Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**Μια Εναλλακτική Διαμορφώση του Αλγόριθμου Μάθησης Artificial Immune Recognition System** **An Alternative Formulation of the Artificial Immune Recognition System Learning Algorithm**
Ονοματεπώνυμο Φοιτητή	**Δημήτρης Γιατζιτζόγλου**
Πατρώνυμο	**Γαβριήλ**
Αριθμός Μητρώου	**ΜΠΠΛ/ 16007**
Επιβλέπων	**Τσιχριντζής Γεώργιος, Καθηγητής**

Ημερομηνία Παράδοσης **Νοέμβριος 2018**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)	(υπογραφή)	(υπογραφή)
Τσιχριντζής Γεώργιος Καθηγητής	Αλέπης Ευθύμιος Επίκουρος Καθηγητής	Σωτηρόπουλος Διονύσιος Μεταδιδακτορικός Ερευνητής

ACKNOWLEDGEMENT

I want to express my gratitude to all those who contributed in the completion of this thesis. I would like to thank Prof. George Tsihrintzis, chairmain of my committee, for giving me the opportunity to work on this subject.

Especially, I would like to express my sincere gratitude to Dr. Dionisios Sotiropoulos as it is through his invaluable ideas that this work has come to fruition. He consistently allowed me to explore new ideas, but he was also always there to provide advise and assistance whenever I ran into a trouble spot or had any question.

Contents

ABSTRACT

The Artificial Immune Recognition System (AIRS) is an immune-inspired supervised learning algorithm that has been shown to perform competitively on some common datasets. The purpose of this thesis is the presentation of an alternative formulation of the Artificial Immune Recognition System, followed by a comparative study with emphasis on classification accuracy, data reduction capability and algorithmic efficiency in order to evaluate the performance difference between the proposed version of AIRS and the original AIRS classifier. The comparison suggests that the proposed formulation holds significant performance advantages over the original AIRS algorithm and further exploration of the main functionality of the algorithm could identify and address deficiencies that leave AIRS lacking in future research.

Ο AIRS είναι ένας αλγόριθμος μάθησης με επίβλεψη που έχει δείξει οτί μπόρει να πετύχει ανταγωνιστικά αποτελέσματα σε κάποια γνωστά σετ δεδομένων. Ο σκοπός αυτής της παρούσας εργασίας είναι η παρουσίαση μιας εναλλακτικής διαμόρφωσης του αλγορίθμου AIRS, συνοδευόμενη από μια συγκριτική μελέτη με έμφαση στη αξιολόγηση της ακρίβειας ταξινόμησης, της δυνατότητας μείωσης των δεδομένων και της αλγοριθμικής αποδοτικοτητάς, ακολουθούμενη από εκτίμηση, μέσω πειραματικών αποτελεσμάτων, της διαφοράς απόδοσης μεταξύ της προτεινόμενης εκδοχής του αλγορίθμου και του αυθεντικού AIRS ταξινομητη. Η σύγκριση υποδηλώνει οτί η προτεινόμενη εκδοχή έχει σημαντικό πλεονέκτημα απόδοσης σε σχέση με τον πρωτότυπο αλγόριθμο και περαιτέρω αναζήτηση της κύριας λειτουργικότητας του αλγορίθμου θα μπορούσε να εντοπίσει και να αντιμετωπίσει ανεπάρκειες του AIRS, βελτιώνοντας περαιτέρω την απόδοση του.

1 Introducton

1.1 Objective of the Thesis

The hypothesis of the thesis is that an alternative formulation of the Artificial Immune Recognition System (AIRS), an established immune-inspired supervised learning algorithm, that exhibits better results than the original formulation, can be developed by undertaking some key modifications. With this hypothesis in mind, the primary goal of this work is to present a proposed alternative implementation of the AIRS algorithm, named iAIRS, and explore the classification capabilities and other properties of the proposed new algorithm in comparison to the original version through a detailed presentation of the newly developed iAIRS learning algorithm and a comparison of its performance both on synthetic and real world data sets to the performance of the original AIRS algorithm.

The algorithm that serves as the basis for our proposed alternative formulation, AIRS, exhibits several desirable characteristics for supervised learning paradigm. These include one-shot and continuous learning, data reduction capabilities, and competitive classification accuracy (Watkins, 2001). In that regard, the main objective of this work is to develop a novel formulation of AIRS that further improves the aforementioned characteristics of AIRS and performs better, especially on the more challenging data sets.

1.2 Theoretical Background

1.2.1 Machine Learning

The ability to learn is one of most distinctive characteristics of human behaviour. It is a process that includes the acquisition of new skills through interaction with the surrounding environment, accurate representations, discovery of new knowledge through observation and experimentation and generalization of new knowledge among others.

On the other hand, machine learning is a term that applies to a broad range of computer programs that use statistical techniques to improve its performance through training with data without being explicitly programmed (Koza et al., 1996). A more formal definition of the algorithms studied in the machine learning field is provided by (Mitchell, 1997): ”A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.”

Machine learning constitutes an integral part of artificial intelligence and machine learning algorithms overcome the necessity to follow strictly static program instructions to solve problems by learning from and making prediction on data. As such, machine learning is employed in a range of applications where programming explicit algorithms is difficult or infeasible. Such applications include email filtering, computer vision and data mining.

Machine learning tasks are typically classified into two categories, unsupervised learning and supervised learning. In supervised learning, the system is presented with example data inputs and their desired outputs, or labels, and the goal is to discover a general rule that successfully maps inputs to outputs through the learning process. In unsupervised learning, on the other hand, no labels are given to the system, leaving it on its own to find patterns in its input.

1.2.2 Classification

Machine learning applications, concerning the desired output of a machine-learned system, involve some of the following:

- Classification
- Regression
- Clustering
- Density estimation
- Dimensionality reduction

In this thesis, the focus is on classification, namely the problem of identifying to which of a

set of categories a new observation belongs. Classification is typically tackled in a supervised manner where inputs are divided into two or more classes, and the system must produce a model that successfully assigns unseen data instances to one or more these classes.

The algorithms, such as the AIRS algorithm, that implement classification are usually referred as classifiers. Classification can be thought as part of the more general problem of pattern recognition, which is the assignment of some sort of output value to a presented observation.

Classification can also be considered as two separate problems, binary classification and multi-class classification. Binary classification involves only two classes, while multi-class classification involves assigning an object to one of several classes.

The performance of a classifier depends on the characteristics of the data to be classified and as such there is no single that works best for all given problems, meaning that various empirical tests have to be performed to find the suitable classifier for the specific data to be classified.

Some examples of classification algorithms include:

- Perceptron
- Support vector machines
- Quadratic classifiers
- Kernel estimation
- Decision trees
- Neural networks
- Learning vector quantization

1.2.3 Feature Vectors and Distance Metrics

In machine learning, most algorithms describe an individual observation using a feature vector of measurable properties of the observation. These properties are termed features and they may be binary, categorical, ordinal, integer-valued or real-valued.

The vector space associated with these vectors is known as the feature space and often in order to reduce the dimensionality of the feature space, dimensionality reduction techniques are employed.

Additionally, most classifiers require the use of similarity measures or distance metrics, which are real-valued functions that quantify the similarity or dissimilarity between two objects, respectively. Many machine learning algorithms, such as K Nearest Neighbour (KNN), heavily rely on the distance metric for their performance.

Distance functions provide a way to measure how close two feature vectors are, and thus are often used as error or cost functions to be minimized in an optimization problem. Some of the most well-known similarity/distance measures in machine learning include:

- Euclidean Distance
- Manhattan Distance
- Minkowski Distance
- Cosine Similarity
- Pearson Correlation Coefficient
- Jaccard Similarity

1.3 Organization

The remainder of this thesis is organized as follows:

- Section 2 presents some key natural and artificial immunological concepts that are essential to the development of AIRS and the field of artificial immune systems in general.
- Section 3 provides an overview of AIRS and presents an analytical description of the proposed novel implementation of the AIRS algorithm, named iAIRS, as well as a discussion of the key modifications made on the AIRS algorithm.

- Section 4 provides an experimental evaluation of the iAIRS algorithm and a comparative analysis between the novel iAIRS implementation and the original AIRS algorithm, demonstrating their behaviour and classification capability on synthetic and real-world data sets.
- Section 5 briefly discusses the decision process of the algorithm, offering an alternative to the unweighted majority vote process of AIRS.
- Section 6 offers conclusions and points of future research.

2 Natural and Artificial Immune Systems

2.1 Principal Immunological Concepts

In this section some basic principles of natural immune systems, fundamental to the development of AIRS algorithm, are explored.

To begin with, the most essential immunological concept embodied in the AIRS algorithm is the representation of the B cell. The large number of B cells in natural immune systems allows for adaptive immunological responses. This is possible because of a process of immunological functions like recognition, stimulation and clonal proliferation.

A population of antibodies exist on the surface of each B cell which serve as the primary binding site to foreign cells, thus providing the recognition capability of the B cell. However, it is the B cell itself, that provide the immune response to an antigen through the process of stimulation. Specifically, a B cell becomes more stimulated as the degree of binding between the antibodies and antigens on its surface intensifies. These stimulated B cells, then, undergo a cloning and mutation process and rapidly proliferate. These offsprings of the original B cells then respond to the threatening antigens.

The immunological concept of the B cell, explained above, is quintessential in the field of artificial immune systems for pattern recognition, as the recognition mechanism embodies the antibody-antigen binding principle. In particular, distance metrics, such as the Euclidean distance (Knight and Timmis, 2002) or the Hamming distance (de Castro and Zuben, 2002), between a simulated cell's feature vector and a training data instance's feature vector determine the level of binding between antibodies of the cell and antigens of the foreign presence.

However, not all the mechanisms in AIRS algorithm are necessarily precise translations from natural immune functions. This is the case with the concept of the artificial recognition ball (ARB) which can be thought as a representation of a number of B cells sharing the same antibody, with this number of cells representing the resources this ARB possesses (Timmis and Neal, 2000). The concept of ARB and resource competition will be further covered in the following sections.

As we discussed earlier, stimulated B Cells respond to a foreign presence by producing mutated offsprings to attack and destroy it. In addition to the primary immune response, a secondary response involves the cells that are the most adept at counteracting the given antigen are provided with longer lifespans and are known as memory cells (de Castro and Zuben, 2002). These memory cells are rapidly stimulated and produce a great number of offsprings when the immune system encounters the same or a similar antigen again. Most importantly, the ability of the memory cells to not only respond to the antigen that they originally respond, but also to generalize the response to similarly structured antigens.

Finally, another principal immunological concept for the development of the AIRS algorithm is that of the immune network theory, that suggests that the immune responses are not only bases on the interaction of B cells and antigens but also the interactions of B Cells with each other, as they provide a stimulation and suppression effect on one another and these effects play a crucial role on how the memory of the system is retained. One of the key points for the translation of this concept into the field of artificial immune systems is the idea of inter-cell affinity and link formation (Timmis, 2000). This link formation occurs when two ARBs show significant affinity for each other. While, the immune network theory is not explicitly incorporated in the AIRS algorithm, the concept of inter-cell affinity plays a significant role as it provides the concept of a data reduction mechanism. Specifically, the AIRS algorithm, after training, typically has fewer memory cells than the original training data instances and this is achieved through memory cell replacement based on the inter-

cell affinity of a newly introduced memory cell and an established memory cell.

In the following sections, we examine how these principles have been used in the development of artificial immune systems and presenting the precursors to the AIRS classification paradigm.

2.2 Artificial Immune Systems

Artificial immune systems (AIS) is an emerging machine learning technology inspired by theoretical immunology and observed immune functions effectively protecting our bodies from microorganisms, bacteria, and viruses and it has been widely used in machine learning, abnormal diagnosis, and production scheduling among other (Timmis et al., 2004). Work done by many theoretical immunologists, such as Perelson (1989), is a source of inspiration for researchers developing AIS as they use useful immune mechanisms as metaphors to help in the development of artificial intelligence technology for problem solving. The primary applications of Artificial Immune Systems include:

- Clustering and Classification,
- Anomaly Detection/Intrusion Detection,
- Optimization,
- Automatic Control,
- Bioinformatics,
- Information Retrieval and Data Mining,
- User Modelling/Personalized Recommendation and
- Image Processing

To apply the term AIRS to a given system, there must be a level of immunology involved, an incorporated immune principle such as the clonal and the negative selection, or an immune network model. Some of the most important properties of the immune system that are of interest to computing (Dasgupta, 1998) and are incorporated by artificial immune systems can be summarised as follows:

- Pattern Recognition: The immune system has the ability to recognise, identify and respond to a broad range of different patterns through various functions like surface molecules binding to antigens or the use of lymphokines to recognize molecular signals.
- Diversity: There are various types of elements (cells, molecules, proteins) and processes that are utilized in the maintenance of the immune system. A process that assists with diversity in the immune system is that of somatic hypermutation. In this process, the immune system responds to invading antigens with reproducing immune cells, during which, they are subjected to a somatic mutation process that allow the creation of novel receptor molecules, thus increasing the diversity of the immune receptors (Kepler and Perelson, 1993).
- Learning: The aforementioned mechanism of somatic hypermutation also allows the immune system to successfully respond to any invading pathogen. This process is known as affinity maturation (Berek and Ziegner, 1993), and allows the immune system to become increasingly better at the task of pattern recognition.
- Memory: After the immune systems responds to a given antigen, some sets of cell are endowed with increased lifespans with the purpose to provide faster and stronger immune responses to future infections by the same antigens.
- Distributivity: There is inherent distribution within the immune system and therefore there is no need of centralized control as immune cells are distributed all over the body, with each one of them specifically stimulated to respond to invading new antigens.
- Self-regulation: Immune systems dynamics are such that the immune cell population is the result of local interactions and not of centralized control. After an infection has been successfully combated, it returns to its normal state, until it is needed to respond again at some point to another antigen.
- Metadynamics: The creation of new cells and molecules is constant, as well as the elimination of those are too old or not useful. This cycle of constant creation, recruitment and death

of immune cells is known as metadynamics in theoretical immunology (Varela et al., 1988).

