wide range of applications such as anomaly detection (Greensmith & Aickelin, 2006), bot detection (Greensmith & Aickelin, 2008), syn scan detection (Greensmith & Aickelin, 2007b), etc.

Indeed, the DCA is used as a classifier for a static machine learning data set (Aickelin & Cayzer, 2005), as it has been proved that it can process data classification, but is at the same time sensitive to the data order. This is due to an environment characterized by a crisp separation between normality (the semi-mature context) and abnormality (the mature context). This absurd separation badly affects the classification task of the algorithm. Besides, the standard DCA uses imprecise terms such as "mature" and "semi-mature". These terms control the classification task of the DCA. Thus, it seems necessary to handle this imprecision. One possible technique for handling imprecision is the fuzzy set theory.

Actually, there are many works dealing with the hybridization of AIS as well as DT with fuzzy set theory such as the FAIS (Nasraoui & Dasgupta, 2002), FAIRS (Xu, 2006) and others like in (Polat & Kodaz, 2006), (Jaradat & Langari, 2008), (Visconti & Tahayori, 2008), (Mezyk & Unold, 2009), etc. These works, generally try to develop several new enhancements by introducing imprecision into their systems to deal with some of their weaknesses. Other works such as (Fu & Li, 2008), (Fu & Zhang, 2009) and (Aickelin & Cayzer, 2005) 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 by a fuzzy one, etc.

As an inspiration from these works, in this master thesis, we propose to develop a fuzzy dendritic cell method (FDCM), a new classification technique based on dendritic cells within the framework of fuzzy set theory. Our FDCM aims to smooth the crisp separation between the semi-mature context and the mature context, since we can neither identify a clear boundary between them nor quantify exactly what is meant by "semi-mature" or "mature". Hence, this hybridization will improve the classification accuracy taking into account the mentioned limitations of the DCA.

Our FDCM differs in some way from the previous works, which are based on the mentioned hybridization, since it requires specific hypotheses such as working on a signal database. In addition, our FDCM sheds more light on the DCA's context assessment phase.

Dealing with problems under imprecision using fuzzy set theory seems promising since it allows to handle the complicated systems in simple way. This is the

main reason why fuzzy set theory is widely applied in various domains. It is applied to solve a great diversity of problems in engineering, business, medical and related health sciences, natural sciences, and so on.

This report consists of four chapters belonging to two main parts:

Part I: *Theoretical aspects*. This part presents theoretical aspects regarding fuzzy set theory and danger theory which are detailed, respectively, in Chapter 1 and Chapter 2.

Part II: *Danger theory based on fuzzy set theory*. In this part, Chapter 3 deals with the definition of our fuzzy dendritic cell method as a new technique associating the dendritic cell algorithm with the fuzzy set theory. Throughout this Chapter, the characteristics of this new approach - namely its definition, its objectives and its representation - are presented. Chapter 4 deals with simulations which have been performed in order to analyze and evaluate results given by the proposed fuzzy dendritic cell method. The major results that we have developed in this part are in (Chelly & Elouedi, 2010).

Finally, a conclusion summarizes all the work presented in this report and proposes further works to improve our method.

An appendix is provided to present the description of data sets used in simulations.

Part I

Theoretical aspects

Chapter 1

Fuzzy Set Theory

1.1 Introduction

Fuzzy set theory was introduced in 1965 by Zadeh (Zadeh, 1965). It is considered as a useful theory for modeling and reasoning with imprecision 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 (H. Zimmermann, 1996). Imprecision in expressions like “low frequency”, “high demand” or “small number” can be called fuzziness.

The applications which may be adapted to fuzzy set theory are wide-ranging. It is used to solve a great diversity of problems in engineering (Ross, 1995), intrusion detection (Shah & Joshi, 2003), sciences (Zadeh, 1994), and so on.

The primary purpose of this chapter is to introduce and to elucidate the fuzzy set theory. This chapter is organized as follows: Section 1.2 covers the definition and notations of fuzzy sets. In the next Section, we present the membership function. Section 1.4 deals with operations on fuzzy sets. Various aspects of fuzzy relations are then detailed in Section 1.5. Section 1.6 focuses on fuzzy composition. Section 1.7 introduces hedges of fuzzy systems and finally, we shed some light on the fuzzy logic concept in Section 1.8. All these concepts are illustrated by examples.

1.2 Fuzzy sets: Definition and notations

Throughout this Section, we start first of all by defining the fuzzy set concept and then we will give the according notations.

1.2.1 Fuzzy set definitions

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* which will be explained in details in the next Section.

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 1.1 *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".*

1.2.2 Fuzzy set notations

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 (see Section 1.3).

Example 1.2 *As mentioned above, the central concept of fuzzy set theory is that the membership function μ can have values between 0 and 1. This is shown in Figure 1.1. If an element x of the universe of discourse X lies within fuzzy set "half-mature", it will have a value between 0 and 1.*

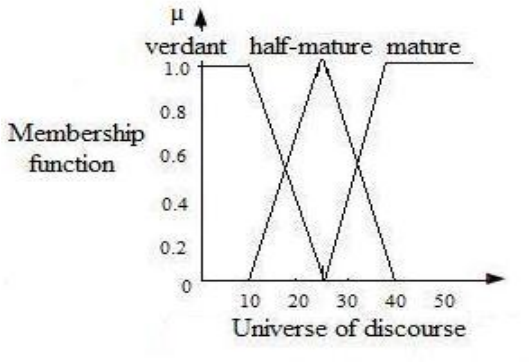

Figure 1.1: Membership functions of a fuzzy set

1.3 Membership functions

1.3.1 Definition

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.

The grade of membership $\mu_A(x_0)$ of a membership function $\mu_A(x)$ describes for the special element x = x_0, to which grade it belongs to the fuzzy set A. This value is in the unit interval [0, 1]. Obviously, x_0 can simultaneously belong to another fuzzy set B, such that $\mu_B(x_0)$ characterizes the grade of membership of x_0 to B. This case is shown in Figure 1.2:

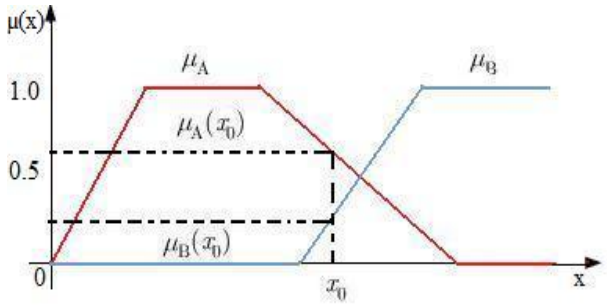

Figure 1.2: Membership grades of x_0 in the sets A and B

As we remark, if x_0 belongs to the fuzzy set A, then $\mu_A(x_0) = 0.75$. However, if it belongs to the fuzzy set B, then $\mu_B(x_0) = 0.25$.

Once we have defined the concept of the membership function, we will cover its properties.

1.3.2 Properties

Throughout this Section, a set of important properties and characteristics of the membership function will be described (Goguen, 1967) (Dubois & Prade, 1997).

The support

The support of a fuzzy set A in the universal set X is the crisp set that contains all the elements of X that have a nonzero membership grade in A. That is, supports of fuzzy sets in X are obtained by Equation 1.1:

$$Supp(A) = \{x|\mu_A(x) > 0, \forall x \in X\} \tag{1.1}$$

The core

The core of a fuzzy set A is the crisp set of all points in the universe of discourse X where the membership function of A is 1 (see Equation 1.2).

$$Core(A) = \{x|\mu_A(x) = 1, \forall x \in X\} \tag{1.2}$$

The height

The height of a fuzzy set is the largest membership grade attained by any element in that set (see Equation 1.3):

$$hgt(A) = supp_{x \in X}\mu_A(x) \tag{1.3}$$

A fuzzy set A is called *normal* when hgt(A) = 1, and it is *subnormal* when hgt(A) < 1.

The boundary

The boundary of a fuzzy set A is the crisp set of all points in the universe of discourse X where the membership function of A is between 0 and 1 (see Equation 1.4).

$$Boundaries(A) = \{x|0 < \mu_A(x) < 1, \forall x \in X\} \tag{1.4}$$

An illustration of all these properties is shown in Figure 1.3.

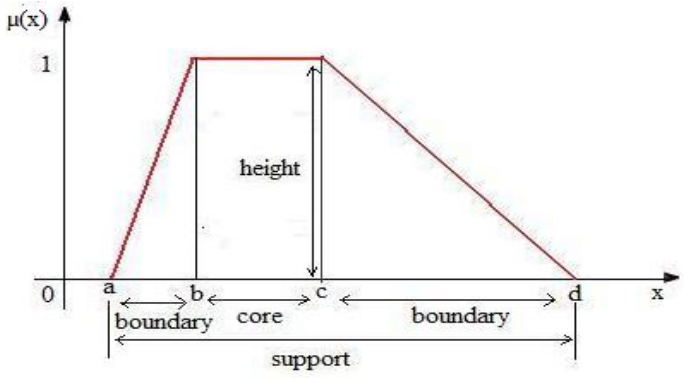

Figure 1.3: Some characteristics of a membership function

1.3.3 Membership function representations

The type of representation of the membership function depends on the universe of discourse. If it consists of many values, or is the base set a continuum, then linear membership functions are preferred, because of their simplicity and efficiency with respect to computability (Fortuna & Graziani, 2007). Mostly, these are triangular (Figure 1.4(a)) or trapezoidal (Figure 1.4(b)) functions.

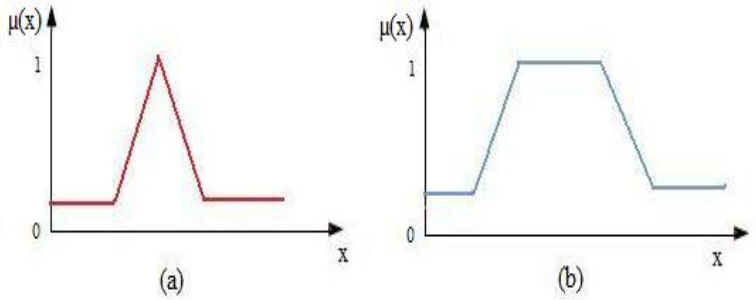

Figure 1.4: Triangular and trapezoidal membership functions

For some applications the modeling requires continuously differentiable curves and therefore smooth transitions which the trapezoids do not have. Three of these functions are mentioned on Figure 1.5:

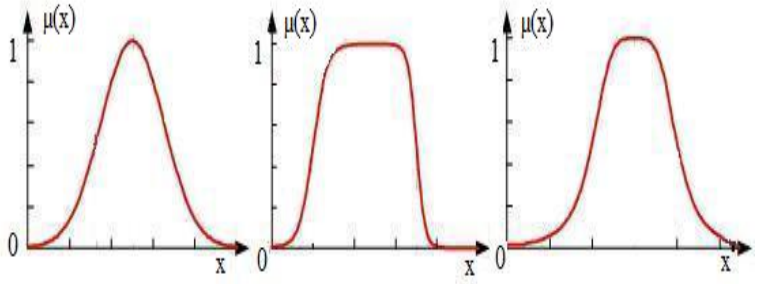

Figure 1.5: Membership functions with smooth transitions

The membership function provides us with a vehicle for developing operations with fuzzy sets. This will be detailed in the following Section.

1.4 Operations on fuzzy sets

The basic connective operations in classical set theory are those of intersection, union and complement. These operations on characteristic functions can be generalized to fuzzy sets (J. Zimmermann, 2001) (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.

1.4.1 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 1.5:

$$c : [0, 1] \rightarrow [0, 1]$$

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \tag{1.5}$$

This could be illustrated by Figure 1.6(a) which shows a fuzzy set A and its complement in Figure 1.6(b).

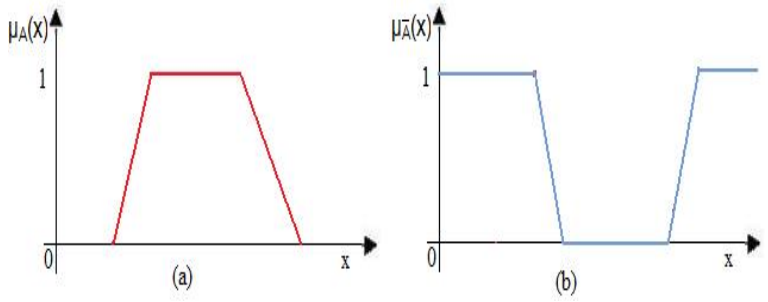

Figure 1.6: A fuzzy set and its complement

1.4.2 Fuzzy intersection

The intersection of two fuzzy sets A and B corresponds to the *Boolean AND* function and is given by Equation 1.6:

$$i : [0, 1] \times [0, 1] \rightarrow [0, 1]$$

$$\mu_{A\cap B}(x) = min[\mu_A(x), \mu_B(x)] \tag{1.6}$$

An illustrative figure is as follows:

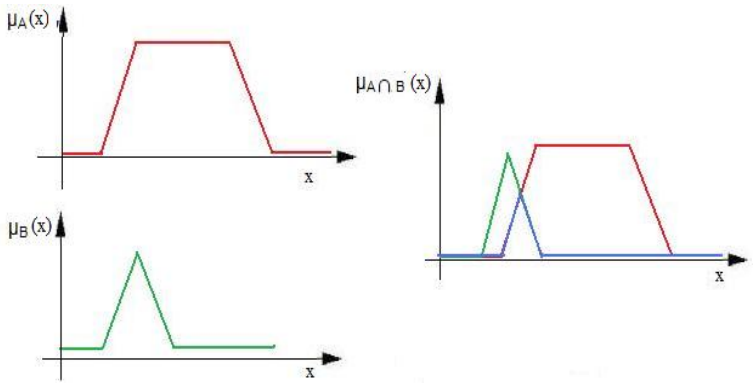

Figure 1.7: Intersection of two fuzzy sets A and B

The intersection (the blue color) between the two fuzzy sets A and B in Figure 1.7 contains the elements shared by these sets. Thus, the degree of membership is the lower membership in both sets of each element.

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 (see Equation 1.7):

$$\mu_{A\cap B}(x) = \mu_A(x) \cdot \mu_B(x), x \in X \tag{1.7}$$

1.4.3 Fuzzy union

The union of two fuzzy sets A and B corresponds to *Boolean OR* function and is given by Equation 1.8:

$$u : [0, 1] \times [0, 1] \rightarrow [0, 1]$$
$$\mu_{A\cup B}(x) = max[\mu_A(x), \mu_B(x)] \tag{1.8}$$

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 1.9:

$$\mu_{A\cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x) \tag{1.9}$$

Figure 1.8 shows the union (the blue color) of the two previous fuzzy sets A and B. The union consists of every element that falls into either set.

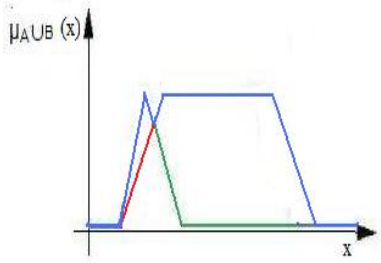

Figure 1.8: Union of two fuzzy sets A and B

1.4.4 Fuzzy equality

Two fuzzy sets A and B are equal if they have the same membership function within a universe of discourse X (see Equation 1.10).

$$\text{A} = \text{B}, \text{ iif } \mu_A(x) = \mu_B(x) \tag{1.10}$$

1.4.5 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 ∪ B = B ∪ 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
  * $\neg(A \cap B) = \neg A \cup \neg B$
  * $\neg(A \cup B) = \neg A \cap \neg B$

1.5 Fuzzy relations

An important aspect of fuzzy set theory is the ability to relate sets with different universes of discourse. Thus, we talk about *fuzzy relations* (Deschrijver & Kerre, 2003). First, relations are explained by a simple example using discrete fuzzy sets.

Example 1.3 *Let us describe the relationship between the color of a fruit x and the grade of maturity y and characterize the linguistic variable color by a crisp set X with three linguistic terms as:*

$$X = \{green, yellow, red\}$$

and similarly the grade of maturity as:

$$Y = \{verdant, half\text{-}mature, mature\}$$

The crisp formulation of a relation $X \rightarrow Y$ between the two crisp sets would look like this in Table 1.1.

Table 1.1: A crisp set built from two crisp base sets (X and Y)

	verdant	half-mature	mature
green	1	0	0
yellow	0	1	0
red	0	0	1

The zeros and ones describe the grade of membership to this relation. This relation is now a new kind of crisp set that is built from the two crisp base sets X and Y. This new set is now called R and can be expressed also by the rules:

(1) IF the color is green THEN the fruit is verdant
(2) IF the color is yellow THEN the fruit is half-mature
(3) IF the color is red THEN the fruit is mature

As can be seen from this example, a relation, which is called a *rule* or *rule base*, can be used to provide a model.

This crisp relation *R* represents the presence or absence of association, interaction or interconnection between the elements of these two sets. This can be generalized to allow for various degrees of strength of association or interaction

between elements. Degrees of association can be represented by membership grades in a fuzzy relation in the same way as they are represented in a fuzzy set.

Example 1.4 *Applying this to the fruit example, Table 1.1 can be modified in such a way that there are now real numbers in [0, 1].*

Table 1.2 represents a fuzzy relation and models the connectives in a fuzzy rule base. It is a two-dimensional fuzzy set and the question now is, how can this set be determined from its elements?

Table 1.2: A fuzzy relation

	verdant	half-mature	mature
green	1	0.5	0
yellow	0.3	1	0.4
red	0	0.2	1

In order to determine the set from its elements, the elements are generalized. In the example above, the linguistic terms were treated as crisp terms. For instance, when we represent the colors on a color spectrum scale, the colors would be described by their spectral distribution curves that can be interpreted as membership functions, then, a particular color is a fuzzy term. Treating also the grades of maturity as fuzzy terms, the above relation is a two-dimensional fuzzy set over two fuzzy sets.

Example 1.5 *For instance, we take from the fruit example the relation between the linguistic terms red and mature, and represent them by the membership functions as shown in Figure 1.9.*

A fruit can be characterized by the property red AND mature. This expression can be re-written in mathematical form using elementary connective operators for the membership functions by Equation 1.11:

$$\mu_R(x, y) = min\{\mu_A(x), \mu_B(y)\} \tag{1.11}$$

or by Equation 1.12:

$$\mu_R(x, y) = \mu_A(x) \cdot \mu_B(y) \tag{1.12}$$

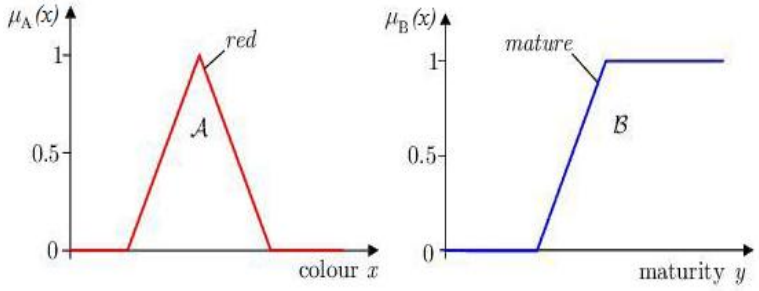

Figure 1.9: Membership function red and mature

Figure 1.10(a) shows a 3-dimensional view of these two membership functions and Figure 1.10(b) illustrates the membership function of the relation after applying the connective operation stated above to (a).

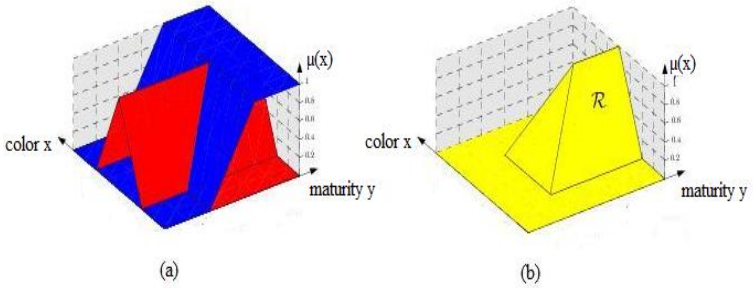

Figure 1.10: Relation between two fuzzy sets

This result combines the two fuzzy sets by an operation that is a Cartesian product:

$$R : X \ltimes Y \rightarrow [0, 1]$$

It is obvious that the connective operation in a rule for the "$\rightarrow$" operation is simply performed by a fuzzy intersection in two dimensions. For this, both intersection operators *min* or *algebraic product* can be used.

For the complete rule base R one can combine the relations formed for each individual rule with a fuzzy union operator, which is the fuzzy OR.

Example 1.6 *When combining rules into a rule base, the example above may be rewritten as:*

(1) IF the color is green THEN the fruit is verdant
OR
(2) IF the color is yellow THEN the fruit is half-mature
OR
(3) IF the color is red THEN the fruit is mature

which describes in a linguistic way a union of three rules.

The combination of the representations of different rules into a representation of a rule base can be specified by applying the union operator by writing the rule base with max/min operators as follows (see Equation 1.13):

$$\mu_R(x_1, x_2, \ldots, x_n, \mu) = max_R\{min\{\mu_{P_r}(x_1, x_2, \ldots, x_n), \mu_{B_r}(u)\}\} \tag{1.13}$$

where $\mu_{P_r}(x_1, x_2, \ldots, x_n)$ is the premise of r^{th} rule. This representation is the standard max/min representation of a rule base. Instead of the max/min representation, a so called *max-prod* representation is also usual where the algebraic product (see Equation 1.14)

$$\mu_R(x_1, x_2, \ldots, x_n, \mu) = max_R\{\mu_{P_r}(x_1, x_2, \ldots, x_n) \cdot \mu_{B_r}(u)\} \tag{1.14}$$

is used to build the relation between the premise and the conclusion.

1.6 Fuzzy compositions

Throughout this Section, a new fuzzy operation type is introduced. It allows us to operate with a given fact and a fuzzy relation in order to produce an output that represents the decision in a fuzzy way (Deschrijver & Kerre, 2003).

This operation is called *fuzzy reasoning*, which is a special case of the more general operation called *fuzzy composition*. We can explain the need for the fuzzy reasoning method by the following example.

Example 1.7 *Taking again the fruit example. It is assumed that one has a crisp fact: a green fruit. The decision from the rule base is obvious: the fruit is verdant, and this is similar for the other facts: yellow and red. However, if one*

has a fact like: the fruit is orange, one does not know how to determine which rule fires the decision and what the decision is.

Let R and S be two relations of the forms:

$$R : X \ltimes Y \rightarrow [0, 1]$$
$$S : Y \ltimes Z \rightarrow [0, 1]$$

These two relations can be composed to one relation T:

$$T : X \ltimes Z \rightarrow [0, 1]$$

This process is known as *composition* and, using the max and min operators for union and intersection, one can express the composition operation $T = R \circ S$ by the corresponding membership functions (see Equation 1.15):

$$\mu_T(x, z) = max_{y \in Y}\{min\{\mu_R(x, y), \mu_S(y, z)\}\} \tag{1.15}$$

Example 1.8 *When one takes the above fruit example again with the color-maturity relation R (Table 1.3) and define for S a maturity-relation (Table 1.4),*

Table 1.3: Color maturity relation R

R	verdant	half-mature	mature
green	1	0.5	0
yellow	0.3	1	0.4
red	0	0.2	1

Table 1.4: Taste maturity relation S

S	sour	tasteless	sweet
verdant	1	0.2	0
half-mature	0.7	1	0.3
mature	0	0.7	1

then by applying the max and min operators expressed by Equation 1.15 to the elements of these two tables, Table 1.5 is obtained.

Table 1.5: Composition relation T

$T = R \circ S$	sour	tasteless	sweet
green	1	0.5	0.3
yellow	0.7	1	0.4
red	0.2	0.7	1

When the fuzzy set S is now interpreted as a rule base and the fuzzy set R as a fact obtained from some measurement data, then the fuzzy set T is the result of the reasoning process, which is in this case a relation.

In the same manner as relations can be composed, the one-dimensional facts can be composed with the rule base to realize the reasoning operation. This can now be precisely re-formulated.

Let R be the rule base:

$$R : X \ltimes Y \rightarrow [0, 1]$$

its membership function $\mu_R(x, y)$ (see previous Section) and if there is a fact described by the fuzzy set:

$$A^{'} : X \rightarrow [0, 1]$$

and its membership function $\mu_{A'}(x)$, the result:

$$B^{'} = A^{'} \circ R : Y \rightarrow [0, 1]$$

of the fuzzy reasoning is represented by the membership function (see Equation 1.16):

$$\mu_{B'}(z) = max_{x \in X}\{min\{\mu_{A'}(x), \mu_R(x, y)\}\} \tag{1.16}$$

Example 1.9 *Define the fruit color green as a fact by the singleton:*

$$C^{'} = \{1\ 0\ 0\}$$

where the numbers are the intensity grades of the colors green, yellow and red. When one calculates the composition $T^{'} = C^{'} \circ R$ by applying the composition formula, where in this case the first operand has only one dimension, the fuzzy set for the maturity

$$T' = \{1\ 0.5\ 0\}$$

is obtained. The result is obvious from the first rule of the rule base. When a different color is taken then included in the rule base entries, say orange as

$$T' = \{0\ 0.5\ 0.5\}$$

then there is no problem to obtain the value for the maturity

$$T' = \{0.3\ 0.5\ 0.5\}$$

by applying the composition formula. The reasoning process is now solved.

The fuzzy composition is elucidated in this Section. However, it is still necessary to be closer to the natural language. This could be achieved by the fuzzy set theory since it offers the appropriate operators. These will be explained in the following Section.

1.7 Fuzzy set hedges

Another important feature of fuzzy set theory is the ability to define *hedges*, or *modifiers* of fuzzy values. These operations are provided in an effort to maintain close ties to natural language (Roth & Mervis, 1983). Examples of such operations are: *very*, *little*, *more or less*, *definitly*, *sort of*, *somewhat* and so on.

The definition of hedges is entirely subjective, but their operation is consistent: they serve to modify the meaning of a term and to transform membership/truth values in a systematic manner according to standard mathematical functions.
For instance, hedges *very*, *extremely* and *slightly* are usually defined respectively as follows:

$$\mu_{veryA}(x) = \mu_A(x)^2$$
$$\mu_{extremelyA}(x) = \mu_A(x)^3$$
$$\mu_{slightlyA}(x) = \mu_A(x)^{1/3}$$

Example 1.10 *We take the fruit example again with the linguistic variable maturity with its three linguistic terms "verdant", "half-mature" and "mature" in the universe of discourse Y. We want to transform the statement "The fruit is verdant" to "The fruit is very verdant", "The fruit is extremely verdant" and to*

"The fruit is slightly verdant".

If the fuzzy set verdant takes $\mu_{verdant}(x) = [1, 0.3, 0]$ *and by applying "very", "extremely" and "slightly" hedges then we get the following mathematical functions.*

$$\mu_{very-verdant}(x) = \mu_{verdant}(x)^2 = [1,\ 0.09,\ 0]$$
$$\mu_{extremely-verdant}(x) = \mu_{verdant}(x)^3 = [1,\ 0.027,\ 0]$$
$$\mu_{slightly-verdant}(x) = \mu_{verdant}(x)^{1/3} = [1,\ 0.67,\ 0]$$

1.8 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 1.11.

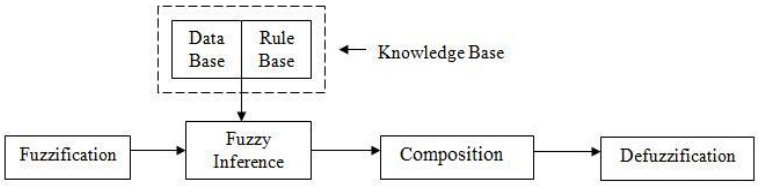

Figure 1.11: Fuzzy Logic Controller

The FLC is composed of five boxes which are explained in the next sub-Sections.

1.8.1 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.

1.8.2 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.

1.8.3 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).

1.8.4 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.

1.8.5 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).

Example 1.11 *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 1.12.*

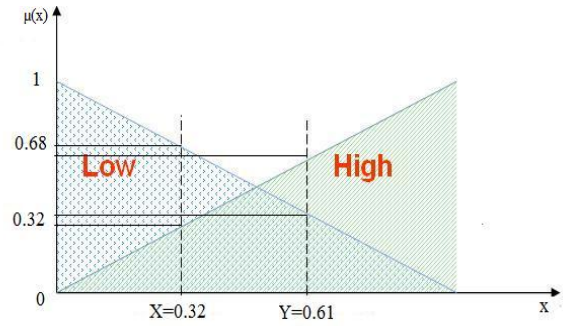

Figure 1.12: Defining the membership functions value's of each input

  - *Suppose that the crisp inputs are x = 0.32 and y = 0.61. We project both of the inputs in 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 1.13(a)) to get a new membership function (Figure 1.13(b)).*

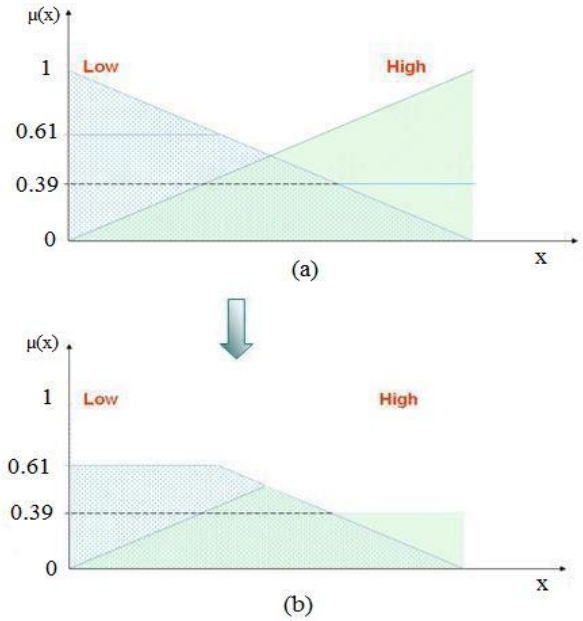

Figure 1.13: Composition process

- *The defuzzification process:*
  By applying the centroid method we get the following crisp output (see Figure 1.14):

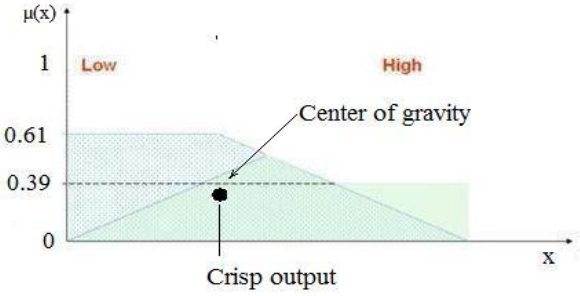

Figure 1.14: Centroid method for defuzzification

1.9 Conclusion

In this chapter, we have elucidated the basics of fuzzy set theory which is a generalization of the classical set theory which offers a natural model to handle uncertain information. In the first part, we have introduced the basic elements of this theory. Then, in the second part, we have shed some light on the fuzzy logic.

Fuzzy set theory can be applied in several techniques such as the *artificial immune system*, especially the *danger theory* which will be the object of the next chapter.

Chapter 2

The Danger Theory

2.1 Introduction

Although we are in permanent contact with innumerable germs in the environment of which some are pathogenic, the infections which we develop are relatively rare. The reason is that our organism has multiple means of defence which constitute the immune system (IS).

The main task of the (IS) is 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. The system is able to discern differences between foreign, and possibly pathogenic, invaders and non-foreign molecules by a process called "self-nonself discrimination" (Janeway, 1992) which is the basis of the IS.

The efficient mechanisms of a biological immune system are able to perform several tasks such as learning (Hunt & Cooke, 1996), classification (Secker & Timmis, 2003a), optimization (Chun & Hong, 1997), etc. These remarkable characteristics and capabilities have caught the attention of many researchers and have led to the development of new algorithms inspired by the immune system. These algorithms gave rise to a new branch of computational intelligence known as *Artificial immune system (AIS)* (Hofmeyr & Forrest, 1999) (Hofmeyr & Forrest, 2000).

Nowadays, AIS is emerging as an active and attractive field involving models,

techniques and applications of great diversity (Hart & Timmis, 2005) (Andrews & Timmis, 2005). Several AIS applications have used the self non-self theory such as the negative selection algorithm (NSA). Nevertheless, it was proved in (Stibor, 2006) that the NSA has serious limitations such as false positives, problems with detector generation/holes, the need for an initial learning phase, etc. These issues arise since the NSA was based on a simplified version of the immunological self-nonself theory.

In (Aickelin & McLeod, 2003), a new immunological theory came to light as a possible alternative to the NSA known as the *Danger Theory* (DT).

The DT is based on the behavior of special cells called the "dendritic cells" (DCs). This led to the development of an inspired immune system algorithm termed the "dendritic cell algorithm" (DCA).