- Immune Network: In 1974, the immune network theory was proposed (Jerne, 1974) as an alternative explanation of how the immune system functions. It was suggested that the immune system is a dynamic system whose cells and molecules are capable of recognizing each other, thus creating a network of communication within the organism. This network forms the basis for immunological memory to be achieved.

2.3 AIS-Based Classification

There are various artificial immune models or algorithms at present, but the focus of this thesis will be on a supervised classification algorithm known as Artificial Immune Recognition System (AIRS). AIRS was proposed by Watkins et al. (2004) and is based on RLAIS (Timmis and Neal, 2001) and is one of the most competitive AIS-Based classification algorithms, especially for dealing with non-linear high-space classification problems (Goodman et al., 2003). Once the training of AIRS is completed, the set of evolved memory cells can be regarded as an extension classifier of k-near neighbour (KNN), while possessing greater generalization than traditional KNN. In pattern recognition, the k-nearest neighbors algorithm (k-NN) is a non-parametric method used for classification and regression. Concerning classification, k-nearest neighbor is an extensively used algorithm owing to its simplicity, ease of implementation and effectiveness.

The development of AIS-Based classification techniques, such as AIRS, were inspired by existing machine learning paradigms that exploit immune mechanism for unsupervised learning, where the class of data is unknown, and specifically aiNet (Castro and Zuben, 2000), an learning algorithm that employs the metaphor of the immune network theory and applies it to data clustering. Experimentation with the aiNet algorithm revealed that evolved artificial immune networks, when combined with traditional statistical analysis tools, were very effective at extracting interesting and useful clusters from data sets. Additionally, the aiNet algorithm can be utilized as a data compression technique since it produces a significantly smaller number of memory cells than that of the initial set of antigenic patterns. Moreover, aiNet may be employed as a dimension reduction technique when the utilized distance function can serve as a correlation measure between a given a pair of feature vectors. These feature vectors, however, correspond to a column-wise interpretation of the original data set, such that the resulting memory antibodies may be considered as an alternative feature set for the original data points.

AIRS is a population-based learning classifier system inspired from the learning mechanisms of human immune system, such as antigen-antibody binding, affinity maturation and clone expansion among others. The primary mechanism of AIRS for producing the evolved memory set is the concept of artificial recognition ball (ARB) through competition for limited resources among the population (Timmis and Neal, 2001). The ARB, or else B lymphocyte, is composed of an antibody molecule, a number of resources held by the ball and the current stimulation value within the antigen–antibody shape-space and it is when many antigens exist in the radius of an ARB that the characteristics of this group of antigens can be recognized and remembered. This mechanism determines the survival and reproduction of each cell, thus reinforcing the classification ability of the system.

In addition, as the primary goal of AIRS is the development of a set of competent memory cells, evolved from an initial population of molecules, there is need of a fitness concept incorporated in the system that allows the evaluation of the recognition ability of any given cell is. In the case of the AIRS algorithm, fitness is determined by the stimulation response of a antibody molecule to a presented antigenic pattern and according to this value the limited number of resources is allocated among the population of antibodies. As it is clear, this enforces a great deal of selective pressure towards better pattern recognition and ensures the survival of only those cells that will provide the highest classification accuracy.

As mentioned before, the purpose of this work is to propose a novel formulation of the established AIRS algorithm. This formulation is partly inspired by the negative selection algorithm, aiming to improve the efficiency of the learning process and achieve the development of evolved memory cells of higher quality that would improve the classification accuracy and the data reduction capabilities of the AIRS algorithm.

The Negative Selecion algorithm proposed by Forrest et al. (1994) is a supervised learning algorithm, with application to computer security, network security and anomalies detection problems, that is based on the principles of self/non-self discrimination in the natural immune system. This discrimination is possible through the generation of T-Cells, which undergo a censoring process, known as negative selection, where cells that react against self-proteins are destroyed and such only those that bind to self-proteins are allowed to live.

Basically our proposed version of AIRS, partly inspired by the concept of negative selection, involves a similar censoring technique to the aforementioned fitness concept of AIRS, such that the evaluation of the recognition ability of each antibody molecule is not only determined to its stimulation reaction to a currently presented antigenic pattern but also its stimulation reaction to the closest (to the currently presented antigen) antigenic pattern of different class, preferably minimizing it in order to further enforce the selective pressure into the evolution of antibody molecules.

In the following section, this matter is properly addressed as it includes a detailed description of the modifications made on the learning process of AIRS, as well as a thorough presentation of the proposed novel formulation of AIRS, called iAIRS.

3 An Alternative Implementation of the Artificial Immune Recognition System Learning Algorithm (iAIRS)

3.1 Overview of the AIRS Algorithm

This section presents an conceptual overview of the AIRS algorithm. Namely, it presents fundamental principles employed by the AIRS algorithm which are also employed for the most part by our alternative implementation of the algorithm. In the following sections a formal description of the proposed novel implementation, which is named iAIRS, is presented and the changes that have been made to the original AIRS algorithm are thoroughly explained. The presented overview of AIRS and the notation used in the description of iAIRS algorithm follows closely the work of Sotiropoulos and Tsihrintzis (2017a).

The fundamental principles of the AIRS algorithm may be summarized as follows:

1. Initialization:
   (a) Data normalization: All antigenic attribute vectors are normalized such that the range of each attribute lies strictly in the [0, 1] interval. In addition, the distance measured between any two antigenic patterns must also be in the [0, 1] interval.
   (b) Affinity threshold computation: The affinity threshold is calculated according to the mean affinity between antigens in the training dataset.
   (c) Memory cells initialization: The memory cell pool for each class of training antigenic patterns is set to be null.
   (d) Antibody cells initialization: The set of antibody molecules for each class of training antigenic patterns is set to be null.
2. Antigenic presentation: for each antigenic pattern do:
   (a) Memory cell identification: The memory cell presenting the highest stimulation level to the currently presented antigen. If the memory cell pool of the same class as the currently presented antigen is empty, then this antigen is incorporated within the memory cell pool. Additionally, the current antigen is denoted as the matching memory cell.
   (b) Antibody molecules generation: Once the matching memory antibody has been denoted, it is subsequently utilized in order to create a number of mutated clones. These mutated clones of the matching memory cell will eventually be added to the available antibodies repertoire. Specifically, the number of mutated clones to be created is proportional to the stimulation level of the matching memory cell to the current antigenic pattern. The mutation routine, in particular, is performed element-wise for each constituent of the antibody attribute string to be mutated. Additionally, the mutation or not of a specific antibody is randomly taken such that a number of Mutation Rate antibodies will be finally modified for each attribute. The mutation range for each attribute is also

proportional to the stimulation level of the matching memory cell, while the mutation magnitude is an uniformly distributed randomly chosen value in the [0, 1] range.

(c) Stimulations computation: Each antibody in the available repertoire is presented with the current antigen to compute the corresponding stimulation level.

(d) Actual learning process: The objective of this process is the development of a candidate memory cell which is the most successful in correctly classifying the current antigenic pattern. Hence, this step strictly concerns the antibody molecules belonging to the same class with the currently presented antigen. The following steps are repeated until the average stimulation level of the available antibodies repertoire becomes greater than or equal to a predefined stimulation threshold:

(i) Stimulations normalization: The stimulation level for each antibody is normalized across the whole antibodies repertoire.

(ii) Resource allocation: Based on the aforementioned normalized stimulation value, a number of limited resources are allocated to each antibody molecule.

(iii) Competition for limited resources: In the event of the finite number of resources distributed across the population exceeds a predefined maximum value, resources are removed from the least stimulated antibody until the total number of resources in the population returns to the number of resources allowed. Moreover, antibodies that have no resources are removed from the available antibodies repertoire.

(iv) Stimulations computation: Estimate the stimulation level for each available antibody to the currently presented antigenic pattern.

(v) Candidate memory cell identification: The antibody with the highest stimulation level to the currently presented antigen is denoted as the candidate memory cell.

(vi) Mutation of surviving antibody molecules: All antibodies in the population are given the opportunity to generate mutated clones, even if the stopping criterion is met.

(e) Memory cell introduction: The last step in the learning process is the possible introduction of the developed candidate memory cell into the memory cell pool. If the candidate memory cell is more stimulated by the current training antigen than the matching memory cell, then the candidate memory cell is incorporated within the set of memory cells. Whether the candidate memory cell eventually replaces the matching memory cell that was previously identified is dictated by the affinity threshold calculated initially, as if the affinity between the matching memory cell and the candidate memory cell is less than the product of the affinity threshold and the affinity threshold scalar, then the matching memory cell is removed from the set of memory cells and is replaced by the candidate memory cell.

3. Classification: The classification occurs using a k-Nearest Neighbour approach. The stimulation level for each memory cell to each data instance is estimated. The classification of a data instance is determined via a majority vote of the outputs of the k most stimulated memory cells.

3.2 The iAIRS Learning Algorithm

In this section, a proposed alternative formulation of the AIRS algorithm is described thoroughly as a tour of the training routine of the proposed algorithm, named iAIRS, is presented.

3.2.1 Notation

The following notation will be adapted throughout the formulation of the iAIRS learning algorithm:

- *AT* denotes the Affinity Threshold.
- *ATS* denotes the Affinity Threshold Scalar.
- *CR* denotes the Clonal Rate.
- *MR* denotes the Mutation Rate.
- *HCR* denotes the Hyper Clonal Rate.

- *TR* denotes the number of Total Resources, which is the maximum number of resources allowed within the system.
- *RA* denotes the number of Resources Allocated within the system during the training process on a particular antigenic pattern.
- *NRR* is the Number of Resources to be Removed in case RA > TR.
- *C* denotes the number of classes pertaining to a given classification problem.
- *DT* denotes the Distance Threshold, which determines the Stimulation Threshold ST_j of each antigenic pattern.
- *DTS* denotes the Distance Threshold Scalar.
- Ag $\in M_{M\times L}$ is the matrix storing the complete set of training antigenic patterns. This matrix may be considered as being constructed by the following concatenation of matrices:

$$Ag = [Ag^{(1)}; ...; Ag^{(C)}] \tag{3.1}$$

  where $Ag^{(k)} \in M_{M_k\times L}$ denotes the sub-matrix storing the training instances for the k-th class of patterns, such that:

$$M = \sum_{k=1}^{C} M_k \tag{3.2}$$

  while $Ag^{(k')}$ denotes the sub-matrix storing the training instances for all classes of patterns except the k-th class.
- Ab $\in M_{N\times L}$ is the matrix storing the available antibody repertoire for the complete set of classes pertaining to the classification problem under investigation. This matrix may be considered as being constructed by the following concatenation of matrices:

$$Ab = [Ab^{(1)}; ...; Ab^{(C)}] \tag{3.3}$$

  where $Ab^{(k)} \in M_{N_k\times L}$ denotes the sub-matrix storing the available antibodies for the k-th class of patterns, such that:

$$N = \sum_{k=1}^{C} N_k \tag{3.4}$$

- S $\in M_{1\times N}$ is the vector storing the stimulation levels to the currently presented antigenic pattern for the complete set of available antibodies in the repertoire. This vector may be considered as being constructed by the following concatenation of matrices:

$$S = [S^{(1)}, ..., S^{(C)}] \tag{3.5}$$

  where $S^{(k)} \in M_{1\times N_k}$ denotes the sub-vector storing the stimulation levels of the available antibodies for the k-th class of patterns to the current antigen which is also of the same classification.
- R $\in M_{1\times N}$ is the vector storing the resources allocated for the complete set of available antibodies in the repertoire after the presentation of the current antigenic instance. This vector may be considered as being constructed by the following concatenation of matrices:

$$R = [R^{(1)}, ..., R^{(C)}] \tag{3.6}$$

  where $R^{(k)} \in M_{1\times N_k}$ denotes the sub-vector storing the resources allocated for the available antibodies of the k-th class of patters after the presentation of the current antigenic instance which is also of the same classification.
- M $\in M_{m\times L}$ is the matrix storing the memory antibodies for each class of patterns pertaining

to a given classification problem. This matrix may be thought of as being constructed by the following concatenation of matrices:

$$M = [M^{(1)}; ...; M^{(C)}] \tag{3.7}$$

where $M^{(k)} \in M_{m_k \times L}$ denotes the sub-matrix storing the memory antibodies for the k-th class of patterns such that:

$$m = \sum_{k=1}^{C} mk \tag{3.8}$$

- s $\in M_{1 \times C}$ denotes the vector storing the average stimulation level of the available antibodies for each class of patterns, such that:

$$s = [s^{(1)}; ...; s^{(C)}] \tag{3.9}$$

3.2.2 Analytical Description of iAIRS training routine

The analytical description of our variation of AIRS algorithm, illustrated in figures 3.1, 3.2 and 3.3, involves the following steps:

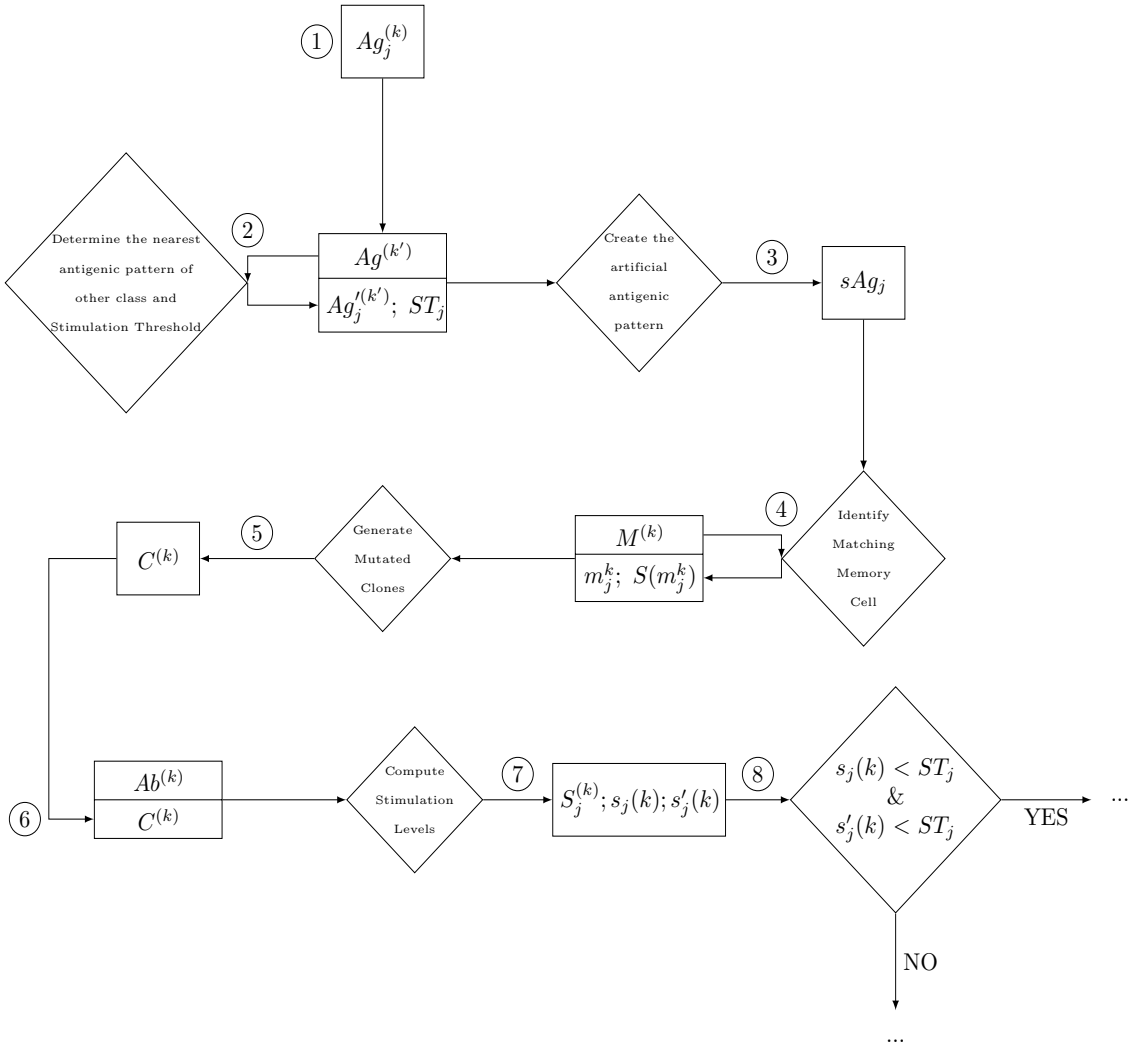

Figure 3.1: iAIRS learning algorithm 1

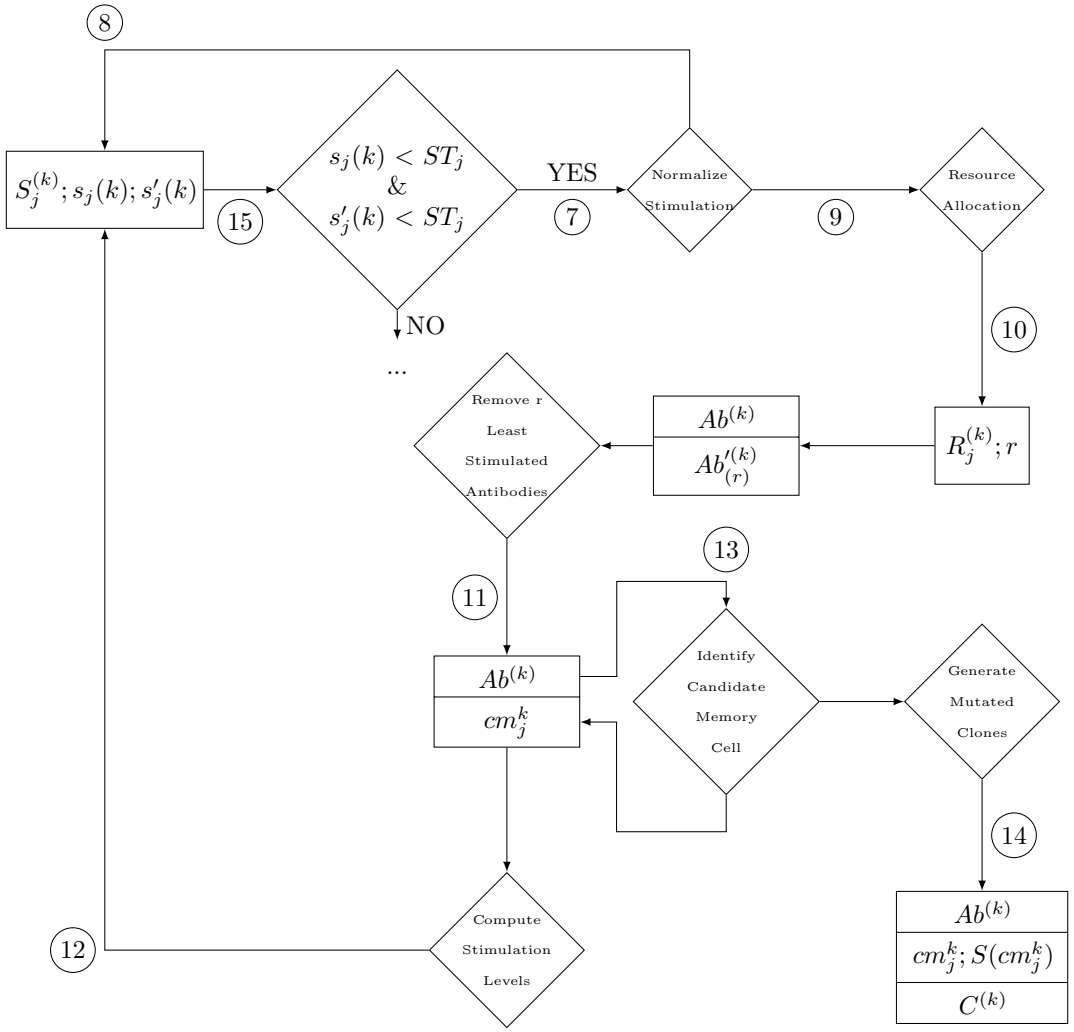

Figure 3.2: iAIRS learning algorithm 2

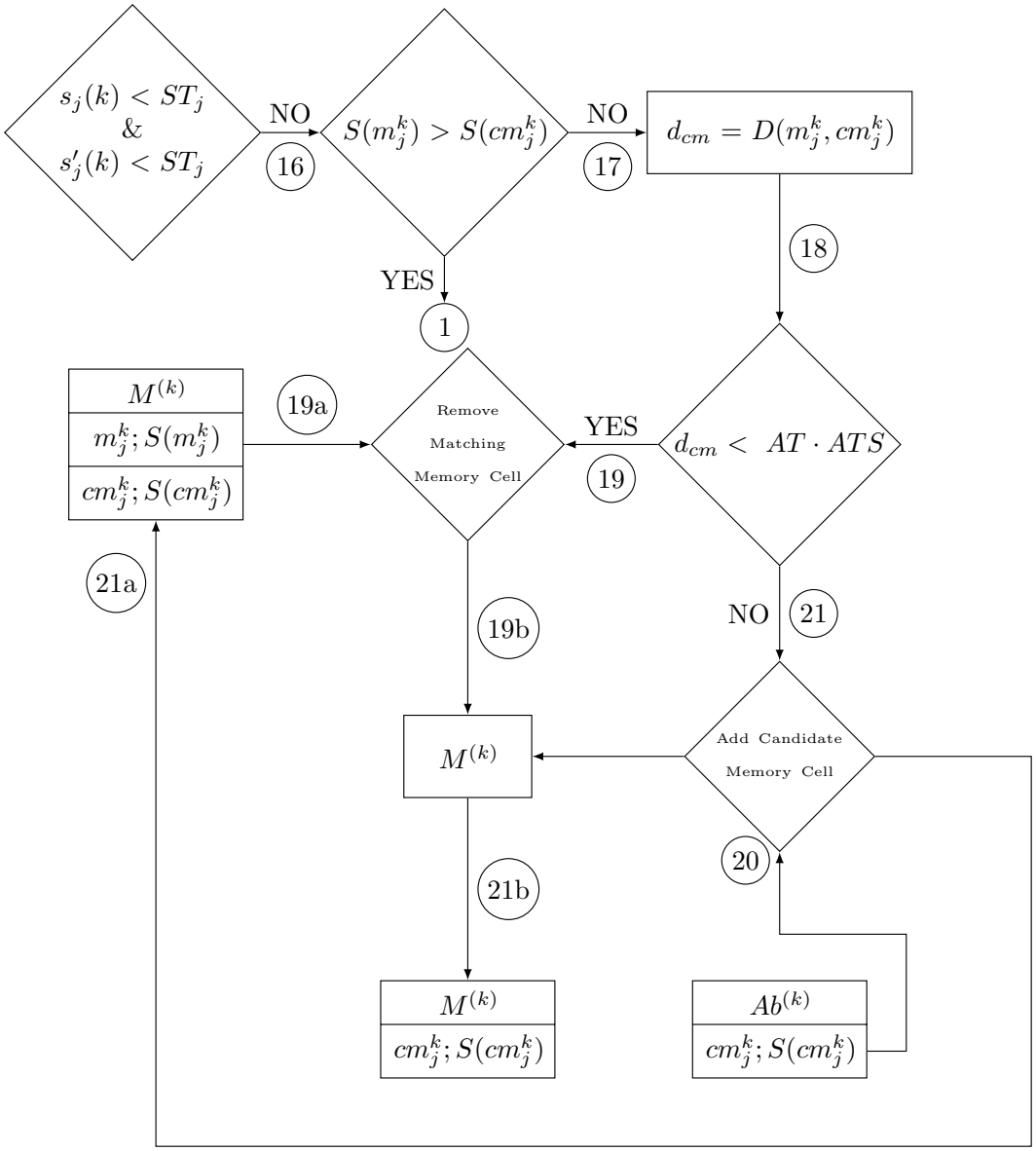

Figure 3.3: iAIRS learning algorithm 3

1. Initialization:
   (a) Compute the matrix $D \in M_{M \times X}$ of distances between all possible pairs of antigens in Ag, where the utilized distance function is:

$$D(Ag_i, Ag_j) = \frac{1}{\sqrt{L}} \|Ag_j - Ag_i\| \tag{3.10}$$

   The Normalized Euclidean Distance provides an affinity measure for which the distance value between any given pair of antigenic pattern lies within the [0, 1] interval such that:

$$0 \leq D(Ag_i, Ag_j) \leq 1, \ \forall i \in [M], \ \forall j \in [M] \tag{3.11}$$