In order to understand the danger theory, we have to study, first of all, the several characteristics and the behavior of our immune system. This will be dealt with in the following Section. In Section 2.3, we introduce the basic characteristics of the AIS. In Section 2.4, we focus on the classification task covered by the AIS. We present the danger theory concepts in Section 2.5. Then, in Section 2.6, we describe the dendritic cells followed by a definition of signals and antigen in Section 2.7. Section 2.8 emphasizes the dendritic cell model and finally, danger theory procedures are detailed in Section 2.9.

2.2 Overview of the biological immune system

The immune system is made up of a network of cells, tissues and organs that work together to protect the body against germs and microorganisms every day. Its main task is 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. The system is capable of distinguishing the non-self from our *self* cells by a process called *self-nonself discrimination* (Janeway, 1992). Furthermore, it can remember each infection, so that a second exposure to the same antigen is dealt with more efficiently. Another characteristic is its ability to react faster to any structurally related antigen. This phenomenon is called *"Cross-reactivity"* response.

The biological immune system prevents antigens from harming our body. This is achieved through several lines of defense of the immune system. Hence, we talk about a multilayered system. The protection layers can be divided as: *physical barriers* such as the skin and the respiratory system; *physiological barriers* such as destructive enzymes and stomach acids; and the immune system, which can be broadly divided into two heads which are *Innate Immunity* and *Adaptive Immunity*. They are interlinked and they influence each other.

2.2.1 The innate immunity

The *innate immunity* is present at birth. It has the ability to recognize some microbes and react against them by destroying these pathogens on the first encounter. The innate immunity works as follows:

Physiological conditions such as pH, temperature and a variety of chemicals provide unsuitable living conditions for foreign pathogens. Some specialized cells have also the ability to capture the foreign microorganism by a process called *"phagocytosis"* in order to ingest it. Another aspect of the innate immunity is the production of proteins, called *"cytokines"*, allowing cells to communicate with each other (Mayer, 2005).

The innate immunity protects the body *non specifically* (in contrast to the adaptive immunity). It gives the same type of response to any pathogen and it is not capable of recognizing or producing a specific response to a specific invader (McEnery, 2008). The innate immunity also plays a leading role in the boost of the adaptive immunity.

2.2.2 The adaptive immunity

It is also called *acquired* or *specific* immunity. It allows the immune system to launch an attack against any antigen that the innate immunity can not remove. The adaptive immunity is managed by white blood cells, specifically, *T cells* and *B cells* (Janeway & Travers, 2001).

T-cells are of three types namely *T helper cells* which are essential to the activation of B-cells, *killer T-cells* which bind to foreign invaders and inject poisonous chemicals into them causing their destruction, and *suppressor T cells* which inhibit

the action of other immune cells, thus preventing allergic reactions and autoimmune diseases.

B-cells are responsible for the production and secretion of antibodies, which are specific proteins that bind to the antigen. Each B-cell can only produce one particular antibody. The antigen is found on the surface of the invading organism and the binding of an antibody to the antigen is a signal to destroy the invading cell.

The acquired immunity is subdivided into two heads: the *humoral* immunity and *cellular* immunity (Alder & Pancer, 2005).

Humoral immunity

It is arbitrated by antibodies. The humoral branch of the immune system involves the interaction of B cells with antigens. This mechanism leads to the proliferation (clones) and differentiation of B cells into antibody secreting *"plasma cells"*. These antibodies - by binding to the antigens - facilitate their elimination. The cloning of B cells also leads to the production of "memory cells". They are able to live longer than plasma cells so that they can remember specific intruders and respond quickly following a second expose to the same antigen (Vos & Lees, 2000).

Cellular immunity

It is based on the activation of T cells and the release of various cytokines which activate various phagocytic cells, enabling them to phagocytose and kill microorganisms more effectively. This type of immunity response is especially important in host defense against intracellular bacteria (Snapper & Mond, 1996).

2.3 Functions of the artificial immune system

The characteristics of the artificial immune system include (Castro & Timmis, 2003):

- **Pattern recognition:** Cells and molecules of the immune system have several ways of recognizing specific antigens (patterns) and they generate appropriate responses. This is accomplished by a recognition mechanism

based on the chemical binding of receptors and antigens. This binding depends on the molecular shape of cells.

- **Feature extraction (noise tolerance):** In general, antibodies bind to a portion of the antigen, rather than to the whole antigen. In this way, the immune system can recognize an antigen just by matching segments of it.
- **Learning and Memory:** The system can "learn" the structures of pathogens, so that future responses to the same pathogens are faster and stronger.
- **Self-regulation:** Depending on the severity of the attack, the response of the immune system can range from very light and almost imperceptible to very strong. A stronger response uses a lot of resources to help repel the attacker. Once the invader is eliminated, the immune system regulates itself in order to stop the delivery of new resources and to release the used ones.
- **Self-protection:** By protecting the whole body the immune system is protecting itself. It means that there is no other additional system to protect and maintain the immune system

One more important characteristic of the AIS is its ability to offer suitable algorithms like the *clonal selection algorithm* (Castro & Zuben, 2000) (Castro & Zuben, 2002) (Kim & Bentley, 2002) and the *immune network algorithm* (Chun & Hong, 1997) (Huang, 2002) (Cayzer & Aickelin, 2002b) designed for and applied to difficult problems such as intrusion detection (Kim & Bentley, 2001) (Breunig & Albert, 2002) (Shulin & Wenhu, 2002) (Dasgupta & Majumdar, 2002), data clustering (Younsi & Wang, 2004) (Secker & Timmis, 2003a), classification (Wu & Chung, 2005), (Greensmith & Cayzer, 2003) and search problems (Tay & Kwoh, 2005), (Derakhshanfar & Minaei-Bidgoli, 2009), etc. We will focus on the classification task.

2.4 Artificial immune system classification

AIS offers two main algorithms to do the classification task named the *artificial immune recognition system (AIRS)* and the *negative selection algorithm (NSA)*.

2.4.1 The artificial immune recognition system

The Artificial Immune Recognition System (AIRS) introduced in (Watkins & Timmis, 2004) exhibited initial success as a classification algorithm.

In AIRS, there are two different populations, the Artificial Recognition Balls (ARBs) and the memory cells. If a training antigen is presented, ARBs (lymphocytes) matching the antigen are activated and awarded more resources. Through this process of stimulation, mutation and selection a candidate memory cell is selected which is inserted to the memory cell pool if it contributes enough information. This process is repeated for all training instances and finally classification takes place by performing a nearest neighbor search on the memory cell population (Meng & Wang, 2005). This system has been proven to perform well in data mining tasks and other non-linear classification tasks (Watkins & Timmis, 2004).

Another classification technique which is the fuzzy artificial immune recognition system (FAIRS) which integrates AIRS and a fuzzy classification scheme called E-algorithm, is developed in (Xu, 2006). FAIRS aims to quickly develop inference rules (with sufficient flexibility in rule length) for classification tasks.

2.4.2 The negative selection algorithm

The negative selection algorithm is a classification algorithm which uses the self-nonself principals.

This algorithm creates a set of randomly generated "detectors" which stand for the system's normal behavior, then selects those which deviate from normal. This results in a detector set tuned to only respond to "non-self" or anomalous strings. In order to explain the functioning of the negative selection algorithm, we take the following example.

Example 2.1 *As mentioned above, the negative selection algorithm (NSA) starts with defining a set of normal behavior (self set). Then, in the case of classifying a new instance, the NSA verifies the attributes' values of the new object. If they belong to the self set (already defined), then the new instance is classified as normal, else it is classified as anomalous. Let us take a simple example reflecting the management of bank credits. The self set in this case reflects the possibility of the client to have a credit. In other words, each client having the same attributes'*

values as in the self set is allowed to have a credit. The self set defined by the bank is presented in Table 2.1:

Table 2.1: Self set defined by the bank

Client	Age	Income	Number of credit cards	Duration of the loan
Client1	25	800	1	20
Client2	30	1000	3	10
Client3	36	1300	3	8
Client4	20	600	1	20
Client5	32	900	2	13
Client6	33	1100	4	9

Once the set of clients having a normal behavior is defined, the NSA is ready to classify any new instance. Let us consider the classification of the following object (see Table 2.2):

Table 2.2: Object to be classified

Client	Age	Income	Number of credit cards	Duration of the loan
Client7	25	400	1	20

As we remark, the client "Client7" does not belong to the self set since his attributes' values are different from those in the self set. Hence, Client7 is not allowed to have a credit from the bank. He is considered as a non-self.

The negative selection algorithm proved to have a number of shortcomings. The nature by which the detectors are generated relies on the initial creation of a sufficient amount of detectors to cover the potential self-nonself feature space. Obviously, as the dimensionality or size of this feature space increases, the number of detectors required to fully cover such space increases exponentially. This has been proven both experimentally (Kim & Bentley, 2001) and theoretically (Stibor & Jimmis, 2006). In addition to such scaling problems, the algorithm also is prone to the generation of false alarms. These misclassification errors are thought to arise partially due to the fact that it is difficult to accurately represent what is "normal".

Hence, a new theory emerged as an alternative to the negative selection paradigm known as the *Danger Theory (DT)*.

2.5 The danger theory

The *Danger Theory* (DT), is a new theory which has become popular amongst immunologists. Its chief advocate is Matzinger (Matzinger, 2001). DT was proposed to explain current anomalies in the understanding of how the immune system recognizes foreign invaders.

2.5.1 Basic concepts

As mentioned previously, the immune system discriminates between self and non-self by attacking foreign antigens and by being tolerant to self. However, the original self-nonself theory (the NS) did not fit in experimental observations. Thus, by the mid-1990s, immunologist had made several modifications to the self-nonself theory which led to the introduction of the danger theory.

The danger theory 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 and certain types of tumors that are fought by the immune system (both attacks against self) and successful transplants (no attack against non-self).

Matzinger concludes that the immune system actually discriminates “some self from some non-self” (Matzinger, 2001). The author asserts that the danger theory introduces not just new labels, but a way of escaping the semantic difficulties of self and non-self, and thus provides grounding for the immune response. Hence, we can take care of “non-self but harmless” and of “self but harmful” invaders

into our system.

The central idea in the danger theory is that the immune system does not respond to non-self but to danger. Thus, just like the self-nonself theory, it fundamentally supports the need for discrimination. However, it differs in the answer to what should be responded to. Instead of responding to foreignness, the immune system reacts to danger.

2.5.2 Signals under apoptosis and necrosis

Among the many potential definitions for "danger", is the following one: "danger is anything that causes cell stress or lytic cell death" (Cayzer & Aickelin, 2002a). We find death in the thymus, the bone marrow and blood, the brain, the skin, the gut and liver. But this is normal, programmed cell death called "apoptosis" and the dying cells are scavenged by specialized cells. This sort of death does not appear dangerous to the immune system.

In the case of apoptosis, cells that undergo suicide, send out signals to nearby scavenger cells (phagocytes), which helps prevent the dying cell from releasing harmful toxins (see Figure 2.1(a)).

However, cells can also die due to "necrosis"; which means that they get killed accidentally by harmful pathogens.

In the case of necrosis, the cell death is not organized. The 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 unorganized chemical reaction (see Figure 2.1(b)).

Danger Theory was built on the concept that the intracellular contents released by damaged cells were actually a form of danger signal that alerted the nearby *antigen-presenting-cells* (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. Dendritic cells play the leading role in the danger theory.

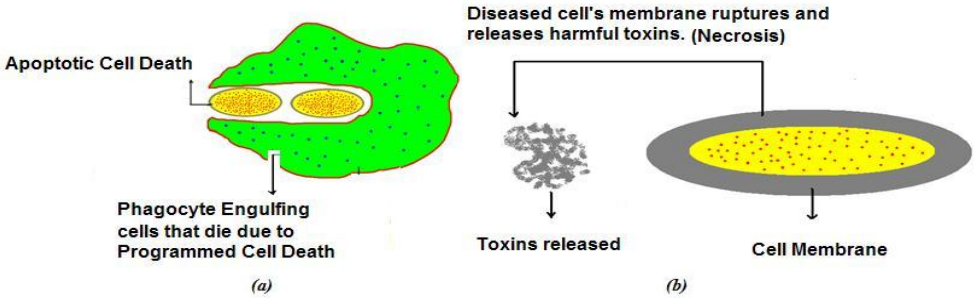

Figure 2.1: Apoptosis and Necrosis illustrations

2.6 Introducing dendritic cells

Dendritic cells (DCs) are antigen presenting cells playing a leading role in the DT. They are responsible for capturing, processing and displaying antigens to T-cells. Furthermore, DCs express receptors on their surfaces to receive signals from their neighborhood. DCs' behavior depends on the concentration of the signals received. Thus, they differentiate into three states termed *immature*, *semi-mature* and *mature*.

2.6.1 Immature DCs

The DC immature state is the initial maturation state of a DC. Immature DCs (iDCs) reside in the tissue where they collect signals and antigen which could be a "safe" molecule or something foreign. Differentiation of iDCs depends on the combination of the various signals received leading to a full or partial maturation state.

2.6.2 Semi-mature DCs

In the presence of cell death (apoptosis) and by exposure to safe signals, iDCs migrate to a terminate semi-mature state known as *semi-mature DCs (smDCs)*. They also migrate from the tissue to the lymph node.

By the receipt of safe signals, smDCs produce a cytokine in response known as *interleukin-10 (IL-10)*. This cytokine suppresses T-cells which match the presented antigen. Hence, causing T-cell tolerance. As a consequence, antigens collected with safe signals are presented in a tolerant context (Greensmith, 2007).

2.6.3 Mature DCs

iDCs migrate to the mature state if they are more exposed to danger signals and PAMPs than safe signals. Thus, they are termed the *mature DCs (mDCs)* drifting from the tissue to the lymph node.

mDCs produce an inflammatory cytokine termed *interleukin-12 (IL-12)* which stimulates T-cells activation in order to be reactive to antigen presentation. Moreover, mDCs produce costimulatory molecules (CSMs) which are known for facilitating the antigen presenting process (Medzhitov & Janeway, 2002).

2.7 Signals and antigen

Various signals urge the behavior of the system since they are a reflection of the state of the environment. There are four categories of signals: *PAMPs*, *danger signals*, *safe signals* and *inflammation*. The combination of these different signals directs the DC population down to two distinct pathways: one causing the activation of the immune system and one responsible for generating peripheral tolerance (Greensmith, 2007).

Dendritic Cells process these signals in order to produce their output signals including a *costimulation signal (CSM)* which shows that the cell is prepared for antigen presentation and two context signals, the *mature* and *semi-mature* output signals (Greensmith & Aickelin, 2004).

2.7.1 PAMPs

PAMPs are essential molecules produced by microbes, but not produced by the host. They are definite indicators of abnormality. In fact, capturing PAMPs signals with a high concentration by a DC, leads to the production of high values of both CSM and the mature output signal. Thus, indicating the presence of a non-host entity.

For instance, in (Greensmith & Aickelin, 2006), the high frequency of networking errors is translated as a high value of PAMP signal.

2.7.2 Danger signals

Danger signals (DS) are signals released as a result of necrosis. They are indicators of abnormality, but with lower value of confidence than PAMPs signals.

The receipt of DS by a DC also causes differentiation to the mature state. Nevertheless, the potency of DS is less than that of PAMP. DS reception causes the presentation of antigen in a dangerous context.

For example, in (Greensmith & Aickelin, 2006), the amount of packets transmitted per second is measured and forms the DS.

2.7.3 Safe signals

Safe signals (SS) 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.

The receipt of SS by a DC causes differentiation to the semi-mature state and the production of CSM in a similar manner as PAMP and DS.

In the situation where tissue contains cells undergoing both apoptosis and necrosis, the receipt of safe signals suppresses the production of IL-12 in response to the danger and PAMP signals present in the tissue. This appears to be one of many regulatory mechanisms provided by the immune system to prevent the generation of false alarms. This is a key mechanism of suppression of the response to antigen not directly linked to a pathogen (Greensmith, 2007).

In (Greensmith & Aickelin, 2006), SS can be derived from the rate of sent/received network packets per second.

2.7.4 Inflammation

Inflammation signals imply the presence of inflammatory cytokines which prove that there is an increase in temperature in the affected tissue. Furthermore, they are evidence that a great number of cells are collected in the tissue area under distress. However, they are insufficient to initiate the maturation of an iDC (Sporri &

Caetano, 2005).

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.

2.7.5 Output signals

After the receipt of the different categories of input signals, DCs produce a set of three output signals:

1- *CSM output*: limits the lifespan of a DC, through being assessed against a *migration threshold*. If this threshold is exceeded, the state of the cell changes from immature to either semi-mature or mature. The cell then enters the antigen presentation stage where its context is assessed.