   (b) Compute the affinity threshold (AT) according to the following equation:

$$\frac{1}{M \times M} \sum_{i=1}^{M} \sum_{j=1}^{M} D(Ag_i, Ag_j) \tag{3.12}$$

   (c) Initialize matrices Ab, M and vectors S, R according to the following equations:

$$Ab^{(k)} \leftarrow [\,], \forall k \in [C] \tag{3.13}$$
$$S^{(k)} \leftarrow [\,], \forall k \in [C] \tag{3.14}$$
$$R^{(k)} \leftarrow [\,], \forall k \in [C] \tag{3.15}$$
$$M^{(k)} \leftarrow [\,], \forall k \in [C] \tag{3.16}$$

   such that:

$$N_k \leftarrow [\,], \forall k \in [C] \tag{3.17}$$
$$m_k \leftarrow [\,], \forall k \in [C] \tag{3.18}$$

   where $[\,]$ denotes the empty matrix or the empty vector.
2. For each class k ∈ [C], of patterns do:

- For each antigenic pattern $Ag_j^{(k)} \in M_{1 \times L}$, where $j \in [M_k]$, do:
   (a) Determine the normalized euclidean distance d between $Ag_j^{(k)}$ and its nearest antigenic pattern of different class $Ag_j^{\prime(k')}$, where $k' \neq k$

$$Ag_j^{\prime(k')} = arg \min_{ag \in Ag^{(k')}} D(Ag_j^{(k)}, ag) \tag{3.19}$$

$$d = minD(Ag_j^{(k)}, Ag_j^{\prime(k')}) \tag{3.20}$$

   and compute the corresponding stimulation threshold ST_j

$$ST_j = 1 - DT \cdot d \tag{3.21}$$

   (b) Create the corresponding artificial antigenic pattern $sAg_j \in M_{1 \times L}$

$$sAg_j = Ag_j^{(k)} + DT \cdot DTS \cdot Diff(Ag_j^{(k)}, Ag_j^{\prime(k')}) \tag{3.22}$$

   where $Diff(Ag_j^{(k)}, Ag_j^{\prime(k')}) \in M_{1 \times L}$ denotes the element-wise subtraction array of $Ag_j^{(k)}$ and $Ag_j^{\prime(k')}$, according to the equation:

$$Diff(A, B) =< A^{(1)} - B^{(1)}, ..., A^{(L)} - B^{(L)} > \tag{3.23}$$

(c) Determine the matching memory cell $(\hat{m}_j^k)$

$$\hat{m}_j^k = \begin{cases} Ag_j^{(k)},\ j = 1; \\ \\ arg \max\limits_{m \in M_j^{(k)}} stimulation(Ag_j^{(k)}, m),\ otherwise. \end{cases} \tag{3.24}$$

and corresponding stimulation level $(\hat{s}_j^k)$

$$\hat{s}_j^k = 0.5 \cdot stimulation(Ag_j^{(k)}, \hat{m}_j^k) + 0.5 \cdot stimulation(sAg_j, \hat{m}_j^k) \tag{3.25}$$

The stimulation level for a given pair of vectors x, y is given by the following equation:

$$stimulation(x, y) = 1 - D(x, y) \tag{3.26}$$

such that the distance D(x,y) is once given by Eq.(3.10).

(d) Compute the matrix $C^{(k)} \in M_{Nc \times L}$ of mutated clones, originating from the matching memory cell $\hat{m}_j^k$, where the number of produced antibodies Nc is given by the following equation:

$$N_c = HCR \cdot CR \cdot \hat{s}_j^k \tag{3.27}$$

The computation of matrix $C^{(k)}$ requires the computation of two auxiliary random matrix $P_j \in M_{Nc \times L}$ and $Q_j \in M_{Nc \times L}$ such that their elements $P_j(l, p)$ and $Q_j(l, p)$ are uniformly distributed in the [0,1] interval, where $l \in [Nc]$ and $p \in [L]$. Matrix $C^{(k)}$, then, may be computed according to the following equation:

$$C^{(k)}(l, p) = \begin{cases} \hat{m}_j^k(p) + 0.5 \cdot P_j(l, p) \cdot \delta_j^k(p),\ Q_j(l, p) < MR; \\ \hat{m}_j^k(p),\ otherwise \end{cases} \tag{3.28}$$

where $\delta_j^k(p) = 0.5 \cdot Diff(sAg_j(p), \hat{m}_j^k(p)) + 0.5 \cdot Diff(Ag_j^{(k)}(p), \hat{m}_j^k(p))$ and $Diff$ is once given by Eq.(3.23) . Finally, matrix $Ab^{(k)}$ and variable N_k are updated according to the following equations:

$$Ab^{(k)} \leftarrow [Ab^{(k)}; C^{(k)}] \tag{3.29}$$

$$N_k \leftarrow N_k + N_c \tag{3.30}$$

(e) Compute the sub-matrix $S_j^{(k)} \in M_{1 \times N_k}$ containing the stimulations of the available antibodies for the k-th class of patterns after the presentation of the j-th antigen of the k-th class $Ag_j^{(k)}$ and its corresponding artificial antigen sAg_j according to the following equation:

$$S_j^{(k)}(i) = 0.5 \cdot stimulation(Ag_j^{(k)}, Ab_i^{(k)}) + 0.5 \cdot stimulation(sAg_j, Ab_i^{(k)}), \quad \forall i \in [N_k] \tag{3.31}$$

Compute the average stimulation level for the k-th class of patterns after the presentation of the j-th antigenic instance $Ag_j^{(k)}$ originating from the same class and its corresponding artificial antigen sAg_j according to the following equations:

$$s_j(k) = \sum_{i=1}^{N_k} stimulation(Ag_j^{(k)}, Ab_i^{(k)}),\ \forall i \in [N_k] \tag{3.32}$$

$$s'_j(k) = \sum_{i=1}^{N_k} stimulation(sAg_j, Ab_i^{(k)}),\ \forall i \in [N_k] \tag{3.33}$$

(f) *Actual learning process*: While $s_j(k) < ST_j$ and $s'_j(k) < ST_j$, do:

i. Normalize stimulations of the antibodies according to the following equations:

$$S_{j,min}^{(k)} = \min_{i \in [N_k]} S_j^{(k)}(i) \tag{3.34}$$

$$S_{j,max}^{(k)} = \max_{i \in [N_k]} S_j^{(k)}(i) \tag{3.35}$$

$$S_j^{(k)}(i) \leftarrow \frac{S_j^{(k)} - S_{j,min}^{(k)}}{S_{j,max}^{(k)} - S_{j,min}^{(k)}} \tag{3.36}$$

ii. Compute the sub-vector $R_j^{(k)} \in M_{1 \times N_k}$ of the available antibody resources for the k-th class of patterns after the presentation of the j-th antigen from the same class, according to the following equation:

$$R_j^{(k)}(i) = S_j^{(k)}(i) \cdot CR,\ \forall i \in [N_k] \tag{3.37}$$

iii. Compute the number of resources allocated by the complete set of antibodies for the current class and antigen according to the following equation:

$$RA = \sum_{i=1}^{N_k} R_j^{(k)}(i) \tag{3.38}$$

iv. Compute the number of resources to be removed according to the following equation:

$$NRR = RA - TR \tag{3.39}$$

v. Re-order the elements in matrix $Ab^{(k)}$ and vectors $S^{(k)}$ and $R^{(k)}$ according to the permutation $\pi : [N_k] \rightarrow [N_k]$, such that:

$$Ab^{(k)} \leftarrow \pi(Ab^{(k)}) \tag{3.40}$$

$$S_j^{(k)} \leftarrow \pi(S_j^{(k)}) \tag{3.41}$$

$$R_j^{(k)} \leftarrow \pi(R_j^{(k)}) \tag{3.42}$$

This permutation rearranges the elements in $R_j^{(k)}$, so that:

$$R_j^{(k)}(i) \leq R_j^{(k)}(i+1),\ \forall i \in [N_k - 1] \tag{3.43}$$

vi. Compute the vector $I_j^{(k)} \in M_{1 \times \mu}$, where $\mu \leq Nk$, such that:

$$I_j^{(k)} = \left\{ r \in [N_k] : \sum_{i=1}^{r} R_j^{(k)}(i) < NRR \right\} \tag{3.44}$$

vii. Compute the optimal value $\hat{r}$, given by the following equation:

$$\hat{r} = arg \max_{i \in I_j^{(k)}} \sigma_j^k(r) \tag{3.45}$$

where $\sigma_j^k(r)$ denotes the partial sum of the r first elements in $R_j^{(k)}$ given by the following equation:

$$\sigma_j^k(r) = \sum_{i=1}^{r} R_j^{(k)} \tag{3.46}$$

viii. Compute the remaining number of allocated resources according to the following equation:

$$RA = \sum_{i=\hat{r}+1}^{N_k} R_j^{(k)} \tag{3.47}$$

This number corresponds to the amount of system wide resources after the removal of the $\hat{r}$ least stimulated antibodies. This value, however, yields a remaining number of resources to be removed which is, once again, given by Eq.(3.39) such that:

$$0 < NRR < R_j^{(k)}(\hat{r}+1) \tag{3.48}$$

ix. Eliminate $\hat{r}$ + 1-th antibody, such that:

$$R_j^{(k)}(\hat{r}+1) \leftarrow R_j^{(k)}(\hat{r}+1) - NRR \tag{3.49}$$

which finally yields $NRR = 0$ and $R_j^{(k)}(\hat{r}+1) > 0$.

x. Remove the $\hat{r}$ least stimulated elements from the available antibodies repertoire corresponding to the current class and antigen according to the following equation:

$$Ab^{(k)} \leftarrow Ab^{(k)}\{Ab_1^{(k)}, ..., Ab_{\hat{r}}^{(k)}\} \tag{3.50}$$

and re-estimate the number of antibodies for the current class of patterns as:

$$N_k \leftarrow N_k - \hat{r} \tag{3.51}$$

xi. Re-estimate vector $S_j^{(k)}$ containing the stimulations for the available antibodies against the current antigen and associated class according to Eq.(3.31).Subsequently, re-estimate the corresponding average stimulation levels $s_j^{(k)}$ and $s_j'^{(k)}$ according to Eq.(3.32) and Eq.(3.33).

xii. Determine the candidate memory cell $\tilde{m}_j^k$ and corresponding stimulation level $\tilde{s}_j^k$ according to the following equations:

$$\tilde{m}_j^k = arg \max_{m \in Ab^{(k)}} stimulation(Ag_j^{(k)}, m) \tag{3.52}$$

$$\tilde{s}_j^k = stimulation(Ag_j^{(k)}, \tilde{m}_j^k) \tag{3.53}$$

xiii. Compute the matrix $\tilde{C}^{(k)} \in M_{N_c \times L}$ of mutated offsprings corresponding to the surviving antibodies, such that:

$$N_c = \sum_{i=1}^{N_k} N_c(i) \tag{3.54}$$

The number of mutated clones to be produced from each surviving antibody will be given by:

$$N_c(i) = S_j^{(k)}(i) \cdot CR \tag{3.55}$$

In this context, matrix $\tilde{C}^{(k)}$ of mutated clones may be considered as constructed through the following concatenation process:

$$\tilde{C}^{(k)} = [\tilde{C}_1^{(k)}; ...; \tilde{C}_{N_k}^{(k)}] \tag{3.56}$$

Determining sub-matrices $\tilde{C}_i^{(k)}, \forall i : Nc(i) \neq 0$ requires the computation of two random matrices $P_j \in M_{Nc \times L}$ and $Q_j \in M_{Nc \times L}$ such that their elements $Q_j(l,p)$ are uniformly distributed in the [0,1] interval where $l \in [Nc]$ and $p \in [L]$. The elements of matrix $\tilde{C}_i^{(k)}$ may be then computed by the following equation:

$$\tilde{C}_i^{(k)}(l,p) = \begin{cases} Ab_i^{(k)}(p) + 0.5 \cdot P_j(l,p) \cdot \delta_j^k(p), \ Q_j(l,p) < MR; \\ Ab_i^{(k)}(p), \ otherwise \end{cases} \tag{3.57}$$

where $\delta_j^k(p) = 0.5 \cdot Diff(sAg_j(p), Ab_i^{(k)}(p)) + 0.5 \cdot Diff(Ag_j^{(k)}(p), Ab_i^{(k)}(p))$. Finally, matrix $Ab^{(k)}$ and variable N_k are updated according to the following equations:

$$Ab^{(k)} \leftarrow [Ab^{(k)}; \tilde{C}^{(k)}] \tag{3.58}$$

$$N_k \leftarrow N_k + N_c \tag{3.59}$$

(g) Memory cell introduction: If $\tilde{s}_j^{(k)} > \tilde{s}_j'^{(k)}$, where $\tilde{s}_j'^{(k)} = stimulation(Ag_j^{(k)}, \hat{m}_j^k)$, then:

i. $d_{cm} = D(\tilde{m}_j^{(k)}, \hat{m}_j^k)$

ii. If $d_{cm} < ATS \cdot AT$ then:

– Update the sub-matrix $M^{(k)}$ of memory cells for the class of the currently presented matrix and the corresponding number of memory cells m_k according to the following equations:

$$M^{(k)} \leftarrow M^{(k)} \setminus \{\hat{m}_j^k\} \tag{3.60}$$

$$m_k \leftarrow m_k - 1 \tag{3.61}$$

These operations correspond to the substitution of the matching memory cell with the candidate memory cell.

iii. Incorporate the candidate memory cell within the sub-matrix of memory cells pertaining to the same class as the currently presented antigenic pattern according to the following equations:

$$M^{(k)} \leftarrow [M^{(k)}; \hat{m}_j^k] \tag{3.62}$$

$$m_k \leftarrow m_k + 1 \tag{3.63}$$

3.3 Points of Change

In this section, the main points of change that have been made to the original AIRS algorithm and differentiate the proposed alternative iAIRS from the original version are thoroughly discussed.

3.3.1 Stopping Criterion and Evolution of Memory Cells

Several studies in the field of Artificial Immune Systems have offered insight into why AIRS performs as it does and identified some deficiencies. It has been suggested (McEwan and Hart, 2009) that the algorithm has an overly elitist selection criteria as only the best matching memory cell initiates a response, and only the best matching mutant becomes a candidate memory cell. Also, ”best” implies simply closeness rather than a more general criterion of ”fittest”, while AIRS is further limited because there is no feedback between data of different classes. The training process is blind to any ambiguity in the regions where data of different classes overlap. Therefore, in order to improve compression, generalisation and discrimination it could be beneficial to implement a variant stimulation threshold to be lower in coherent, homogeneous regions and higher in ambiguous regions by utilizing information about data of different class when a given antigenic pattern is presented. In order to address some of these deficiencies, modifications to be made to the training process of the algorithm are proposed.

In the original version of AIRS, Stimulation Threshold is a hyper-parameter between 0 and 1 and is used as the main stopping criterion for the training of an antigen, as only when the average stimulation level of the antibodies of each class is greater than the Stimulation Threshold, does training of the particular antigen stops. In addition, Stimulation Threshold as a hyper-parameter is a user-defined value that remains fixed throughout the training process of all antigenic patterns, meaning the stopping criterion is the same for all antigenic patterns used for training and the specific conditions of each antigen are not involved in shaping the stopping criterion that determines when its training can be stopped. Therefore, a more flexible stopping criterion, one that is shaped by the specific conditions surrounding each antigen, is implemented in iAIRS.

To begin with, Stimulation Threshold as an user-defined hyper-parameter is substituted for two new hyper-parameters, named Distance Threshold and Distance Threshold Scalar, that define a variant stopping criterion adapted to the conditions surrounding the currently presented antigen. More specifically, for each antigenic pattern presented, Stimulation Threshold is calculated by determining the normalized euclidean distance d between the currently presented antigenic pattern and its closest antigenic pattern of different class as defined by Eq.3.21. Distance Threshold is a value that is multiplied by d to define the Stimulation Threshold for each antigen.

The reasoning behind this modification is that because the Stimulation Threshold defines how ”close” or stimulated to a specific antigen the candidate memory cell is going to be, with the implementation of this modification, the ”closer” an antigen is to an antigenic pattern of different class the stricter the criterion to stop its training is going to be while the developed candidate memory cell is going to be correspondingly ”closer” to the currently presented antigen. Essentially, by making the Stimulation Threshold proportional to the distance between the specific antigen and its closest antigenic pattern of different class a possible overlapping between the candidate memory cell and an antigenic pattern of different class can be avoided as long as Distance Threshold value is between 0 and 1 without unnecessarily applying a strict stopping criterion to all training antigenic patterns, but only to those that is needed. As a result, it allows for a greater generalization in coherent areas while at the same time providing finer granularity in the more ambiguous areas.

Another implemented change affecting the stopping criterion and the development of the candidate memory cell is the creation of a secondary artificial antigenic pattern sAg_j for each presented antigenic pattern $Ag_j^{(k)}$ as described by Eq. 3.22. To further clarify, the artificial antigenic pattern is created so that its euclidean distance from the antigenic pattern $Ag_j^{(k)}$ is equal to $DTS \cdot DT \cdot d$, while its euclidean distance from the closest antigenic pattern of different class $Ag_j^{\prime(k')}$ is equal to $DTS \cdot DT \cdot d + DT \cdot d$, meaning that is an synthetic antigenic pattern created to be further from the closest antigenic pattern of different class $Ag_j^{\prime(k')}$ than the currently presented antigen $Ag_j^{(k)}$ by $DT \cdot d$ (or differently put $1 - ST_j$). DTS is the value that determines how ”far” from the cur-

rently presented antigen $Ag_j^{(k)}$, and also the closest antigen of different class, $Ag_j'^{(k')}$, the artificial antigenic pattern is going to be.

Furthermore, in the iAIRS implementation the stimulation level is calculated as the average of two stimulation sub-levels, one for antibodies presented with the current antigenic pattern $Ag_j^{(k)}$ and one for antibodies presented with the artificial antigenic pattern sAg_j, as given by the equations Eq.3.24 and Eq. 3.31.

Finally, for the training of a currently presented antigen $Ag_j^{(k)}$ to be stopped not only the average stimulation level of the available antibodies to the currently presented antigen must be greater than the stimulation threshold ST_j, but the average stimulation level of the same available antibodies to the artificial antigenic pattern sAg_j must also be greater than the stimulation threshold.

The purpose of these changes is to target and train for antibodies that not only are adequately stimulated to a specific antigen but at the same time are adequately remote from the closest antigenic pattern of different class to this specific antigen. In the following figures, the effect of these changes to the stopping criterion and development of a candidate memory antibody are illustrated.