2- *Semi-mature output*: output incremented in response to safe signals.

3- *Mature output*: output incremented in response to PAMP and danger signals; reduced in response to safe signals (Greensmith & Aickelin, 2004).

2.7.6 Antigen

Antigen is the data that are to be classified, with the basis of classification derived not from the structure of this antigen but from the relative proportions of the different categories of input signals (Greensmith & Aickelin, 2004).

For instance, antigens could be presented by the system calls invoked by running processes (Greensmith & Aickelin, 2007b), or by the process IDs (Greensmith & Aickelin, 2006).

2.8 Overview of the DC model

DCs are known to be antigen presenting cells (APCs) with various functional properties that are interesting and useful to be incorporated into an algorithm.

The different characteristics of DCs are listed below and represented graphically in Figure 2.2 (Aickelin & Cayzer, 2005):

- iDCs have the ability to differentiate in two ways, resulting in mature or semi-mature cells.
- Each iDC can sample multiple antigens within the cell, leading to generalization of the antigen context.
- The collection of antigen by iDCs is not enough to cause maturity. Exposure to certain signals causes the up-regulation of various molecules that initiate antigen presentation.
- Both smDCs and mDCs show expression of costimulatory molecules, inferring that both types have antigen presenting capabilities.
- The cytokines' outputs by mature and semi-mature cells are different, providing contextual information. The concentration of the output cytokines is dependent on the input signals and can be viewed as an interpretation of the original signal strength.

In the same section of tissue, DCs have the ability to sample a finite number of antigens. Thus, an antigen collection threshold is used. Such condition stops DCs from collecting antigens, leading them to migrate from the tissue to the lymph node.

The final state of a DC depends on the concentration of input signals received from the environment. If the concentration of safe signals is greater than the other three categories of signals then DCs migrate to the semi-mature state. However, if the concentration of PAMPs and danger signals is greater than the safe signals then DCs migrate to the mature context.

The same antigen can be presented in both contexts (semi-mature and mature). So in order to determine the final context of an antigen, it is possible to count how many times this antigen had been presented in either contexts. If the antigen is presented more in the mature context than in the semi-mature, then that antigen is classified as anomalous.

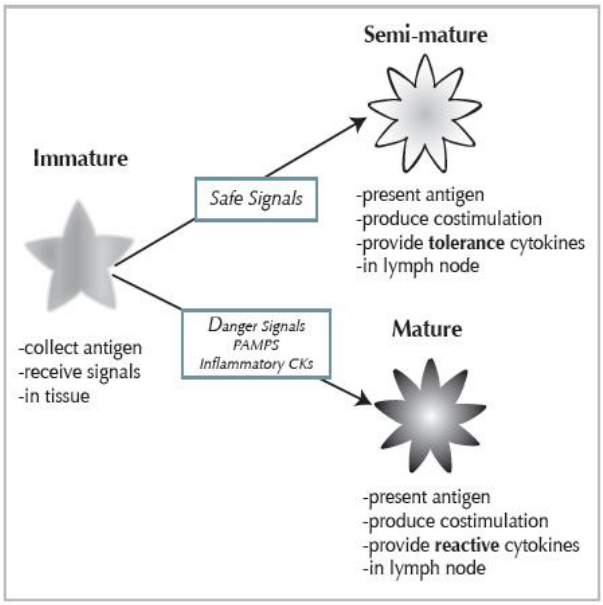

Figure 2.2: DCs behavior. Cytokines (CKs)

The different properties and behavior of DCs are interesting and useful to solve real world classification problems. Hence, we talk about a DC inspired algorithm whose basics are given in the next Section.

2.9 The danger theory procedures

The Dendritic Cell Algorithm (DCA) was first introduced in 2005 (Greensmith & Aickelin, 2005). It has since been applied to two-class classification of a static machine learning dataset (Aickelin & Cayzer, 2005) (Greensmith & Aickelin, 2005) (Greensmith & Aickelin, 2007c).

DCA aims at correlating different data-streams in the form of antigen and signals. In addition, the algorithm attempts to classify groups of identical antigens as *normal* or *anomalous*.

The DCA is not only a classification algorithm. It also allows to know how anomalous a group of antigen is. This is achieved by the generation of an anomaly coefficient value termed the MCAV.

The DCA functions under four levels of abstraction namely the *sampling phase*, the *signal processing phase*, the *context assessment phase* and the *classification phase*.

2.9.1 The sampling phase

Each DC in the system is represented by an object. Throughout the sampling phase DCs are still in their first state: the immature state. iDCs receive and process the set of input signals to produce three output signals (mentioned previously) through a signal processing function.

iDCs keep collecting antigens until they differentiate to either smDCs or mDCs. This differentiation depends on a sufficient exposure to signals which is limited by a migration threshold.

Whilst in the immature state, the DC has three functions which are performed each time a single DC is updated (Greensmith, 2007):

1 *Sample antigen*: the DC collects antigen from an external source (in our case, from the "tissue") and places it in its own antigen storage data structure.

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.

2.9.2 The signal processing phase

iDCs are responsible for the signal processing procedure which is in the form of a weighted sum equation (see Equation 2.1):

$$C_{[CSM,smDC,mDC]} = \frac{((W_{PAMP} * \sum_i PAMP_i) + (W_{SS} * \sum_i SS_i) + (W_{DS} * \sum_i DS_i))}{(W_{PAMP} + W_{SS} + W_{DS})} * \frac{1 + I}{2} \tag{2.1}$$

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 semi-mature output (smDC) and mature output (mDC) signals. These values are cumulatively summed over time (Greensmith, 2007).

The three output signals generated from the weighted sum equation perform two roles. The first role is to limit the sampling duration of a DC, hence if the CSM is greater than the migration threshold (a user defined parameter), iDC migrates to either smDC or mDC. The second role is to determine which final state the iDC should reach. This is achieved by a comparison of the other two output values.

The weights used in the signal processing procedure are either derived empirically from the data or are user defined values.

2.9.3 The context assessment phase

Once the iDC has migrated, the cell context has to be fixed. The cell context is used to label all antigen collected by the DC with the derived context binary value of 1 or 0.

The cell context is assigned as 0 if the semi mature output is greater than the mature output. This means that the antigen collected is likely to be normal, else (if the cell context is assigned as 1) it indicates that the collected antigen may be anomalous.

2.9.4 The classification phase

After the context assessment to each antigen collected by DCs, the MCAV[1] is generated per antigen type for the classification task.

To perform classification, a threshold must be applied to the MCAVs. This threshold could be either a user defined parameter which requires some expert knowledge to define or generated automatically from the data. Hence, if the

[1]Mean Mature Context Antigen Value

MCAV is greater than the anomaly threshold then the antigen is classified as anomalous else it is classified as normal.

2.9.5 The dendritic cell algorithm

The DCA introduced by Greensmith is capable of joining several signals and antigen to assess the context of each object (Greensmith, 2007). Signals pre-categorized as "PAMP", "danger" and "safe" which reflect the input signals of the system are processed by the algorithm, in order to get three output signals: co-stimulation signal (CSM), semi-mature signal (smDC) and mature signal (mDC).

A migration threshold is incorporated in the DCA in order to determine the lifespan of a DC. As soon as the CSM exceeds the migration threshold; the DC ceases to sample signals and antigens. The DCs differentiation direction is determined by the comparison between cumulative smDC and cumulative mDC. If the cumulative smDC is greater than the cumulative mDC, then the DC goes to semi-mature (context=0, DC "thinks" the antigen is normal), otherwise it goes to mature (contex=1, DC "thinks" the antigen is anomalous).

At the end, the mature context antigen value (MCAV), which reflects the probability of an antigen in being anomalous, is calculated. An anomalous threshold is also introduced. Those antigens whose MCAV are greater than the anomalous threshold are classified into the anomalous category, while the others are classified into normal category. The major parts of this algorithm are described as follows.

Algorithm 1 The Dendritic Cell Algorithm

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

2.9.6 The DCA: An example

This example consists of sample calculations of both the signal processing and antigen analysis components for the problem of management of bank credits. The data set for this example is presented in Table 2.3.

Table 2.3: 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
Client6	33	1100	4	9	yes

The dendritic cell algorithm starts by selecting some attributes and pre-categorizing them as PAMP, DS, SS and inflammation. Then, the obtained data set is transformed into a signal data set which is the second step of the DCA. The signal data set (we present only 3 instances) is illustrated in Table 2.4.

Table 2.4: 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 2.4. The weights are presented in Table 2.5.

Table 2.5: Example of weights used for signal processing

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

The signal processing equation is the following:

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

The worked example is performed in the following itemized list:

1. The antigen vector (A) is updated:
   A = {Ag1; Ag1; Ag1; Ag1; Ag1; Ag2; Ag2; Ag2; Ag2; Ag3; Ag3; Ag3}

2. Cycle l = 0:
   DC samples antigen, 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 antigen may be anomalous.

3. The antigen vector now consists of:
   A = {Ag1; Ag1; Ag2; Ag2; Ag3; Ag3; Ag3}

4. Cycle l = 1:
   DC samples randomly selected antigen, 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 antigen is likely to be normal.

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

6. Cycle l = 2:
   DC samples antigen, 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). Even though there are a mixture of signals and the highest signal value comes from the danger signal value, $C_{smDC} < C_{mDC}$ and therefore DC3 is assigned a cell context value of 0. This is due to the negative weight of the safe signal, which has a suppressive effect on the other two categories of signal.

7. Now the antigen can be analyzed and MCAV coefficients derived as shown in Table 2.6.

Table 2.6: 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 0.5. Therefore, client1 (Ag 1) and client2 (Ag 2) are classed as anomalous - they are not allowed to have a credit - however, client3 (Ag3) is classified as normal (Greensmith, 2007).

2.10 Conclusion

In this Chapter, we have briefly introduced the natural immune system; then we have presented the basics and the characteristics of the artificial immune system. In the second part, we have focused on the new branch of AIS which is the danger theory. The main principles and models are elucidated.

Despite the good results provided by the danger theory in a wide range of applications, several researchers are focusing on improving more and more the results of this technique, especially, in an environment where imprecision may exist. Imprecision which could be find in the definition of some words.

The fuzzy set theory presented in the previous Chapter seems to be one of the appropriate formalisms to cope with imprecision. Thus, our objective will be to develop what we call a fuzzy dendritic cell method that will be presented in the following part of this master thesis.

Part II

Danger theory based on fuzzy set theory

Chapter 3

Fuzzy dendritic cell method

3.1 Introduction

The danger theory using the dendritic cell algorithm is considered as one of the most interesting and used techniques in classification tasks. That is why, it is widely applied to a variety of fields notably in web mining (Secker & Timmis, 2003b), robotic (Greensmith & Aickelin, 2007a), bot detection (Greensmith & Aickelin, 2008), etc.

However, the standard dendritic cell method does not perform well its classification task in the case of a disordered contexts (data randomized between class one and class two). This is because that each DC gathers multiple antigens over a period of time. If an iDC differentiates to a mDC, then every antigen contained in that DC is perceived as dangerous (class 2). Similarly, antigens within a smDC are all perceived as safe (class 1). Other explanations could be the environment which is characterized by the crisp boundary in the context assessment phase, which may badly affect the correctness of the classification results, and the use of imprecise terms such as "semi-mature" and "mature". Hence, it shows serious limitations. One possible technique for handling such imprecision is the fuzzy set theory.

In order to deal with problems under imprecision, there are many works introducing the hybridization of DT with fuzzy set theory such as (Fu & Li, 2008), (Fu & Zhang, 2009) and (Aickelin & Cayzer, 2005). Thus, we propose to overcome the limitations of the DCA using the notions of fuzzy set theory. We propose to develop what we call a *fuzzy dendritic cell method (FDCM)*, a new classification

technique based on dendritic cells within the framework of fuzzy set theory. Previous works which are based on such hybridization do not use the same hypothesis as in our FDCM since we shed more light on the DCA's context assessment phase.

This Chapter is consecrated to our development related to the presentation of the danger theory approach under imprecision. Section 3.2 covers the definition and the motivations of our fuzzy dendritic cell method (FDCM). Section 3.3 deals with the objectives of our new method. Then, the different parameters of the fuzzy dendritic cell method are detailed in Section 3.4. Lastly, Section 3.5 sketches an illustrative example of our fuzzy dendritic cell method.

3.2 Definition and motivations

A fuzzy dendritic cell method is a dendritic cell algorithm in a fuzzy environment. The crisp context assessment will be represented and handled by the means of the fuzzy set theory.

As seen in the previous Chapter, in the classical dendritic cell algorithm, the context assessment of each object relies on a crisp boundary. In other words, in order to affect the context (semi-mature or mature) to each object, it is necessary to go by a crisp comparison between the values of these two contexts. Nevertheless, such strict comparison ignores the case where the difference value between the two context values is very low. In such a case, the final context of the object is hard to be defined.

Another issue with the standard DCA is the imprecision found in the definition of some words such as "semi-mature" and "mature" which are quantified numerically. However, we can not affect a precise value to such a term since it is difficult to fix to what extent can we talk about a semi-mature or a mature context.

To overcome these limitations, we introduce our FDCM where fuzzy set theory can handle these issues. In fact, using fuzzy set theory allows us to alleviate the already mentioned crisp separation between the contexts as well as offering the possibility to quantify qualitatively, using linguistic terms, the imprecise words used by the DCA.

3.3 Objectives

The objective of this work is to develop a new concept that we will call the fuzzy dendritic cell method. In addition to the objectives of the standard dendritic cell algorithm, the fuzzy dendritic cell one aims at ensuring three major objectives:

1. Smoothing the abrupt separation of normality (semi-mature) and abnormality (mature) using the fuzziness of fuzzy set theory, since there is no clear boundary between the two contexts.

2. Describing the context of each object using linguistic variables. Fuzzy subsets and the corresponding membership functions describe the context of the object. In other words, ensuring the induction of the fuzzy set theory.

3. Building a knowledge base, comprising rules to support the fuzzy inference.

This new approach is based on both the dendritic cell algorithm and fuzzy set theory in order to cope with the crisp problem, hence, smoothing such a case as well as replacing and describing the context with linguistic variables.

3.4 The fuzzy dendritic cell method parameters

3.4.1 Introduction

As with the standard dendritic cell algorithm, implementing our FDCM falls to the definition of its fundamental phases, namely, the attribute selection and categorization phase, the sampling phase, the signal values derivation phase, the fuzzy context assessment phase and the classification phase. These parameters must take into account the imprecision/fuzziness encountered in the system.

3.4.2 The attribute selection and categorization phase

The attribute selection is based on the semantic of each attribute. Experts are supposed to select a subset of attributes (from the initial data set) and categorize them as PAMPs, danger signals, safe signals and inflammation. Thus, an attribute reduction is achieved, which is one characteristic of the algorithm.

The general guidelines are presented in the list below:

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

3.4.3 The signal value derivation phase

The signal value derivation phase can be broadly divided into two processes. The first one is about calculating PAMPs and SS and the second one is concerned with the calculation of DS.

Process for calculating PAMPs and SS

As stated in the general signal selection rules, both the PAMP and safe signal are positive indicators of an anomalous and normal signal. To achieve this, one attribute is used to form both PAMP and safe signal. This way, we contrive the scenario where the algorithm is given a context of either PAMP or safe signal. Using one attribute for these two signals requires a threshold level to be set: values greater than this can be classed 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 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 DS

A similar process is used to calculate the values for the danger signals. As shown in the general signal selection guidelines, 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, though this is not certain. As part of pre-processing, the mean value for each attribute set is required from the normal class alone (just class 1, not class 1 and class 2).

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 means calculated.

3. Use the calculated distance values in a further calculation to form the single value for the danger signal, DS. This value is the mean value of the absolute distances calculated, with the derivation shown in Equation 3.1:

$$DS = \frac{\sum \text{absolute distances}}{\text{number of attributes}} \tag{3.1}$$

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

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

Additionally, three further specifications must be performed:

- Since each antigen in the data set is unique, appearing only once, it has been proved that this is insufficient for a DCA to function. This is overcome through the use of an antigen multiplier. In other words, each antigen will appear n (a user defined parameter) times in the data set.

- The migration threshold has to be defined.

- The different weights have to be defined for each signal.

3.4.4 The fuzzy process

The fuzzy process consists in the definition of a new model (a fuzzy one) of the standard dendritic cell algorithm taking into account the fact of alleviating and mitigating the crisp assessment phase. The fuzzy procedure is composed of four main steps. This is shown in Figure 3.1:

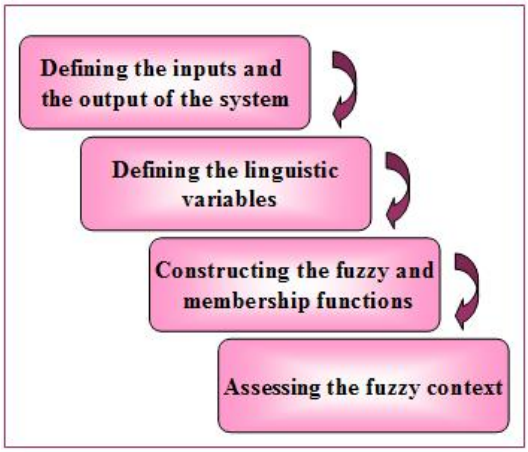

Figure 3.1: Steps of the fuzzy process

Fuzzy system Input-Output variables

As stated above, our objective is to smooth the abrupt separation between normality (semi-mature) and abnormality (mature) using fuzzy concepts since there is no clear boundary between the two contexts. 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 $C_{SemiMature}$ and C_{Mature} are considered as the input variables to the fuzzy system. The final state "maturity" of a DC (object), $S_{Maturity}$, is chosen as the output variable.

All the system's inputs and output are defined using fuzzy set theory.

$$C_{SemiMature} = \{\mu_{C_{SemiMature}}(c_{SemiMature_j})/c_{SemiMature_j} \in X_{C_{SemiMature}}\} \qquad (3.2)$$

$$C_{Mature} = \{\mu_{C_{Mature}}(c_{Mature_j})/c_{Mature_j} \in X_{C_{Mature}}\} \tag{3.3}$$

$$S_{Maturity} = \{S_{Maturity}(s_{Maturity_j})/s_{Maturity_j} \in X_{S_{Maturity}}\} \tag{3.4}$$

where $c_{SemiMature_j}$, c_{Mature_j} and $s_{Maturity_j}$ are, respectively, the elements of the discrete universe of discourse $X_{C_{SemiMature}}$, $X_{C_{Mature}}$ and $X_{S_{Maturity}}$.

$\mu_{C_{SemiMature}}$, $\mu_{C_{Mature}}$ and $\mu_{S_{Maturity}}$ are, respectively, the corresponding membership functions.

Linguistic variables

Basic tools of fuzzy set theory are linguistic variables. Their values are words or sentences in a natural or artificial language, providing a means of systematic manipulation of vague and imprecise concepts. More specifically, a linguistic variable is characterized by a quintuple (x, T(x), U, G, M), where x is the variable name; T(x) is the set of names of the linguistic values of each fuzzy variable x, denoted generically by x and ranging over a universe of discourse U. G is a syntactic rule for generating the names of x values; M is the semantic rule associating a meaning with each value.

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

$$T(S_{Maturity}) = \{Semi - mature, Mature\} \tag{3.5}$$

Each term in $T(S_{Maturity})$ is characterized by a fuzzy subset in a universe of discourse $X_{S_{Maturity}}$.

Semi-mature might be interpreted as an object collected under safe circumstances, reflecting a normal behavior and Mature as an object collected under dangerous circumstances, reflecting an anomalous behavior. Figure 3.2 gives an illustration of $S_{Maturity}$ as a linguistic variable.

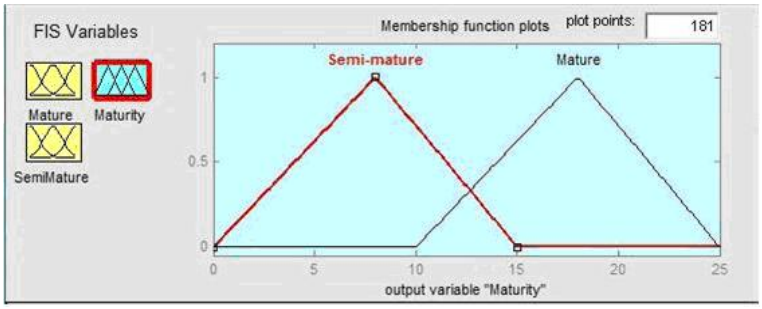

Figure 3.2: Fuzzy membership functions for the Maturity output

Similarly, the input variables $C_{SemiMature}$ and C_{Mature} are interpreted as linguistic variables with:

$$T(Q) = \{Low, Medium, High\} \tag{3.6}$$

where Q = $C_{SemiMature}$ and C_{Mature} respectively.

Fuzzy and membership function construction

In order to specify the range of each linguistic variable, we have generated all the semi-mature and mature values, which are the two outputs generated by the DCA, at the beginning. This generation of these values is done by running once the standard dendritic cell algorithm. Then, we pick up the minimum and maximum values of each of the two generated values to fix the borders of the range. We assume that the extents and midpoints of the membership functions were determined a priori by the user.

The range of the output variable is determined as follows:

$$min(range(S_{Maturity})) = min(min(range\lfloor C_{Mature}\rfloor), min(range\lfloor C_{SemiMature}\rfloor)) \tag{3.7}$$

$$max(range(S_{Maturity})) = max(max(range\lfloor C_{Mature}\rfloor), max(range\lfloor C_{SemiMature}\rfloor)) \tag{3.8}$$

A knowledge base, comprising rules, is built to support the fuzzy inference. The different rules of the fuzzy system are extracted from the information reflecting the effect of each input signal on the state of a dendritic cell which is the following:

– Safe signals: in increase in value is a probable indicator of normality. High values of the safe signal can cancel out the effects of both PAMPs and DS.

– Danger signals: in increase in value is a probable indicator of damage, but there is less certainty than with a PAMP signal.

– PAMPs: in increase in value is a definite indicator of anomaly.

– 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 this information, we can generate a set of rules of the fuzzy system. In fact, the number of rules depends on the number of the membership functions of each input. The number of rules generated is calculated using Equation 3.9:

$$NumberOfRules = \prod_{i=1}^{i=x} \text{Number Of Membership (i)} \tag{3.9}$$

The parameter x locates the number of inputs in the system.

Since each linguistic variable is represented using three membership functions, the number of rules generated is a total of nine.

Rule(1): If (C_{Mature} is Low) and ($C_{SemiMature}$ is Low) then ($S_{Maturity}$ is Mature)

Rule(2): If (C_{Mature} is Low) and ($C_{SemiMature}$ is Medium) then ($S_{Maturity}$ is Semi-mature)

Rule(3): If (C_{Mature} is Low) and ($C_{SemiMature}$ is High) then ($S_{Maturity}$ is Semi-mature)

Rule(4): If (C_{Mature} is Medium) and ($C_{SemiMature}$ is Low) then ($S_{Maturity}$ is Mature)

Rule(5): If (C_{Mature} is Medium) and ($C_{SemiMature}$ is Medium) then ($S_{Maturity}$ is Semi-mature)

Rule(6): If (C_{Mature} is Medium) and ($C_{SemiMature}$ is High) then ($S_{Maturity}$ is Semi-mature)

Rule(7): If (C_{Mature} is High) and ($C_{SemiMature}$ is Low) then ($S_{Maturity}$ is Mature)

Rule(8): If (C_{Mature} is High) and ($C_{SemiMature}$ is Medium) then ($S_{Maturity}$ is Mature)

Rule(9): If (C_{Mature} is High) and ($C_{SemiMature}$ is High) then ($S_{Maturity}$ is Mature)

Let us consider Rule (2) as an example: if the C_{Mature} input is set to its first membership function "Low" and the second input $C_{SemiMature}$ to its second membership function "Medium", then the "Semi-mature" context of the output $S_{Maturity}$ is assigned. This could be explained by the effect of the safe signals (which lead to the semi-mature context) on the judgement of the state of the output, since the high values of the safe signal can cancel out the effects of both PAMPs and DS (which lead to the mature context). The same reasoning is affected to the rest of the rules.

The generated list of rules allows the reasoning over statements in the presence of vagueness, since we cannot exactly quantify what we mean by "SemiMature" or "Mature". It is also a solution to smooth such absurd separation between the two contexts.

Concerning the shape of the membership functions, it is a triangular one. It depends on three scalar parameters a, b, and c, as given by Equation 3.10:

$$f(x,a,b,c) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{c-x}{c-b} & b \leq x \leq c \\ 0 & c \leq x \end{cases} \tag{3.10}$$

The parameters a and c locate the "feet" of the triangle and the parameter c locates the peak. An example is illustrated in Figure 3.3.

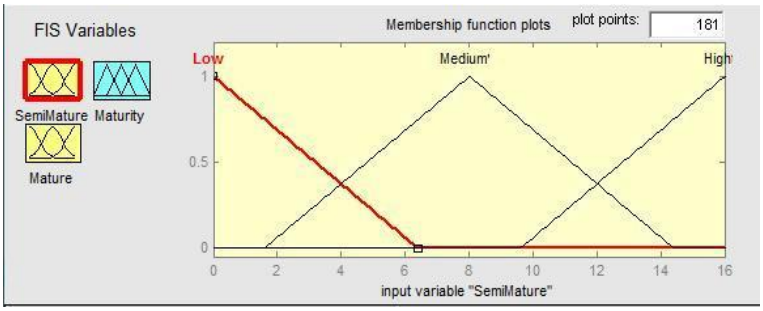

Figure 3.3: Three membership functions of the linguistic variable “SemiMature” in the range [0 16]

To summarize, the fuzzy system is composed of two inputs and one output. The range of both inputs is generated by using the standard DCA, after that, the range of the output is fixed. All the linguistic variables are represented by triangular shape membership functions.

Selecting the fuzzy inference properties

The fuzzy dendritic cell method is based on the *“Min-Max”* inference method. This choice is based on the following reasons:

- The “AND” operator is applied between the linguistic values in the condition part of the rule. Hence, this is mapped as a “Min” operator. It is the fact of selecting the minimum value among the condition memberships of each rule. It could be described as a chain which depends on the lowest value.

- Rules must be combined in some manner in order to make a decision. Hence, the “OR” operator is applied between them. It is mapped as a “Max” operator.

Furthermore, the FDCM is based on the *“centroid defuzzification”* method which is the most popular (the center of gravity of the membership function). This method is the greediest in calculation but gives the most precise results which justify our choice. The centroid defuzzification method is given by Equation 3.11.

$$\sum_{i=1}^{N}(\mu_{(i)} * output(i))/\sum_{i=1}^{N}(\mu_{(i)}) \tag{3.11}$$

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.

To summarize, the fuzzy dendritic cell method is based on the Mamdani inference method and the centroid defuzzifacation mechanism.

The fuzzy context assessment

After defining the fuzzy model, we proceed to the execution of the signal processing phase which is the same as in the standard dendritic cell algorithm seen in the previous Chapter. The outcome of this phase is two different contexts (semi-mature and mature) to compare. These two values represent now the inputs of our new fuzzy method.

Once the inputs are fuzzified and the output (centroid value) is generated, the cell context has to be fixed.

The difference between the context assessment phase in the standard dendritic cell algorithm and the fuzzy context assessment phase in this new approach resides in the comparison of the two contexts. In fact, in our method, the output is compared to the middle of the output 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). In other words, the area of the "Mature" membership function is greater than the "Semi-Mature" one. 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). In other words, the area of the "Semi-Mature" membership function is greater than the "Mature" one. Hence, the final state/context of the object is "Semi-Mature". This means that the antigen collected is likely to be normal.

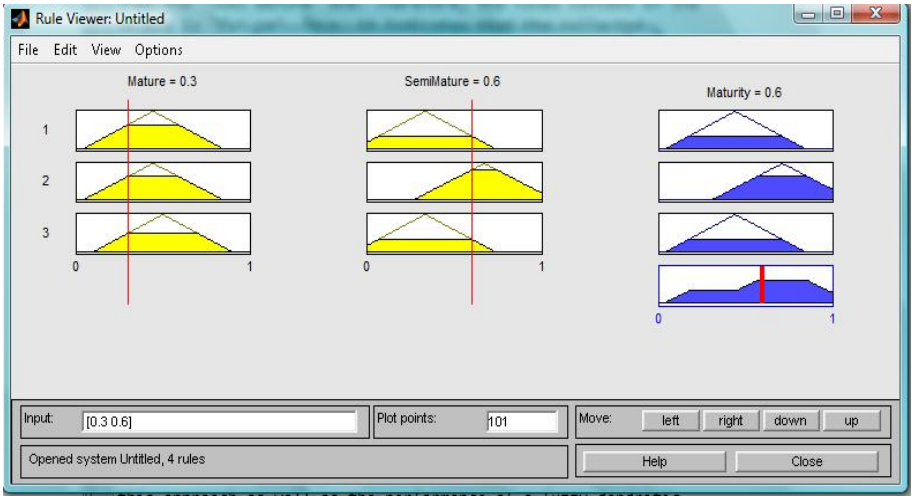

Figure 3.4: An illustrative schema of the context assessment

Figure 3.4 shows an example of the application phase of three rules of two input values [0.3 0.6], so the generation of the centroid value results in (0.6). As we remark, the middle of the output range (0.5) is lower than the centroid value ($0.5 < 0.6$). As shown in the figure, the surface of the "Maturity" output is skewed to the right. Hence, the final context of the object is "Mature".

3.4.5 The classification phase

The classification phase is the same procedure explained in the previous Chapter (see Section 2.9). 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 3.12 shows this process. 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.

$$at = \frac{an}{tn} \tag{3.12}$$

If the MCAV is greater than the anomaly threshold then the antigen is classified as anomalous else it is classified as normal.

3.5 The FDCM: An example

In this Section, we give an example in which we explain how our new approach FDCM could be used. We keep the same example of the management of bank credits. Assume that the initial data set is given by Table 3.1:

Table 3.1: 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

3.5.1 The attribute selection and categorization phase

As mentioned previously, the attribute selection and categorization phase is based on the experts knowledge.

Assume that experts select the attributes Age, Income and Duration of the loan for the FDCM functioning. We also assume that they select the attribute Income to derive the PAMP and safe signal and the rest of the attributes (Age and Duration of the loan) to calculate the danger signal values. Each data item is mapped as an antigen. Hence, Table 3.2 represents the reduced data set.

Table 3.2: Reduced data set after attribute selection

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

3.5.2 The signal values derivation phase

Process for calculating PAMP and safe signals

The attribute Income is chosen to derive PAMP and safe signals and its median value is set to 950. For each attribute value, we determine if it is a PAMP or safe signal.

For instance, for Ag1, the first attribute value (1300) is higher than (950), then the resultant signals are a PAMP of value (1300-950=350) and a SS 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, not including class 2 as with the PAMP and safe signals. 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 3.3:

Table 3.3: 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 danger signal, DS. This value is the mean value of the absolute distances calculated in the block above, with the derivation shown in Equation 3.1:

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

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 3.4. Note how if the value of PAMP is greater than zero, the value for the safe signal is set to zero.

Table 3.4: 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 (detailed in the previous Chapter).

3.5.3 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

As mentioned previously, each input is represented by three triangular membership functions. However, the output is represented by two triangular membership functions. The range of the three linguistic variables is set to [0 25] (we assume that after the generation of all the smDC and mDC, the min and max values are set to 0 and 25). The extents of the different membership functions are parameters given by the user. The inputs and the output of the system are the following:

- 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

The characteristics of the fuzzy model are listed below:

- The fuzzy model named "FDCM example" is composed of two inputs, one output and nine rules. It is based on the min-max aggregation method and the centroid deffuzification method.

How to calculate the values of the membership functions of the inputs

Let us take the example of the first input. The different equations of its three membership functions "Low", "Medium" and "High" are respectively the following:

$$f(x,0,5,10) = \begin{cases} 0 & x \leq 0 \\ \frac{x-a}{b-a} & 0 \leq x \leq 5 \\ \frac{c-x}{c-b} & 5 \leq x \leq 10 \\ 0 & 10 \leq x \end{cases}$$

$$f(x,7.5,12.5,17.5) = \begin{cases} 0 & x \leq 7.5 \\ \frac{x-a}{b-a} & 7.5 \leq x \leq 12.5 \\ \frac{c-x}{c-b} & 12.5 \leq x \leq 17.5 \\ 0 & 17.5 \leq x \end{cases}$$

$$f(x,15,20,25) = \begin{cases} 0 & x \leq 15 \\ \frac{x-a}{b-a} & 15 \leq x \leq 20 \\ \frac{c-x}{c-b} & 20 \leq x \leq 25 \\ 0 & 25 \leq x \end{cases}$$

The three membership functions are represented graphicly by Figure 3.5.