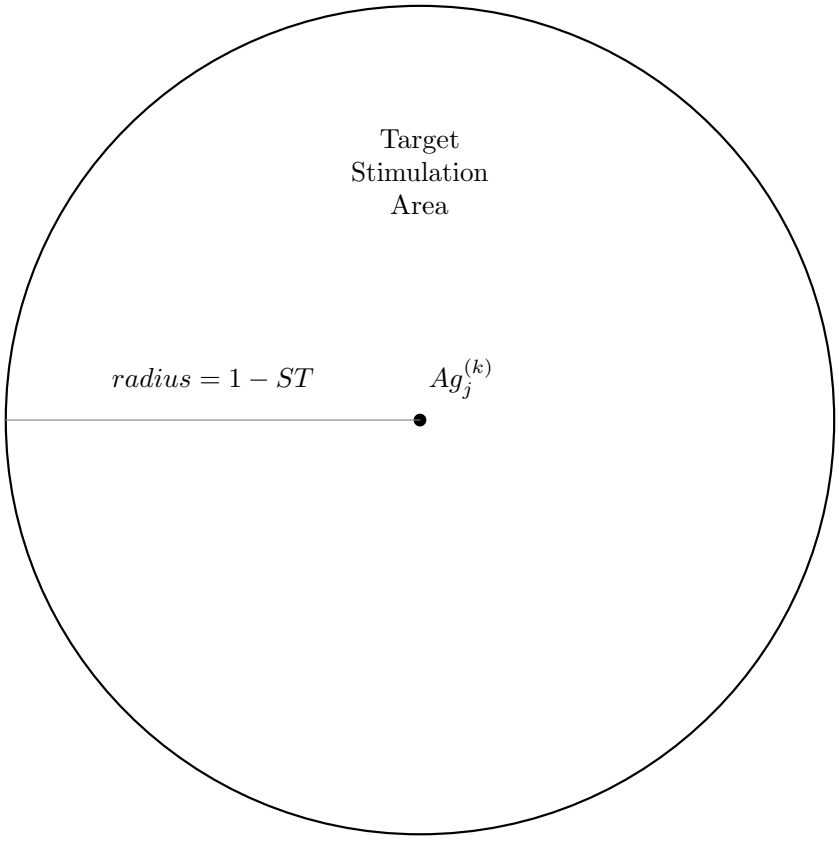

Figure 3.4: Target Stimulation Area-AIRS

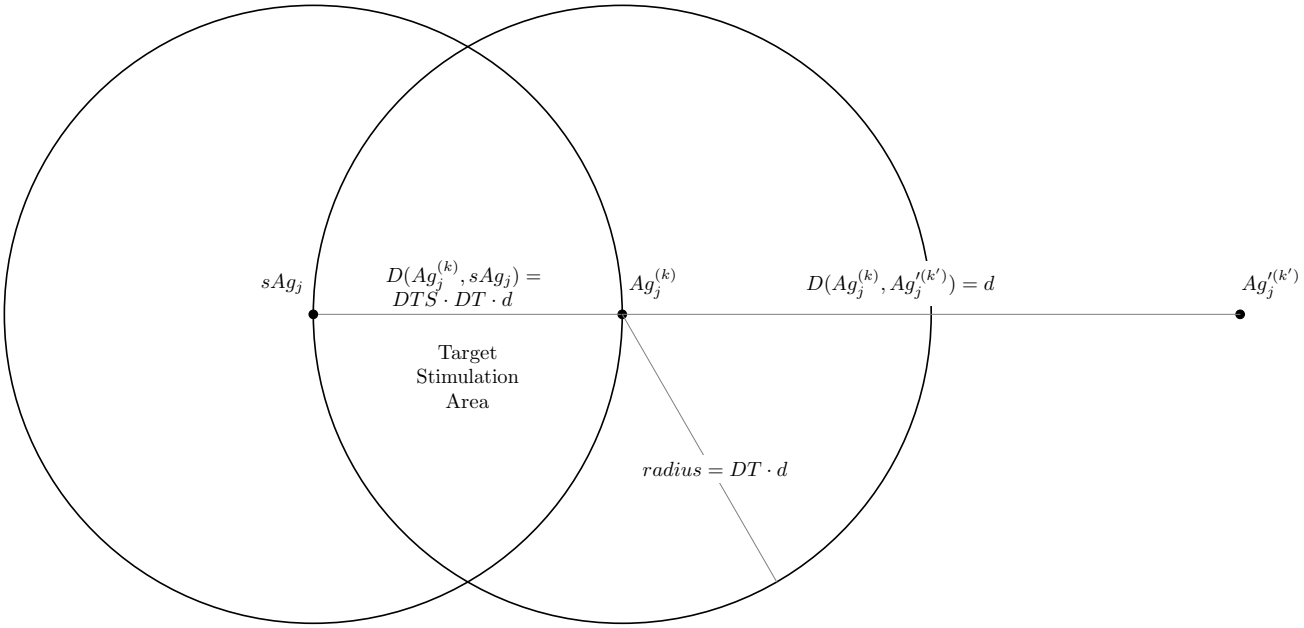

Figure 3.5: Target Stimulation Area-iAIRS,DT=0.5,DTS=1

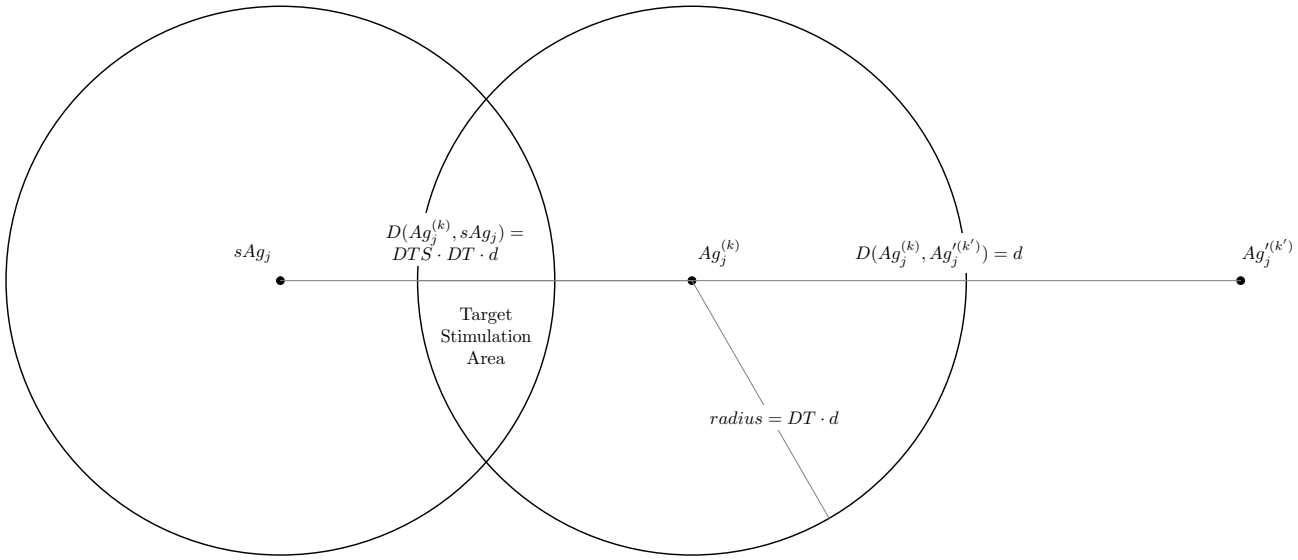

Figure 3.6: Target Stimulation Area-iAIRS,DT=0.5,DTS=1.5

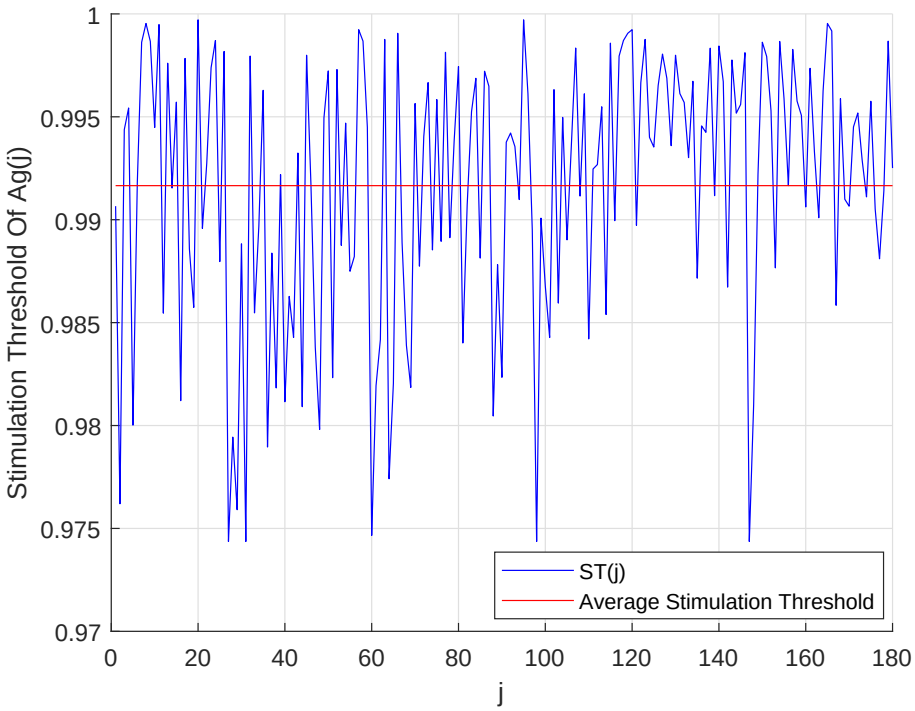

Figure 3.7: Stimulation Threshold of Each Antigen-DT=0.1

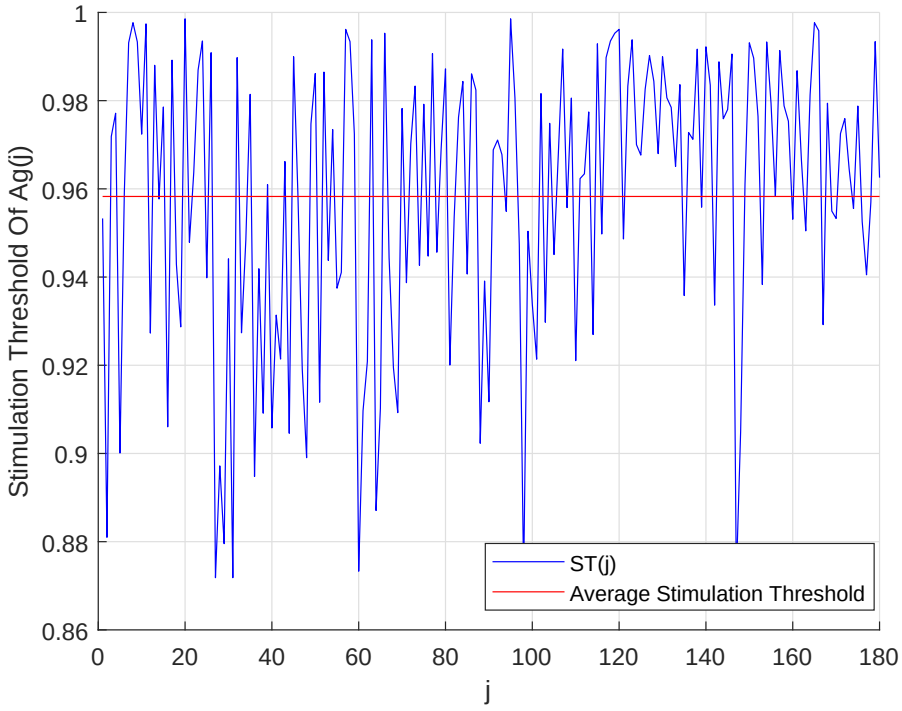

Figure 3.8: Stimulation Threshold of Each Antigen-DT=0.5

3.3.2 Mutation Process

The mutation process of AIRS is another possible area of modification and improvement. Studies (McEwan and Hart, 2009) have questioned the purpose of the random search nature of the mutation process of AIRS, suggesting that it has almost no positive effect on classifiers performance.

AIRS employs an mutation mechanism that is based on purely random changes meaning decreasing of the quality of cells is possible as they may randomly move to less productive areas in the search space. On top of that, AIRS has an particularly elitist selection criteria where only the best matching mutant becomes a candidate memory cell. One hypothesis of this work is that the mutation mechanism, as implemented in AIRS, is unnecessarily hindering the algorithmic efficiency and needs to be modified. The proposed modification is that instead of purely random mutations, more focused yet still random mutations of features could be applied.

In the mutation routine of the original version of AIRS, the mutation range for each element to be altered is proportional to the stimulation level of the matching memory cell or surviving antibody, while the mutation magnitude is a value randomly chosen according to the uniform distribution in the [0,1] interval, meaning that the mutated clones can end up moving away from their original position in any possible direction. As a result, a number of antibodies could end up less stimulated to the currently presented antigen $Ag_j^{(k)}$ and of poorer quality than before the mutation occurred. This negative impacts the run-time complexity of the algorithm, especially when dealing with high-dimensional antigenic patterns, as the purely random changes of features mean that satisfying the stopping criterion is becoming difficult to be achieved. Therefore, to achieve a more focused and directed exploration of the search space, the mutation process was modified accordingly.

The modified mutation process is described by equation 3.28. Concisely, the mutation range for each attribute is proportional to the average of the attribute difference between the current antigen $Ag_j^{(k)}$ and the matching memory cell (or antibody) and the attribute difference of the artificial antigenic pattern sAg_j and the matching memory cell (or antibody). As in the original version, the magnitude of the mutation for each attribute is a value randomly chosen according to the uniform distribution in the [0, 1] interval. The goal of implementing this modification, is to allow for a more focused and efficient refinement of the cells while preserving the diversification through exploration. The aforementioned changes are illustrated in the following figures.