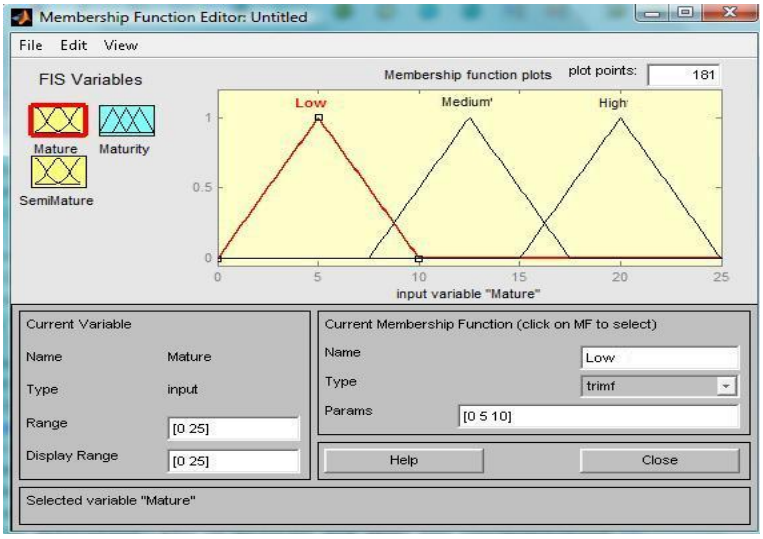

Figure 3.5: An illustrative schema of three membership functions on the “Mature” input

Since x = 8, then by fitting this value to the three equations, we get the following vector [0.4 0.1 0].

$0.4 = \frac{10-8}{10-5}$ since $5 \leq x = 8 \leq 10$

$0.1 = \frac{8-7.5}{12.5-7.5}$ since $7.5 \leq x = 8 \leq 12.5$

0 since $x = 8 \leq 15$

The same process is applied to calculate the value of y. Thus we obtain the following vector: [0.2 0.3 0].

Rules' application

Since the value of the membership function "High" is set to zero for both vectors generates of the two inputs, 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) (1)
2. If (Mature is Low) and (SemiMature is Medium) then (Maturity is Semi-Mature) (1)
4. If (Mature is Medium) and (SemiMature is Low) then (Maturity is Mature) (1)
5. If (Mature is Medium) and (SemiMature is Medium) then (Maturity is Semi-Mature) (0.15)

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 3.6 demonstrates the application phase of the rules and the generation of the centroid value which is 11.6.

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). As shown in Figure 3.6, the surface of the "Maturity" output is skewed to the left (negative asymmetry) which means that the area of the "Semi-Mature" membership function is greater than the "Mature" one. Hence, the final context of the object is "Semi-Mature".

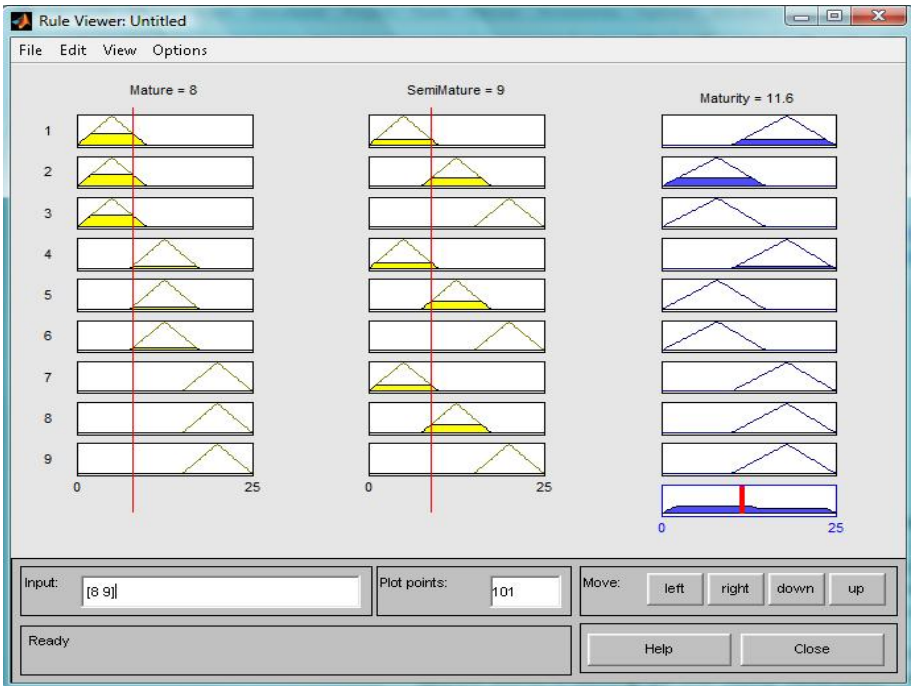

Figure 3.6: An illustrative schema of the centroid value generation

3.5.4 The classification phase

Once the context is fixed for all the objects, the classification phase (seen in the previous Chapter) 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 (normal) since (0.3=3/10<0.5). The same reasoning is applied to the rest of the instances.

3.6 Conclusion

In this Chapter, we have developed the fuzzy dendritic cell method as a new technique associating the dendritic cell algorithm with the fuzzy set theory. We have detailed the characteristics of this new approach namely its definition, objectives and representation.

In the next Chapter, we will present the implementation for checking the performance of our FDCM comparing to the standard DCA. Then, we will show different results obtained from simulations and that have been performed on real database.

Chapter 4

Implementation and simulation

4.1 Introduction

Implementing and testing our fuzzy dendritic cell method (FDCM) is important since it allows us to have an idea concerning the effectiveness of our method as well as its performance compared to the standard dendritic cell method of (Greensmith & Aickelin, 2005).

Hence, we have implemented both methods with Matlab V7.1: the FDCM as well as the approach of (Greensmith & Aickelin, 2005). Then, we have performed simulations on several two-class real databases obtained from the U.C.I. repository (Asuncion & Newman, 2007). Different results carried out from these simulations will be presented and analyzed in order to assess the effectiveness of our method.

This Chapter is composed of two parts. The first one deals with the implementation of our fuzzy dendritic cell method where the major variables and programs are detailed. The principal algorithms are also exposed. The second one is consecrated to the simulation phase where the results obtained from different experimental tests are exposed and analyzed in order to evaluate our method. Note that the objective of these simulations is to prove that our method improves the classification accuracy in the case of successive class transitions compared to the standard DCA.

4.2 Implementation

4.2.1 The framework

As mentioned above, in order to ensure the implementation of our FDCM, we have developed our program with Matlab V7.1. Obviously, we have implemented the standard DCA that we have detailed in Chapter 2. Then, we replace its crisp context assessment phase by the new fuzzy one detailed in Chapter 3. The inputs of our program are mainly:

1. The signals feature vector, where signals are pre-categorized and pre-processed as "PAMP", "danger" and "safe" (the selected attributes).
2. The antigens feature vector.
3. The weights for the signal processing.
4. The migration threshold to control the life-span of the DC that it stops sampling antigens.
5. The fuzzy system comprising the different membership functions, the rule base and the fuzzy inference process.
6. The anomaly threshold which is applied to the MCAVs: values exceeding the threshold results in the classification of an antigen as "anomalous" and vice versa.

The outputs of our program are:

1. A list of anomalous objects (antigens) and another for the normal ones.
2. The mature context antigen value (MCAV) which reflects the probability of an antigen in being anomalous.

4.2.2 Main variables

In this Section, we present the major variables that we have used in our programs to implement the FDCM:

- *data*: includes the attribute values of all the objects of the learning set.

- *all-values*: is a cell array representing the different values calculated to form the PAMP, DS and SS of each attribute.
- *number-attributs*: is the number of attributes selected by the user in order to pre-categorize them as PAMP, DS and SS.
- *parameters*: are user defined values to fix the feet and the peak of each triangular membership function.
- *Mature-input-range*: is the range of the first linguistic variable "Mature" (first input of the fuzzy system).
- *SemiMature-input-range*: is the range of the second linguistic variable "Semi-Mature" (second input of the fuzzy system).
- *Maturity-input-range*: is the range of the linguistic variable "Maturity" (the output of the fuzzy system).
- *Low-membership-function*: represents the "Low" membership function of both inputs Mature and SemiMature.
- *Medium-membership-function*: represents the "Medium" membership function of both inputs Mature and SemiMature.
- *High-membership-function*: represents the "High" membership function of both inputs Mature and SemiMature.
- *Mature-membership-function*: represents the "Mature" membership function of the output Maturity.
- *Semi-mature-membership-function*: represents the "Semi-mature" membership function of the output Maturity.
- *RuleBase*: represents the rule base of the fuzzy system.

4.2.3 Main programs

In this subsection, we will present the major programs that we have developed to construct our software. These programs can be regrouped according to their use into distinct parts:

The detection phase for input data and DC population

- **load-data**: opens an existing database file and loads it into the variable data.
- **calculate interim output signals**: at each iteration, each DC calculates three temporary output signal values from the received input signals which are the CSM, the semi mature output and the mature output.
- **update cumulative output signals**: the DC collects values of all input signals present in the signal storage area.
- **record outputs**: at each iteration, both the semi mature output value and the mature output value are summed and recorded.

The fuzzy process construction phase

- **get-max**: picks up the maximum value of a vector.
- **get-min**: picks up the minimum value of a vector.
- **create-rule-base**: generates a rule base using the different parameters of the fuzzy system.

The fuzzy context assessment phase

- **eval**: performs the fuzzy inference calculations.

4.2.4 The fuzzy dendritic cell method

In this section, we will present the major algorithms relative to our method to ensure the classification of different instances, namely the detection phase for input data and DC population, the fuzzy process construction phase, the fuzzy context assessment phase for output list and the generation of MCAV coefficient algorithms.

Algorithm: Detection-Inputs
input: database-file-path, signals-feature-vector, antigens-feature-vector, weights, migration threshold
output: Semi-vector, Mat-vector

```
1: begin
2: (*define the data set*)
3: [data, all-values, number-attributes] ← load-data (database-file-path);
4: (*process for recording the outputs Semi-vector and Mat-vector*)
5: initialise DCs;
6: while CSM output signal < migration Threshold do
7:     get antigen;
8:     store antigen;
9:     get signals;
10:    calculate interim output signals;
11:    update cumulative output signals;
12:    record outputs;
13: end while
14: Cell location update to lymph node;
15: end.
```

Algorithm: Fuzzy-System-Construction
input: Semi-vector, Mat-vector, parameters, rule-base, term-sets, linguistic-variables
output: fuzzy-system

```
1: begin
2: (*defining ranges*)
3: MaxS ← get-max (Semi-vector);
4: MinS ← get-min (Semi-vector);
5: MaxM ← get-max (Mat-vector);
6: MinM ← get-min (Mat-vector);
7: MinMaturity ← get-min [MinS, MinM];
8: MaxMaturity ← get-max [MaxS, MaxM];
9: (*defining the characteristics of the first input Mature*)
10: Mature-input-range ← [MinM, MaxM];
11: Low-membership-function ← [parameters];
12: Medium-membership-function ← [parameters];
13: Hight-membership-function ← [parameters];
14: (*defining the characteristics of the input SemiMature*)
15: SemiMature-input-range ← [MinS, MaxS];
16: Low-membership-function ← [parameters];
17: Medium-membership-function ← [parameters];
18: Hight-membership-function ← [parameters];
19: (*defining the characteristics of the ouput Maturity*)
```

```
Maturity-input-range ← [MinMaturity, MaxMaturity];
Mature-membership-function ← [parameters];
Semi-mature-membership-function ← [parameters];
(*defining the rule base*)
RuleBase ← create-rule-base (term-sets, linguistic-variables);
end.
```

Algorithm: Fuzzy-Context-Assessment
input: fuzzy-system, Semi-vector, Mat-vector
output: Antigen and their context (0/1)

```
begin
i = 0;
for all values in Semi-vector and Mat-vector do
    mature ← Semi-vector(i);
    semi-mature ← Mat-vector(i);
    centroid-value ← eval ([mature semi-mature], fuzzy-system);
    (*comparing the centroid-value of each object with the middle of the output range
    of the fuzzy system*)
    if centroid-value < middle-of-Maturity-output then
        cell context is assigned as 0;
    else
        cell context is assigned as 1;
    end if
    i = i + 1;
end for
end.
```

Algorithm: The generation of MCAV coefficients for each antigen type sampled by the FDCM
input: total list of antigen plus context values per experiment
output: MCAV coefficient per antigen type

```
begin
for all antigen in total list do
    increment antigen count for this antigen type;
    if antigen context = 1 then
        increment antigen type mature count;
```

```
    end if
end for
(*defining the MCAV for each data instance*)
for all antigen types do
    MCAV of antigen type = mature count / antigen count;
end for
end.
```

4.3 Simulations and results

4.3.1 Experimental setup

The implementation of our fuzzy dendritic cell method will be useful in the simulation phase. It is crucial to mention that in addition to the classification of the instances using FDCM, our objective is to look at the impact of smoothing the abrupt (crisp) separation between normality (semi-mature) and abnormality (mature).

Hence, we have performed several tests and simulations on real databases obtained from the U.C.I repository of Machine Learning databases (Asuncion & Newman, 2007).

Different results carried out from these simulations will be presented and analyzed in order to evaluate our proposed method.

4.3.2 Evaluation criterion

In order to evaluate our method, and as done with the standard DCA, we have based our evaluation on accuracy. In fact, the accuracy of a classification method is determined by measuring the number of instances it, correctly, classifies among the total number of testing instances presented to the classifier. Hence, we will use the Percent of Correct Classification (PCC).

The PCC represents the percent of the correct classification of the testing instances which are classified according to the induced fuzzy dendritic cell method. It is given by Equation 4.1:

$$PCC = \frac{\text{number of well classified instances}}{\text{total number of classified instances}} * 100 \tag{4.1}$$

Hence, to compute the PCC, we have to compare for each object, its real class (in the initial ordinary database) to the class given by the FDCM. The number of well classied instances corresponds then to the number of objects for which, the class obtained from the algorithm concordes with the real class.

Obviously, a PCC equal to 100% qualifies an excellent classifier, whereas a PCC equal to 0% corresponds to a null classifier.

An equivalent criterion, also used in the literature, measure the proposition of incorrectly classified instances. This is knows as the *error rate (r=1-PCC)*.

4.3.3 Validation procedure

In our simulations, in order to obtain an unbiased estimation of the PCC, we have used a method called: *cross validation*.

This method divides a given data set into n parts, (n - 1) parts will be used as the training set and the remaining part will be used to test the induced FDCM. The procedure is repeated n times, each time using another (n - 1) parts as the training set and another part as the testing set.

The parameter n represents the number of folds of the cross validation procedure. In our simulations we have used the 10-folds-cross-validation. Obviously, in each fold, we compute the corresponding PCC and the final PCC is given by the mean of the computed PCCs.

4.3.4 Simulations on the real databases

Description of databases

For the evaluation of our proposed FDCM, we have used real two-class databases obtained from U.C.I repository of Machine Learning databases (Asuncion & Newman, 2007).

In Table 4.1, a brief description of these databases is given. A detailed description is given in Appendix A at the end of this report.

Table 4.1: Description of databases

Database	Ref	♯ instances	♯ attributes
Mammographic Mass	MM	961	6
Pima Indians Diabetes	PID	768	8
Blood Transfusion Service Center	BTSC	748	5
Wisconsin Breast Cancer	WBC	700	9
Haberman's Survival	HS	306	4
SPECTF Heart	SPECTF	267	44

Experimental results

Previous examinations with DCA, in (Aickelin & Cayzer, 2005), show that the DCA is sensitive to the data order since it does not perform well its classification task in the case of a disordered contexts. These misclassifications occur exclusively at the transition boundaries. As a result, the DCA makes more mistakes when the context changes multiple times in a quick succession. One possible explanation of such limitation is the crisp separation between the two contexts (semi-mature and mature) which affects badly the correctness of the classification results.

Applying such algorithm to a database where its instances are randomized between class one and class two, decreases dramatically the classification accuracy.

Let us remind that the aim of our method is to improve the classification accuracy even in the case of contexts' change. This will prove that our FDCM does not depend of the class transitions. In fact, smoothing the absurd separation between normality and abnormality using fuzzy set theory allows to handle the drawbacks of the DCA.

Thus, our experimentations are based on randomizing each time the data set between the two classes.

The order of the data items varies according to experiments. Experiment 1 uses all class 1 items followed by all class 2 items (ordered database). The rest of the experiments uses data from class 1 and class 2 that is randomized once, then

20 times, then 60, 70, 90, 120, 140, 160, 200, and finally 300 times successively. Each experiment is performed 10 times.