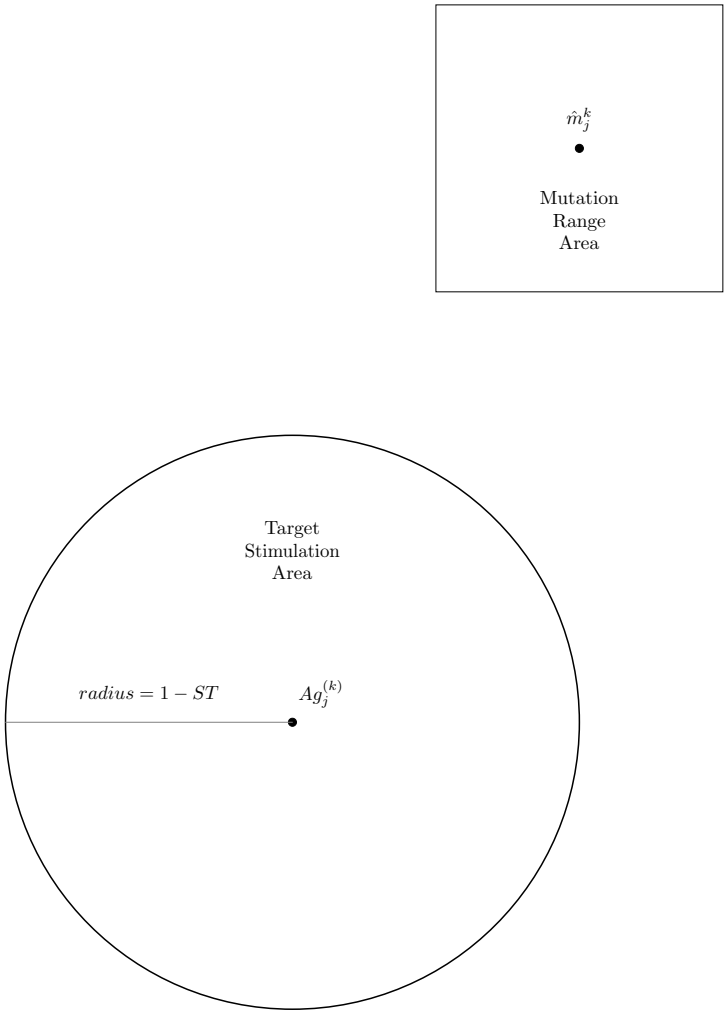

Figure 3.9: Mutation Range Area-AIRS

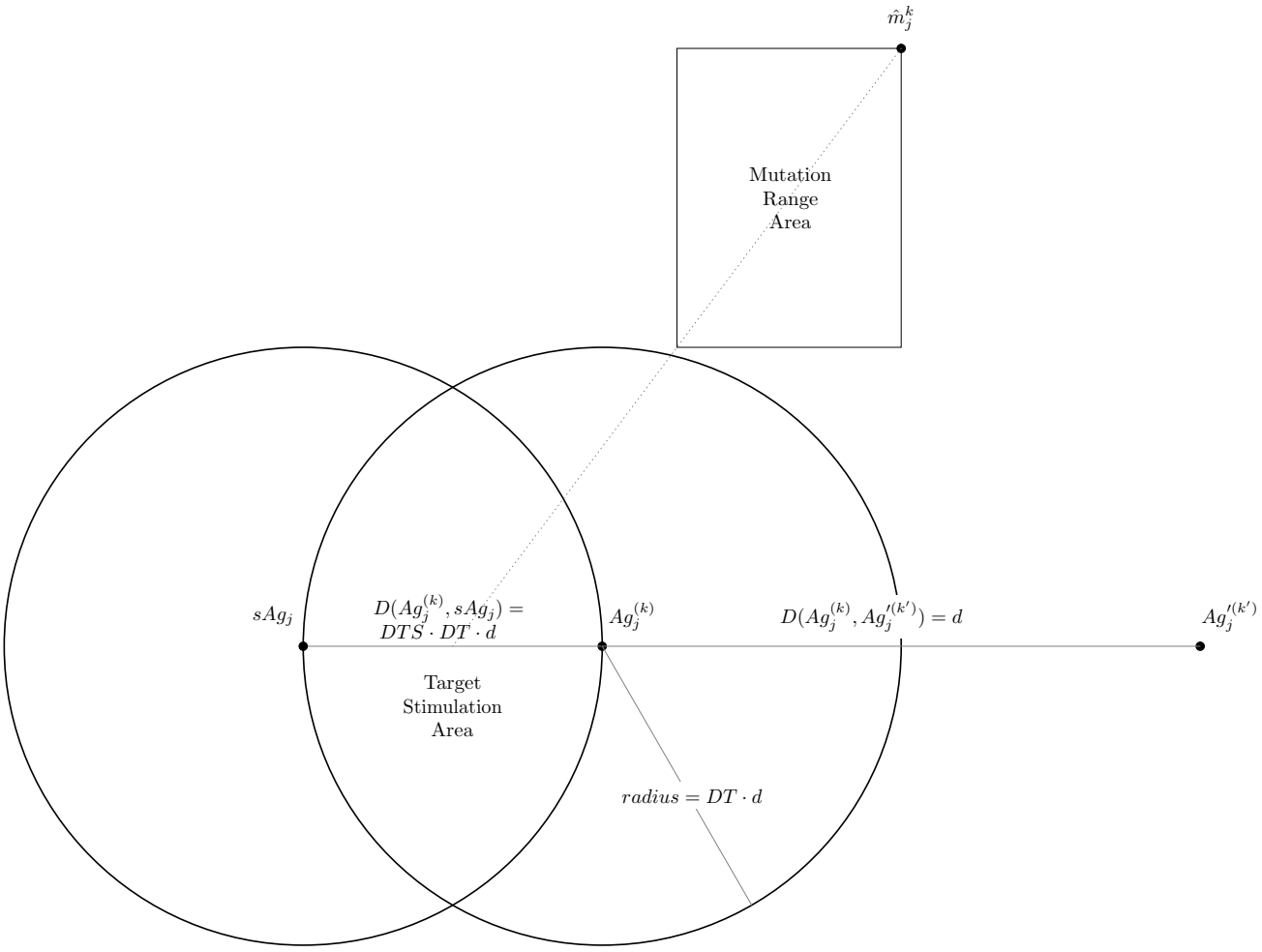

Figure 3.10: Mutation Range Area-iAIRS

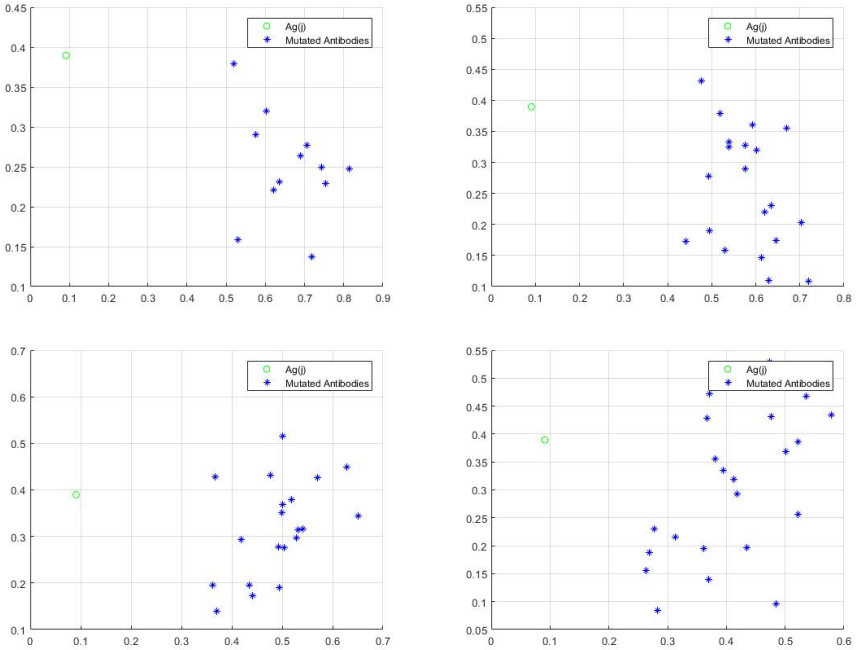

Figure 3.11: Mutation Phases-AIRS

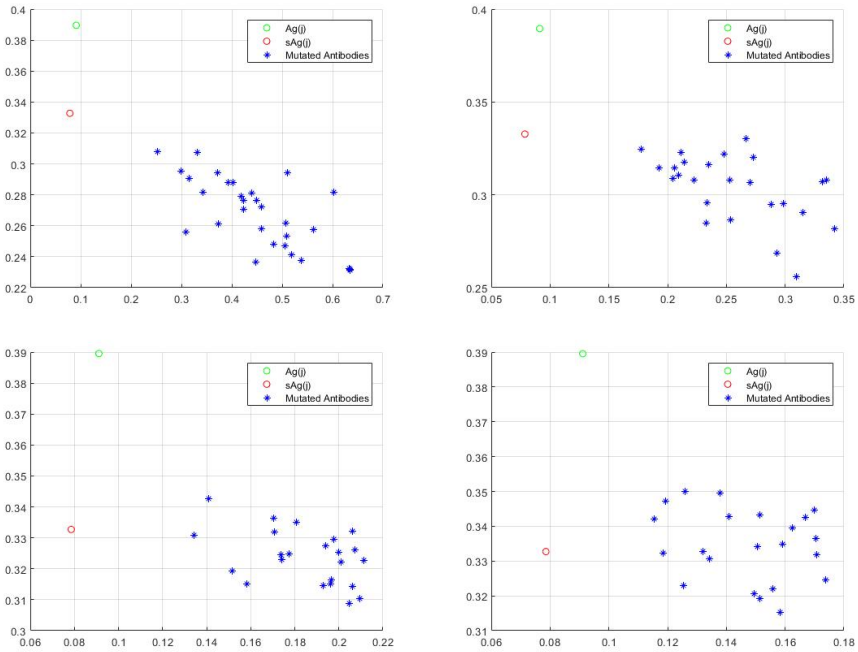

Figure 3.12: Mutation Phases-iAIRS

4 Experimental Evaluation

4.1 Objective and Methodology of Experimentation

This section presents a number of experimental test results performed. The main objective of these experimental tests was the performance comparison between our proposed implementation of the AIRS algorithm (iAIRS) and the original AIRS, as well as the evaluation of the novel implementation, iAIRS, as a valid alternative machine learning paradigm. The focus of this comparative analysis will be on three of the more important features of the AIRS algorithm: classification accuracy, data reduction and algorithmic efficiency.

In order to achieve this, the experimental evaluation included extensive tests on various test data sets, both synthetic and real. To allow for comparison between the two versions of the algorithm, the experiments performed on the original AIRS were also performed on the new formulation of AIRS (iAIRS) following a certain methodology.

The methodology followed involved grid searching to optimally configure the system and find the best test results of both algorithms on each data set. In addition, K-Fold cross-validation was applied to estimate the generalization performance. More specifically, for each testing data set a ten-fold cross validation was employed with each result representing an average of one run across these ten divisions. Finally, the random number generation was set to default settings in order to insure comparability and reproducibility of the test results.

4.1.1 Grid Searching

Grid search is one of the most widely used strategies for hyper-parameter optimization and is an exhaustive searching through a manually specified subset of the hyper-parameter space of a learning algorithm.

In the original AIRS, grid searching was employed to optimally tune two hyper-parameters which are Stimulation Threshold (ST) and Affinity Threshold Scalar (ATS). While in the proposed iAIRS implementation, grid searching was employed to optimally tune three hyper-parameters which are Distance Threshold (DT), Affinity Threshold Scalar (ATS) and Distance Threshold Scalar. All the other hyper-parameters values were fixed for all conducted tests that will be presented in the following sections, as shown in table 4.1.

Common Testing Input Values	
Training data percentage	0.9
Clonal rate	4
Hyper mutation rate	4
Total resources	20
Nearest Neighbors	5
Mutation Rate	0.6

Table 4.1: Testing Input Values

In addition, because of the different nature of the two algorithms and especially the differences between the Stimulation Threshold in AIRS and the Distance Threshold in iAIRS as described in section 3.3.1 the manually specified subset for the Stimulation Threshold hyper-parameter in AIRS grid searching consisted of the average of Stimulation Thresholds (STj) of each training antigen for each corresponding Distance Threshold tested in iAIRS grid searching, as illustrated in figure 3.7. All other hyper-parameter values were the same on both versions.

Ultimately, the objective was to find the optimal configuration for both algorithms that yielded the best test results and compare them in terms of test accuracy, number of memory antibodies and overall algorithmic efficiency.

4.1.2 K-Fold Cross-Validation

K-fold cross-validation is a validation technique for assessing how the results of a statistical analysis will generalize to an independent data set. In other words, it is used to estimate the performance of a predictive model in practice. Specifically, it involves the random partitioning of the data set into k equal sized sub-samples. Of the k sub-samples, one of them is retained as the testing set, the sub-sample of the original dataset against which the model is tested, and the rest k-1 sub-samples are retained as the training set that will be used to train the model. This process is repeated k times, with each of the k sub-samples used only once as the test data. Ultimately, the k test results will be averaged to produce a single test result. The main advantage of this technique is that all observations are used exactly once for testing.

In our test experiments, for each class we used 90% of its antigens for training the model, rounding up if necessary to get a whole number of antigens. To achieve this, the process explained above will be repeated ten times for each class. For example if a class consists of 100 antigens, in each iteration 10 antigens will be regarded as test data and the rest will be regarded as training data. More specifically, in the first iteration the first ten of the antigens will be chosen as the test data and so forth till the tenth iteration as shown by the following figure.

Figure 4.1: 10-fold Cross-Validation

4.2 The Test Data Sets

4.2.1 Synthetic Test Data Sets

In our comparative analysis, we used both synthetic and real test data sets. The reason we used synthetic test data sets was that it allowed for creating various difficult classification problems to test against.

For the creation of the synthetic data sets the mvnrnd function in MATLAB was utilized, a function which generates multivariate normal distributed random numbers. The input values are the mean MU which is a $1-by-d$ vector with d being the number of dimensions, the covariance matrix

SIGMA that is $d - by - d$ matrix and CASES which is the number of antigens to be generated. In our case, SIGMA is the identity covariance matrix meaning all dimensions are statistically independent, and the variance of the data along each of the dimensions is equal to one. The return is a $CASES - by - d$ matrix, which is essentially a class of synthetic data. The process is repeated for each class ,finally creating the complete synthetic data set.

The process is described by the following equations.

$$Ag^k = mvnrnd(MU, SIGMA, CASES) \tag{4.1}$$

$$MU = [mu^1, mu^2, ..., mu^d],\ where\ mu^1 = mu^2 = ... = mu^d \tag{4.2}$$

$$SIGMA = \mathbb{1}^d \tag{4.3}$$

A example of two synthetic classes of 2-d synthetic classes with $MU_1 = 2$ and $MU_2 = 3$ is shown in the following figure.

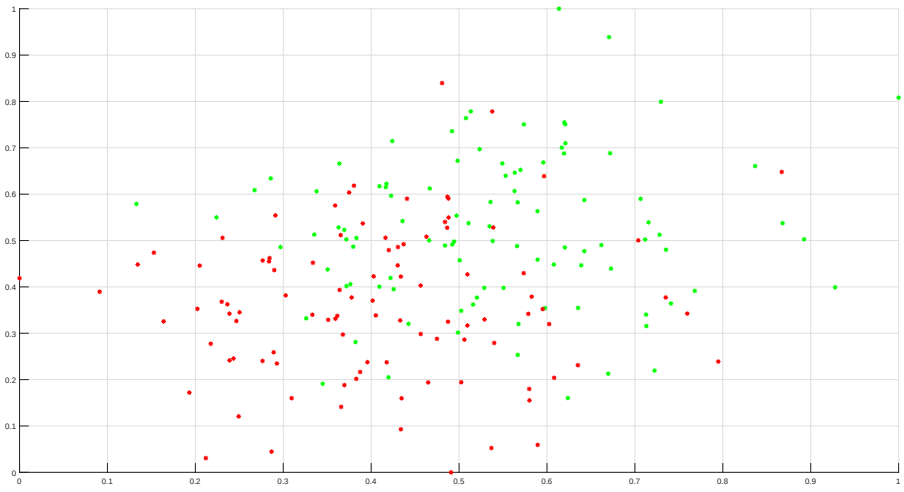

Figure 4.2: Synthetic 2-D Data Set MU_1=2,MU_2=3

4.2.2 Real Test Data Sets

In our comparative analysis, we also included testing experiments on two real-world data sets. The first is an open collection of 1000 pieces from 10 classes of modern music. More specifically, this collection contains 100 pieces, each of thirty second duration, from each of the ten classes of western music. Each piece is represented by a specific set of 30 objective features as shown below.

Class ID	Label
C1	Blues
C2	Classical
C3	Country
C4	Disco
C5	Hip-Hop
C6	Jazz
C7	Metal
C8	Pop
C9	Reggae
C10	Rock

Table 4.2: Classes of western music

Feature ID	Feature name
1	Mean Centroid
2	Mean Rolloff
3	Mean Flux
4	Mean Zero-crossings
5	STD of Centroid
6	STD of Rolloff
7	STD of Flux
8	STD of Zero-crossings
9	Low Energy
[10 . . . 19]	MFCCs
20	Beat A0
21	Beat A1
22	Beat RA
23	Beat P1
24	Beat P2
25	Beat Sum
26	Pitch FA0
27	Pitch UP0
28	Pitch FP0
29	Pitch IP0
30	Pitch Sum

Table 4.3: Feature vector constituents-feature.mat

The other real data set we use is the Wisconsin Diagnostic Breast Cancer data set. The data set used in this paper is publicly available and was created by Dr. William H. Wolberg, physician at the University Of Wisconsin Hospital at Madison, Wisconsin, USA.

This data set contains 569 observations regarding breast cancer diagnosis from two classes, benign and malign. These observations are represented by 30 features that are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. The benign class contains 212 observations while the malign class contains 356, meaning the data set is imbalanced. Each feature is evaluated on a scale of 1 to 10, with 1 being the closest to benign and 10 the closest to malignant.

Feature ID	Feature name
1	Radius
2	Texture
3	Perimeter
4	Area
5	Smoothness
6	Compactness
7	Concavity
8	Concave Points
9	Symmetry
10	Fractal Dimension

Table 4.4: Feature vector constituents-WDBC.mat

4.2.3 Normalization

Normalization of data consists of linearly transforming the data according to the following equation.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{4.4}$$

where x is the original value and x' is the normalized value.

The objective of this method is to rescale the range of features in [0,1]. This is necessary because our classifiers calculate the distance between two points utilizing the Euclidean distance. If one of the features has a broad range of values, the distance will be governed by this particular feature. Therefore, the range of all features should be normalized so that each feature contributes approximately proportionately to the final distance.

4.3 Experimental Results

4.3.1 Synthetic Test Data Set Results

To allow for comparison between the two versions of the algorithm, the same experiments that were performed on the original AIRS were also performed on the alternative formulation iAIRS, testing them against various synthetic data sets previously discussed in Section 4.2.1. In this section, we present both the best classification results against testing subsets and the best classification results against all data points for each version of the algorithm following the methodology discussed in Section 4.1.