So if we increase each time the number of randomization (R) from one experiment to another, this will lead to a database randomized more between the two classes, so to successive transitions (class of data instances changes multiple times). Such randomization allows us to prove that our FDCM, unlike the DCA, does not depend on these class transitions and that it is able the improve the classification accuracy in such cases.

In all the mentioned databases, each antigen is unique, appearing only once. However, in (Aickelin & Cayzer, 2005), it was proved that this is insufficient to the functioning of the algorithm. This is overcome through the use of an antigen multiplier which reflects the number of antigen copies produced.

For our simulations, each antigen is copied ten times using an antigen multiplier giving 9610, 7680, 7480, 7000, 3060 and 2670 antigen presentations for all the mentioned data sets.

In order to determine the final class label of each antigen, an anomaly threshold must be applied to the MCAVs. As shown in Chapter 3, the different thresholds are calculated using Equation 3.12.

The threshold for classification is set to 0.3, 0.9, 0.6, 0.65, 0.2647 and 0.6 to all the mentioned data sets. Items exceeding the threshold are classed as class 2, with lower valued antigen labeled as class 1.

These classifications are compared with the labels presented in the original data sets so the classification accuracy can be measured.

Results: Table 4.2 summarizes the different results relative to all the mentioned databases. This Table presents a comparison between the DCA and our FDCM in terms of Percent of Correct Classification (PCC) after comparing them with all the original data sets.

Table 4.2: Experimental measures PCC (%) (E1=Experiment 1, R=Randomized)

Database	Method	E1	R=1	R=20	R=60	R=70	R=90	R=120	R=140	R=160	R=200	R=300
MM	DCA	98,43	62,85	62,85	62,85	62,85	62,85	62,85	62,85	62,85	62,85	62,85
	FDCM	62,85	97,19	95,63	96,67	94,48	97,5	96,15	95,42	97,09	96,77	96,36
PID	DCA	96,35	98,82	96,48	94,14	98,56	95,31	96,35	96,61	96,35	96,22	96,09
	FDCM	98,05	96,87	97	96,48	97,79	95,83	96,74	98,31	96,22	95,57	97
BTSC	DCA	91,71	93,31	91,44	93,98	90,1	91,57	93,18	92,24	91,71	91,57	91,44
	FDCM	48,66	48,52	98,53	96,8	95,45	96,25	96,25	98,26	98,4	94,92	97,33
WBC	DCA	99,42	91	89,71	91,14	90,42	90,71	89,71	90,71	91	90,85	90,57
	FDCM	98	97,57	98,57	99,57	96,57	97,57	99,28	97,57	97,14	97,14	99
HS	DCA	83	17,32	17,32	17,64	16,99	17,32	16,99	16,99	17,64	16,66	16,66
	FDCM	29	82,35	92,15	90,85	82,02	93,8	92,48	92,15	89,54	92,15	91,83
SPECTF	DCA	93,63	91,76	91,01	91,76	91,38	92,13	91,38	90,63	90,63	89,13	88,01
	FDCM	79,4	79,4	92,5	94,38	95,13	94,75	93,63	94,75	94,75	94,75	94,38

From Table 4.2, we can conclude that our fuzzy dendritic cell method has given good results. In fact, by randomizing the values of R, the PCC of our FDCM is better than the one given by DCA. Hence, such randomization of the values of R generally affects badly the classification accuracy of the DCA, which is not the case with our FDCM.

For instance, by applying the DCA to the Haberman's Survival database and with the variation of the different values of R, the PCC varies from 16,66% to 17,64%. Whereas, with our FDCM, the PCC varies from 82,02% to 93,8%. This is explained by the appropriate use of the fuzzy set theory in the case of a randomized context.

Note that in the case of an ordered database (E1), the PCC of the DCA is generally better than the PCC of our FDCM (except for the PID database) since fuzzy set theory is more appropriate to handle cases of randomization than ordered data sets.

A comparison between our FDCM and the DCA in terms of PCC for all the mentioned databases is illustrated in Figure 4.1.

From Figure 4.1 and Table 4.2, we can conclude that in most cases when the context changes multiple times in a quick succession, it is more appropriate to apply FDCM than the standard DCA since the former produces more accurate results.

Furthermore, regarding to the computational complexity, let us remind that the computational complexity of the DCA is $O(n^2)$. In order to calculate it, the algorithm is divided into two main parts: the detection phase, having a complexity of $O(n)$, where the context of each object is assigned and the analysis phase, having a complexity of $O(n^2)$, where the MCAVs are generated.

The difference between the DCA and our FDCM resides in the detection phase. Although our FDCM uses a fuzzy context assessment, the detection phase keeps the same complexity of $O(n)$. This is because our FDCM only uses elementary instructions of $O(1)$. Hence, this allows the FDCM to have a computational complexity of $O(n^2)$.

As we remark, our FDCM achieves the same computational complexity as the

DCA even by adding the fuzzy technique and getting the min-max boundaries generated by DCA which is another important characteristic of our method.

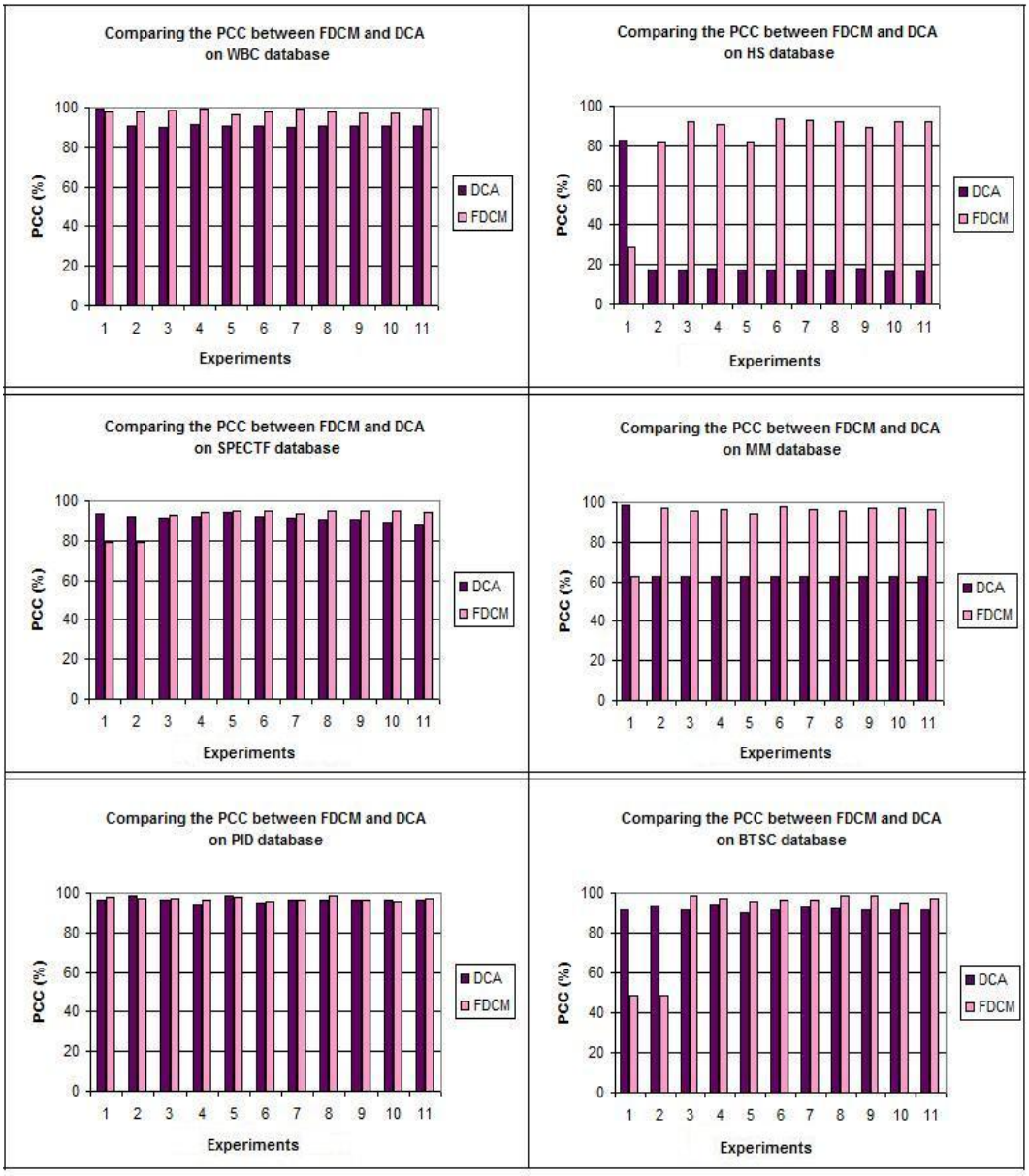

Figure 4.1: Comparing FDCM and DCA PCCs

To summarize, with the variation of the parameter R, our fuzzy dendritic cell method gives better results than the standard DCA in terms of classification accuracy without increasing complexity.

4.4 Conclusion

In this Chapter, we have outlined our proposed method: the FDCM. We have thus presented the main variables and the major implemented programs. We have also detailed the basic algorithms relative to our method.

Next, we have presented the experimental results obtained from several simulations which are based on different types of randomization. These experiments show that our approach gives better results than the standard version of this method. Note that the major results of this work are developed in (Chelly & Elouedi, 2010).

Conclusion

The classification procedure handled by the standard dendritic cell method in danger theory is based on a crisp comparison between the semi-mature output value and the mature output value generated by the algorithm. However, such crisp separation affects badly the classification task. Another limitation of the standard DCA is the fact of using imprecise terms such as "semi-mature" and "mature". In order to solve these issues, we propose to alleviate this crisp comparison as well as to handle this imprecision using the basics of fuzzy set theory.

In this master thesis, we have developed a fuzzy dendritic cell method (FDCM) which is based on both the dendritic cell algorithm and the fuzzy set theory. This hybridization allows the standard DCA to be more effective in terms of classification accuracy in the case of disordered contexts.

At first, we have used the fuzzy set theory to smooth the abrupt separation between normality (semi-mature) and abnormality (mature), since there is no clear boundary between the two contexts.

The second step was the description of the context of each object using linguistic variables. Fuzzy subsets and the corresponding membership functions describe the context of each object. In other words, ensuring the induction of the fuzzy set theory.

Finally, we have built a knowledge base, comprising rules to support the fuzzy inference.

In our work, we have performed simulations on several real databases obtained from the U.C.I. repository in order to evaluate the performance of our fuzzy dendritic cell method comparing it with the standard DCA.

Results of experimentations show that our fuzzy dendritic cell method performs well in most cases. There is a significant improvement of classification accuracy of FDCM, unlike the DCA, in the case of a high rate of class randomization.

Another important characteristic of our fuzzy dendritic cell method is its ability to increase the classification accuracy without increasing complexity even by adding the fuzzy technique and getting the min-max boundaries generated by DCA.

Finally, it is important to mention that regarding the encouraging results obtained in this work, we could propose further works that may be done to improve our method.

As future works, we intend to further explore this new instantiation of the DCA. This investigation will involve an automatical generation of the weights needed for the functioning of the FDCM since they are given by experts.

Another line of research could be the application of rough sets for the selection of attributes. Moreover, since the DCA is applied only to problems with two classes, we aim to extend the application area of this algorithm to problems with multi classes. It would also be interesting to focus on the application of our FDCM to different domains, especially to intrusion detection problem.

Appendix A

Data bases used for simulations

A.1 Introduction

For making simulation, we have used these six two-class static databases: the Mammographic Mass database, the Pima Indians Diabetes database, the Blood Transfusion Service Center database, the Wisconsin Breast Cancer database, the Haberman's Survival database and the SPECTF Heart database (Asuncion & Newman, 2007) to evaluate our method. These databases are presented in this appendix.

A.2 Mammographic Mass database

1. Title: Mammographic Mass Database

2. Sources:

   (a) Original owners of database: Prof. Dr. Rudiger Schulz-Wendtland Institute of Radiology, Gynaecological Radiology, University Erlangen-Nuremberg Universittsstrae 21-23 91054 Erlangen, Germany.

   (b) Donor of database: Matthias Elter Fraunhofer Institute for Integrated Circuits (IIS) Image Processing and Medical Engineering Department (BMT) Am Wolfsmantel 33 91058 Erlangen, Germany. matthias.elter@iis.fraunhofer.de (49) 9131-7767327

   (c) Date received: October 2007.

3. Relevant Information:

   This data set is used to predict the severity (benign or malignant) of a mammographic mass lesion. It contains a BI-RADS assessment, the patient's age and three BI-RADS attributes together with the ground truth (the severity field) for masses that have been identified on full field digital mammograms collected at the Institute of Radiology of the University Erlangen-Nuremberg between 2003 and 2006. Each instance has an associated BI-RADS assessment ranging from 1 (definitely benign) to 5 (highly suggestive of malignancy) assigned in a double-review process by physicians.

4. Number of Instances: 961

5. Number of Attributes:
   5 attributes + goal field = 6

6. Attribute selection and categorization:

   * Attribute used to generate the PAMP and SS: Shape
   * Attributes used to generate the DS: BI-RADS assessment, Age and Margin

7. Weights used for signal processing:

	PAMP	SS	DS
CSM	0	3	0
Semi-output	0	0	1
Mature-output	-0.52	0	0.1

8. The anomaly threshold: 0.3

9. Class Distribution:
   Benign: 516 (53.7%)
   Malignant: 445 (46.3%)

A.3 Pima Indians Diabetes database

1. Title: Pima Indians Diabetes Database

2. Sources:

   (a) Original owners: National Institute of Diabetes and Digestive and Kidney Diseases

   (b) Donor of database: Vincent Sigillito (vgs@aplcen.apl.jhu.edu) Research Center, RMI Group Leader Applied Physics Laboratory The Johns Hopkins University Johns Hopkins Road Laurel, MD 20707 (301) 953-6231

   (c) Date received: 9 May 1990

3. Relevant Information:

   All patients in this database are females at least 21 years old of Pima Indian heritage. ADAP is an adaptive learning routine that generates and executes digital analogs of perceptron-like devices.

4. Number of Instances: 768

5. Number of Attributes:
   5 attributes + goal field = 6

6. Attribute selection and categorization:

   * Attribute used to generate the PAMP and SS: Age

   * Attributes used to generate the DS: Plasma glucose concentration a 2 hours in an oral glucose tolerance test, Diastolic blood pressure (mm Hg), Triceps skin fold thickness (mm) and 2-Hour serum insulin (mu U/ml).

7. Weights used for signal processing:

	PAMP	SS	DS
CSM	2	2	1
Semi-output	0	3	0
Mature-output	1	0	-0.13

8. The anomaly threshold: 0.9

9. Class Distribution:
   Tested negative for diabetes: 500 (65,1%)
   Tested positive for diabetes: 268 (34,9%)

A.4 Blood Transfusion Service Center database

1. Title: Blood Transfusion Service Center Database

2. Sources:

   (a) Original Owner and Donor Prof. I-Cheng Yeh Department of Information Management Chung-Hua University, Hsin Chu, Taiwan 30067, R.O.C. e-mail:icyeh 'at' chu.edu.tw TEL:886-3-5186511

   (c) Date Donated: October 3, 2008

3. Relevant Information:

   This study adopted the donor database of Blood Transfusion Service Center in Hsin-Chu City in Taiwan. The center passes their blood transfusion service bus to one university in Hsin-Chu City to gather blood donated about every three months. 748 donors are selected at random from the donor database. These 748 donor data, each one included R (Recency - months since last donation), F (Frequency - total number of donation), M (Monetary - total blood donated in c.c.), T (Time - months since first donation), and a binary variable representing whether he/she donated blood in March 2007 (1 stand for donating blood; 0 stands for not donating blood).

4. Number of Instances: 748

5. Number of Attributes:
   4 attributes + goal field = 5

6. Attribute selection and categorization:

   * Attribute used to generate the PAMP and SS: F (Frequency - total number of donation)
   * Attributes used to generate the DS: R (Recency - months since last donation), M (Monetary - total blood donated in c.c.) and T (Time - months since first donation)

7. Weights used for signal processing:

	PAMP	SS	DS
CSM	2	2	1
Semi-output	0	3	0
Mature-output	1	0	0.02

8. The anomaly threshold: 0.6

9. Class Distribution:
   He/She donated blood: 178 (23,8%)
   He/She did not donate blood: 570 (76,2%)

A.5 Wisconsin Breast Cancer database

1. Title: Wisconsin Breast Cancer Database

2. Sources:

   (a) Dr. William H. Wolberg (physician) University of Wisconsin Hospitals Madison, Wisconsin USA.

   (b) Donor: Olvi Mangasarian (mangasarian@cs.wisc.edu) Received by David W. Aha (aha@cs.jhu.edu).

   (c) Date: 15 July 1992.

3. Relevant Information:

The database reflects a chronological grouping of data. This grouping information is the following:
Group 1: 367 instances (January 1989)
Group 2: 70 instances (October 1989)
Group 3: 31 instances (February 1990)
Group 4: 17 instances (April 1990)
Group 5: 48 instances (August 1990)
Group 6: 49 instances (Updated January 1991)
Group 7: 31 instances (June 1991)
Group 8: 86 instances (November 1991)
So the total = 699 points (as of the donated datbase on 15 July 1992).

4. Number of Instances: 699

5. Number of Attributes:
   9 attributes + goal field = 10

6. Attribute selection and categorization:

   * Attribute used to generate the PAMP and SS: Clump thickness
   * Attributes used to generate the DS: Epithelial cell size, Cell shape, Bare nuclei and Normal nucleoli

7. Weights used for signal processing:

	PAMP	SS	DS
CSM	2	2	1
Semi-output	0	3	0
Mature-output	2	0	0.75

8. The anomaly threshold: 0.65

9. Class Distribution:
   Benign: 458 (65.5%).
   Malignant: 241 (34.5%).

A.6 Haberman's Survival database

1. Title: Haberman's Survival Database
2. Sources:
   (a) Donor: Tjen-Sien Lim (limt@stat.wisc.edu)
   (b) Date: March 4, 1999.
3. Relevant Information:

   The data set contains cases from a study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of patients who had undergone surgery for breast cancer.
4. Number of Instances: 306
5. Number of Attributes:
   3 attributes + goal field = 4
6. Attribute selection and categorization:
   * Attribute used to generate the PAMP and SS: Number of positive axillary nodes detected
   * Attributes used to generate the DS: Age of patient at time of operation and Patient's year of operation
7. Weights used for signal processing:

	PAMP	SS	DS
CSM	2	2	1
Semi-output	0	3	0
Mature-output	-0.56	0	0.1

8. The anomaly threshold: 0.2647
9. Class Distribution:
   The patient survived 5 years or longer: 225 (73,5%)
   The patient died within 5 year: 81 (26,5%)

A.7 SPECTF Heart database

1. Title: SPECTF heart Database

2. Sources:

   (a) Original owners: Krzysztof J. Cios, Lukasz A. Kurgan University of Colorado at Denver, Denver, CO 80217, U.S.A. Krys.Cios@cudenver.edu Lucy S. Goodenday Medical College of Ohio, OH, U.S.A.

   (b) Donors: Lukasz A.Kurgan, Krzysztof J. Cios

   (c) Date: 10/01/01

3. Relevant Information:

   The data set describes diagnosing of cardiac Single Proton Emission Computed Tomography (SPECT) images. The database of 267 SPECT image sets (patients) was processed to extract features that summarize the original SPECT images. As a result, 44 continuous feature pattern was created for each patient.

4. Number of Instances: 267

5. Number of Attributes:
   44 attributes + goal field = 45

6. Attribute selection and categorization:

   * Attribute used to generate the PAMP and SS: F14S (count in ROI 14 in stress)

   * Attributes used to generate the DS: F8S (count in ROI 8 in stress), F13R (count in ROI 13 in rest), F13S (count in ROI 13 in stress), F15R (count in ROI 15 in rest), F15S (count in ROI 15 in stress), F18S (count in ROI 18 in stress), F20R (count in ROI 20 in rest), F20S (count in ROI 20 in stress), F21R (count in ROI 21 in rest), F21S (count in ROI 21 in stress), F22R (count in ROI 22 in rest) and F22S: continuous (count in ROI 22 in stress).

7. Weights used for signal processing:

	PAMP	SS	DS
CSM	2	2	1
Semi-output	0	3	0
Mature-output	-1	0	1.9

8. The anomaly threshold: 0.6

9. Class Distribution:
   Benign: 55 (20,6%)
   Malignant: 112 (79,4%)

A.8 Conclusion

In this appendix, we have shown the description of the data sets which we have been used in the simulation phase.

Bibliography

Aickelin, U., & Cayzer, S. (2005). Introducing dendritic cells as a novel immune-inspired algorithm for anomaly detection. *Proceedings of the 4th Internatinal Conference on Artificial Immune Systems, Springer, ICARIS' 2005*, *3627*, 153-167.

Aickelin, U., & McLeod, J. (2003). Danger theory: The link between ais and ids. *Proceedings of 2nd International Conference on Artificial Immune Systems, Springer, ICARIS' 2003*, *2787*, 147-155.

Alder, M., & Pancer, Z. (2005). Diversity and function of adaptive immune receptors in a jawless vertebrate. *Science*, *5756*, 1970-1973.

Andrews, P., & Timmis, J. (2005). Inspiration for the next generation of artificial immune systems. *Proceedings of the 4th Internatinal Conference on Artificial Immune Systems, Springer, ICARIS' 2005*, *3627*, 126-138.

Asuncion, A., & Newman, D. (2007). Uci machine learning repository. *http://www.ics.uci.edu/ mlearn/*.

Breunig, H., & Albert, J. (2002). On the use of negative selection in an artificial immune system (mpp). *Proceedings of the International Conference Genetic and Evolutionary Computation*, 957-964.

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*, 904-918.

Castro, L., & Timmis, J. (2003). *Artificial immune systems: A new computational intelligence approach*. Springer-Verlag.

Castro, L., & Zuben, F. (2000). The clonal selection algorithm with engineering applications. *Proceedings of the Genetic and Evolutionary Computation Conference*, 36-37.

Castro, L., & Zuben, F. (2002). Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation*, *6*, 239-251.

Cayzer, S., & Aickelin, U. (2002a). The danger theory and its application to artificial systems. *Proceedings of the 1st International Conference on Artificial Immune Systems, Springer, ICARIS' 2002*, 141-148.

Cayzer, S., & Aickelin, U. (2002b). A recommender system based on the immune network. *Proceedings of the Congress on Evolutionary Computation*, 807-812.

Chelly, Z., & Elouedi, Z. (2010). Fdcm: A fuzzy dendritic cell method. *Proceedings of the 9th International Conference on Artificial Immune Systems, Springer, ICARIS' 2010 (To appear)*.

Chun, J., & Hong, S. (1997). Shape optimization of electromagnetic devices using immune algorithm. *IEEE Transactions on Magnetics*, *33*, 1876-1879.

Dasgupta, D., & Majumdar, N. (2002). Anomaly detection in multidimensional data using negative selection algorithm. *Proceedings of the Congress on Evolutionary Computation*, 1039-1044.

Derakhshanfar, M., & Minaei-Bidgoli, B. (2009). Optimizing spatial trend detection by artificial immune systems. *World Academy of Science, Engineering and Technology*, *4*, 246-252.

Deschrijver, G., & Kerre, E. (2003). On the composition of intuitionistic fuzzy relations. *Fuzzy Sets and Systems*, *136*, 333-361.

Dubois, D., & Prade, H. (1997). The three semantics of fuzzy sets. *Fuzzy Sets and Systems*, 141-150.

Fortuna, L., & Graziani, S. (2007). Membership function shape and the dynamic behaviour of fuzzy systems. *International Journal of Adaptive Control and Signal Processing*, *4*, 369-377.

Fu, H., & Li, G. (2008). Design of an immune-inspired danger theory model based on fuzzy set. *International Symposium on Computational Intelligence and Design*, *1*, 133-136.

Fu, H., & Zhang, C. (2009). Design of a danger signal detecting model based on fuzzy-set. *Proceedings of the 5th International Conference on Wireless communications, networking and mobile computing*, 4566-4568.

Goguen, J. (1967). L-fuzzy sets. *Journal of Mathematical Analysis and Applications*, *18*, 145-174.

Greensmith, J. (2007). *The dendritic cell algorithm*. Unpublished doctoral dissertation, University of Nottingham.

Greensmith, J., & Aickelin, U. (2004). Detecting danger: Applying a novel immunological concept to intrusion detection systems. *Proceedings of the 6th International Conference in Adaptive Computing in Design and Manufacture*, *3239*, 316-329.

Greensmith, J., & Aickelin, U. (2005). Articulation and clarification of the dendritic cell algorithm. *Proceedings of the 5th International Conference on Artificial Immune Systems, Springer, ICARIS' 2005*, *4163*, 404-417.

Greensmith, J., & Aickelin, U. (2006). Dendritic cells for anomaly detection. *Proceedings of the IEEE Congress on Evolutionary Computation*, 664-671.

Greensmith, J., & Aickelin, U. (2007a). The application of a dendritic cell algorithm to a robotic classifier. *Proceedings of the 6th International Conference on Artificial Immune Systems, Springer, ICARIS' 2007*, *4628*, 204-215.

Greensmith, J., & Aickelin, U. (2007b). Dendritic cells for syn scan detection. *Proceedings of the Genetic and Evolutionary Computation Conference*, 49-56.

Greensmith, J., & Aickelin, U. (2007c). Further exploration of the dendritic cell algorithm. *Proceedings of the 7th International Conference on Artificial Immune Systems, Springer, ICARIS' 2007*, *5132*, 142-153.

Greensmith, J., & Aickelin, U. (2008). The dca for bot detection. *Proceedings of the IEEE World Congress on Computational Intelligence*, 1807-1816.

Greensmith, J., & Cayzer, S. (2003). An artificial immune system approach to semantic document classification. *Springer-Verlag*, *2787*, 136-146.

Hart, E., & Timmis, J. (2005). Application areas of ais: the past, present and future. *Proceedings of the 4th International Conference on Artificial Immune Systems, Springer, ICARIS' 2005*, *3627*, 483-499.

Hofmeyr, S., & Forrest, S. (1999). Immunity by design: An artificial immune system. *Proceedings of the Genetic and Evolutionary Computation Conference*, 1289-1296.

Hofmeyr, S., & Forrest, S. (2000). Architecture for an artificial immune systems. *Journal of Evolutionary Computation*, *7*, 1289-1296.

Huang, S. (2002). Application of immune-based optimization method for fault-section estimation in a distribution system. *IEEE Transactions on Power Delivery*, *17*, 779-784.

Hunt, J., & Cooke, D. (1996). Learning using an artificial immune system. *Journal of Network and Computer Application*, *19*, 189-212.

Janeway, C. (1992). The immune system evolved to discriminate infectious nonself from noninfectious self immunol. *Immunology Today*, *13*, 11-16.

Janeway, C., & Travers, P. (2001). *Immunobiology*. Garland Science.

Jaradat, M., & Langari, R. (2008). A hybrid system for fault detection and sensor fusion based on fuzzy clustering and ais. *Applied Soft Computing*, *9*, 415-422.

Kim, J., & Bentley, P. (2001). An evaluation of negative selection in an artificial immune system for network intrusion detection. *Proceedings of the Genetic and Evolutionary Computation Conference*, 1330-1337.

Kim, J., & Bentley, P. (2002). Toward an artificial immune system for network intrusion detection: An investigation of dynamic clonal selection. *IEEE press*, 1015-1020.

Kyosev, Y., & Reinbach, I. (2006). Max-product fuzzy relational equations as inference engine for prediction of textile yarn properties. *9th Fuzzy Days*, 18-20.

Lee, C. (1990). Fuzzy logic in control systems: Fuzzy logic controller - parts 1 and 2. *IEEE Transactions on Systems, Man and Cybernetics*, 404-435.

Mamdani, E., & Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 1-13.

Matzinger, P. (2001). The danger model in its historical context. *Scandinavian Journal of Immunology*, *54*, 4-9.

Mayer, G. (2005). *Microbiology and immunology*. Unpublished doctoral dissertation, University of South Carolina.

McEnery, R. (2008). Understanding innate immunity and hiv. *International AIDS Vaccine Initiative*, 127-130.

Medzhitov, R., & Janeway, C. (2002). Decoding the patterns of self and nonself by the innate immune system. *Science*, *296*, 298-300.

Meng, L., & Wang, H. (2005). A comprehensive benchmark of the artificial immune recognition system (airs). *Lecture Notes in Computer Science*, *3584*, 575-582.

Mezyk, E., & Unold, O. (2009). Improving mining fuzzy rules with artificial immune systems by uniform population. *Springer-Verlag*, *58*, 295-303.

Mizumoto, M. (1990). Fuzzy controls by product-sum gravity-method. *Fuzzy Sets and Systems*, 461-466.

Nasraoui, O., & Dasgupta, D. (2002). The fuzzy artificial immune system: Motivations, basic concepts, and application to clustering and web profiling. *Proceedings of the World Congress on Computational Intelligence*, 711-716.

Polat, K., & Kodaz, H. (2006). A new hybrid method based on fuzzy-artificial immune system and kk-nn algorithm for breast cancer diagnosis. *Computers in Biology and Medicine*, *37*, 415-423.

Ross, T. (1995). *Fuzzy logic with engineering applications* (Wiley, Ed.) (No. 628). McGraw-Hill.

Roth, M., & Mervis, C. (1983). Fuzzy set theory and class inclusion relations in semantic categories. *Journal of Verbal Learning and Verbal Behavior*, *22*, 509-525.

Secker, A., & Timmis, J. (2003a). Aisec: an artificial immune system for e-mail classification. *Proceedings of the Congress on Evolutionary Computation*, 131-139.

Secker, A., & Timmis, J. (2003b). A danger theory inspired approach to web mining. *Springer-Verlag*, *2787*, 156-167.

Shah, H., & Joshi, A. (2003). Fuzzy clustering for intrusion detection. *The 12th IEEE International Conference on Fuzzy Systems*, *2*, 1274-1278.

Shulin, L., & Wenhu, H. (2002). Negative-selection algorithm based approach for fault diagnosis of rotary machinery. *Proceedings of American Control Conference*, *5*, 3955-3960.

Silvert, W. (1979). Symmetric summation: a class of operations on fuzzy sets. *IEEE Transactions on Systems, Man, and Cybernetics*, *9*, 657-659.

Snapper, C., & Mond, J. (1996). A model for induction of t cell-independent humoral immunity in response to polysaccharide antigens. *Immunoly*, 276-294.

Sporri, R., & Caetano, C. (2005). Inflammatory mediators are insufficient for full dendritic cell activation and promote expansion of cd4+ t cell populations lacking helper function. *Nature Immunology*, *6*, 163-170.

Stibor, T. (2006). *On the appropriateness of negative selection for anomaly detection and network intrusion detection*. Unpublished doctoral dissertation, Darmstadt University of Technology.

Stibor, T., & Jimmis, J. (2006). On permutation masks in hamming negative selection. *Proceedings of the 5th International Conference on Artificial Immune Systems, Springer, ICARIS' 2006*, *4163*, 122-135.

Tay, J., & Kwoh, C. (2005). Artificial immune system for solving generalized geometric problems: a preliminary results. *Proceedings of the Genetic and Evolutionary Computation*, 329-336.

Tong, R., & Bonissone, P. (1980). Linguistic approach to decision making with fuzzy sets. *IEEE Transactions on Systems, Man and Cybernetics*, *10*, 716-723.

Visconti, A., & Tahayori, H. (2008). A type-2 fuzzy set recognition algorithm for artificial immune system. *Springer-Verlag*, 491-498.

Vos, Q., & Lees, A. (2000). B-cell activation by t-cell-independent type 2 antigens as an integral part of the humoral immune response to pathogenic microorganisms. *Immunoly*, 176-154.

Watkins, A., & Timmis, J. (2004). Artificial immune recognition system (airs): An immune-inspired supervised learning algorithm. *Genetic Programming and Evolvable Machines*, *5*, 291-317.

Wu, J., & Chung, Y. (2005). Artificial immune system for medical data classification. *Springer-Verlag*, *3516*, 810-812.

Xu, L. (2006). *Biologically inspired intelligent fault diagnosis for power distribution systems*. Unpublished doctoral dissertation, Faculty of North Carolina State University.

Younsi, R., & Wang, W. (2004). A new artificial immune system algorithm for clustering. *Springer-Verlag*, *3177*, 58-64.

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, parts 1, 2, and 3. *Information Sciences*, *8:199-249, 8:301-357, 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.

Zadeh, L. (1994). *Between mind and computer: fuzzy science and engineering* (Vol. 1). Advances in Fuzzy Systems-Applications and Theory.

Zimmermann, H. (1996). *Fuzzy set theory and its applications* (Vol. 101). Kluwer Academic.

Zimmermann, J. (2001). *Fuzzy set theory and its applications* (Vol. 120). Springer-Verlag.