Synthetic Test 1 - Data Set Information			
Classes	MU	Cases	Dimensions
C1	2	100	2
C2	4	100	2

Table 4.5: Data set information-Synthetic Test 1

Synthetic Test 1 - C1 vs C2	
AIRS best overall result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.02
Clonal rate	4
Hyper mutation rate	4
Stimulation threshold	0.9917
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	90%
Training Result	92.44%
Overall Result	**92.20**%
Mean Memory Antibodies	141.5

Synthetic Test 1 - C1 vs C2	
AIRS best test result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.21
Clonal rate	4
Hyper mutation rate	4
Stimulation threshold	0.9833
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	**92%**
Training Result	91.72%
Overall Result	91.75%
Mean Memory Antibodies	44.9

Table 4.6: AIRS classification results-Synthetic Test 1

Synthetic Test 1 - C1 vs C2	
iAIRS best overall result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.03
Clonal rate	4
Hyper mutation rate	4
Distance threshold	0.2
Distance threshold scalar	0.25
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	91.5%
Training Result	92.83%
Overall Result	**92.70%**
Mean Memory Antibodies	128.1

Synthetic Test 1 - C1 vs C2	
iAIRS best test result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.22
Clonal rate	4
Hyper mutation rate	4
Distance threshold	0.3
Distance threshold scalar	0.5
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	**92.50%**
Training Result	91.44%
Overall Result	91.55%
Mean Memory Antibodies	38.6

Table 4.7: iAIRS classification results-Synthetic Test 1

Synthetic Test 2 - Data Set Information			
Classes	MU	Cases	Dimensions
C1	2	100	50
C2	2.5	100	50

Table 4.8: Data set information-Synthetic Test 2

Synthetic Test 2 - C1 vs C2	
AIRS best overall result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.01
Clonal rate	4
Hyper mutation rate	4
Stimulation threshold	0.9910
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	93.50%
Training Result	93.11%
Overall Result	**93.15**%
Mean Memory Antibodies	180

Synthetic Test 2 - C1 vs C2	
AIRS best test result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.20
Clonal rate	4
Hyper mutation rate	4
Stimulation threshold	0.9819
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	**94%**
Training Result	92.83%
Overall Result	92.95%
Mean Memory Antibodies	180

Table 4.9: AIRS classification results-Synthetic Test 2

Synthetic Test 2 - C1 vs C2	
iAIRS best overall result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.56
Clonal rate	4
Hyper mutation rate	4
Distance threshold	0.5
Distance threshold scalar	1.5
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	93.50%
Training Result	95.89%
Overall Result	**95.65%**
Mean Memory Antibodies	176.9

Synthetic Test 2 - C1 vs C2	
iAIRS best test result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.48
Clonal rate	4
Hyper mutation rate	4
Distance threshold	0.4
Distance threshold scalar	1
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	**96.50%**
Training Result	94.72%
Overall Result	94.90%
Mean Memory Antibodies	179.8

Table 4.10: iAIRS classification results-Synthetic Test 2

Synthetic Test 3 - Data Set Information			
Classes	MU	Cases	Dimensions
C1	2	100	2
C2	4	100	2
C3	6	100	2

Table 4.11: Data set information-Synthetic Test 3

Synthetic Test 3 - C1 vs C2 vs C3	
AIRS best overall result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.02
Clonal rate	4
Hyper mutation rate	4
Stimulation threshold	0.9973
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	87.33%
Training Result	89.93%
Overall Result	**89.67**%
Mean Memory Antibodies	241.9

Synthetic Test 3 - C1 vs C2 vs C3	
AIRS best test result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.17
Clonal rate	4
Hyper mutation rate	4
Stimulation threshold	0.9879
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	**88.33%**
Training Result	87.96%
Overall Result	88%
Mean Memory Antibodies	65.5

Table 4.12: AIRS classification results-Synthetic Test 3

Synthetic Test 3 - C1 vs C2 vs C3	
iAIRS best overall result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.02
Clonal rate	4
Hyper mutation rate	4
Distance threshold	0.2
Distance threshold scalar	0.5
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	87%
Training Result	90.67%
Overall Result	**90.30%**
Mean Memory Antibodies	207.3

Synthetic Test 3 - C1 vs C2 vs C3	
iAIRS best test result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.18
Clonal rate	4
Hyper mutation rate	4
Distance threshold	0.5
Distance threshold scalar	1.5
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	**90%**
Training Result	87.11%
Overall Result	87.40%
Mean Memory Antibodies	80.3

Table 4.13: iAIRS classification results-Synthetic Test 3

Synthetic Test 4 - Data Set Information			
Classes	MU	Cases	Dimensions
C1	2	100	2
C2	4	100	2
C3	6	100	2
C4	8	100	2

Table 4.14: Data set information-Synthetic Test 4

Synthetic Test 4 - C1 vs C2 vs C3 vs C4	
AIRS best overall result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.02
Clonal rate	4
Hyper mutation rate	4
Stimulation threshold	0.9963
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	84.75%
Training Result	88.39%
Overall Result	**88.03**%
Mean Memory Antibodies	303.4

Synthetic Test 4 - C1 vs C2 vs C3 vs C4	
AIRS best test result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.12
Clonal rate	4
Hyper mutation rate	4
Stimulation threshold	0.9922
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	**86.50%**
Training Result	86.50%
Overall Result	86.50%
Mean Memory Antibodies	101.5

Table 4.15: AIRS classification results-Synthetic Test 4

Synthetic Test 4 - C1 vs C2 vs C3 vs C4	
iAIRS best overall result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.02
Clonal rate	4
Hyper mutation rate	4
Distance threshold	0.2
Distance threshold scalar	0.5
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	85.75%
Training Result	89.28%
Overall Result	**88.92%**
Mean Memory Antibodies	293.5

Synthetic Test 4 - C1 vs C2 vs C3 vs C4	
iAIRS best test result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.14
Clonal rate	4
Hyper mutation rate	4
Distance threshold	0.2
Distance threshold scalar	0.5
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	**88.50%**
Training Result	86.19%
Overall Result	86.43%
Mean Memory Antibodies	87.8

Table 4.16: iAIRS classification results-Synthetic Test 4

Synthetic Test 5 - Data Set Information			
Classes	MU	Cases	Dimensions
C1	2	100	2
C2	4	100	2
C3	6	100	2
C4	8	100	2
C4	10	100	2

Table 4.17: Data set information-Synthetic Test 5

Synthetic Test 5 - C1 vs C2 vs C3 vs C4 vs C5	
AIRS best overall result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.01
Clonal rate	4
Hyper mutation rate	4
Stimulation threshold	0.9969
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	83.20%
Training Result	87.42%
Overall Result	**87**%
Mean Memory Antibodies	405.7

Synthetic Test 5 - C1 vs C2 vs C3 vs C4 vs C5	
AIRS best test result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.1
Clonal rate	4
Hyper mutation rate	4
Stimulation threshold	0.9907
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	**86.20%**
Training Result	84.76%
Overall Result	84.90%
Mean Memory Antibodies	115.6

Table 4.18: AIRS classification results-Synthetic Test 5

Synthetic Test 5 - C1 vs C2 vs C3 vs C4 vs C5	
iAIRS best overall result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.02
Clonal rate	4
Hyper mutation rate	4
Distance threshold	0.1
Distance threshold scalar	0.25
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	85%
Training Result	88.38%
Overall Result	**88.04%**
Mean Memory Antibodies	293.5

Synthetic Test 5 - C1 vs C2 vs C3 vs C4 vs C5	
iAIRS best test result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.1
Clonal rate	4
Hyper mutation rate	4
Distance threshold	0.3
Distance threshold scalar	0.25
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	**88.40%**
Training Result	85.31%
Overall Result	85.62%
Mean Memory Antibodies	120.6

Table 4.19: iAIRS classification results-Synthetic Test 5

4.3.2 Music Test Data Set Results

To further compare the two versions of the algorithm, their performance as music genre classifiers was tested using the music data set discussed in Section4.2.2. Once more, the same experiments that were performed on the original AIRS were performed on the alternative formulation iAIRS and the best classification results against both testing subsets and the entirety of data set for each version of the algorithm are presented. The data set has been presented in section 4.2.2, while class IDs are presented in table 4.2.

Automated music genre classification constitutes a non-trivial multi-class classification problem since boundaries between genres are extremely overlapping and fuzzy (Sotiropoulos and Tsihrintzis, 2017b). Therefore, it may serve as an ideal framework in order to assess the validity of our proposed reformulation of AIRS algorithm.

Music Test 1 - C1 vs C2	
AIRS best overall result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.56
Clonal rate	4
Hyper mutation rate	4
Stimulation threshold	0.98
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	93.50%
Training Result	95.17%
Overall Result	**95**%
Mean Memory Antibodies	152.5

Music Test 1 - C1 vs C2	
AIRS best test result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.55
Clonal rate	4
Hyper mutation rate	4
Stimulation threshold	0.98
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	**94.50%**
Training Result	94.78%
Overall Result	94.75%
Mean Memory Antibodies	155.7

Table 4.20: AIRS classification results-Music Test 1

Music Test 1 - C1 vs C2	
iAIRS best overall result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.34
Clonal rate	4
Hyper mutation rate	4
Distance threshold	0.5
Distance threshold scalar	1.5
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	93%
Training Result	96.78%
Overall Result	**96.40%**
Mean Memory Antibodies	136.9

Music Test 1 - C1 vs C2	
iAIRS best test result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.41
Clonal rate	4
Hyper mutation rate	4
Distance threshold	0.4
Distance threshold scalar	0.5
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	**95%**
Training Result	95.28%
Overall Result	95.25%
Mean Memory Antibodies	70.1

Table 4.21: iAIRS classification results-Music Test 1

Music Test 2 - C1 vs C2 vs C3	
AIRS best overall result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.35
Clonal rate	4
Hyper mutation rate	4
Stimulation threshold	0.9257
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	72%
Training Result	83.33%
Overall Result	**82.17**%
Mean Memory Antibodies	266.2

Music Test 2 - C1 vs C2 vs C3	
AIRS best test result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.39
Clonal rate	4
Hyper mutation rate	4
Stimulation threshold	0.9257
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	**74.33%**
Training Result	82.11%
Overall Result	81.33%
Mean Memory Antibodies	262.7

Table 4.22: AIRS classification results-Music Test 2

Music Test 2 - C1 vs C2 vs C3	
iAIRS best overall result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.13
Clonal rate	4
Hyper mutation rate	4
Distance threshold	0.4
Distance threshold scalar	1.5
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	73%
Training Result	87.19%
Overall Result	**85.77%**
Mean Memory Antibodies	269.4

Music Test 2 - C1 vs C2 vs C3	
iAIRS best test result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.24
Clonal rate	4
Hyper mutation rate	4
Distance threshold	0.6
Distance threshold scalar	1.5
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	**75.67%**
Training Result	85.96%
Overall Result	84.93%
Mean Memory Antibodies	246.6

Table 4.23: iAIRS classification results-Music Test 2

Music Test 3 - C1 vs C2 vs C3 vs C4	
AIRS best overall result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.32
Clonal rate	4
Hyper mutation rate	4
Stimulation threshold	0.9287
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	66.25%
Training Result	80.36%
Overall Result	**78.95**%
Mean Memory Antibodies	354.8

Music Test 3 - C1 vs C2 vs C3 vs C4	
AIRS best test result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.49
Clonal rate	4
Hyper mutation rate	4
Stimulation threshold	0.9881
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	**68.50%**
Training Result	77.33%
Overall Result	76.45%
Mean Memory Antibodies	336.1

Table 4.24: AIRS classification results-Music Test 3

Music Test 3 - C1 vs C2 vs C3 vs C4	
iAIRS best overall result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.01
Clonal rate	4
Hyper mutation rate	4
Distance threshold	0.6
Distance threshold scalar	1
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	64.75%
Training Result	82.86%
Overall Result	**81.05%**
Mean Memory Antibodies	353

Music Test 3 - C1 vs C2 vs C3 vs C4	
iAIRS best test result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.4
Clonal rate	4
Hyper mutation rate	4
Distance threshold	0.2
Distance threshold scalar	1
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	**69.50%**
Training Result	80%
Overall Result	78.95%
Mean Memory Antibodies	335.9

Table 4.25: iAIRS classification results-Music Test 3

Music Test 4 - C1 vs C2 vs C3 vs C4 vs C5	
AIRS best overall result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.34
Clonal rate	4
Hyper mutation rate	4
Stimulation threshold	0.9544
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	60.40%
Training Result	74.98%
Overall Result	**73.52**%
Mean Memory Antibodies	444.1

Music Test 4 - C1 vs C2 vs C3 vs C4 vs C5	
AIRS best test result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.41
Clonal rate	4
Hyper mutation rate	4
Stimulation threshold	0.9312
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	**61%**
Training Result	74.53%
Overall Result	73.18%
Mean Memory Antibodies	426.2

Table 4.26: AIRS classification results-Music Test 4

Music Test 4 - C1 vs C2 vs C3 vs C4 vs C5	
iAIRS best overall result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.29
Clonal rate	4
Hyper mutation rate	4
Distance threshold	0.6
Distance threshold scalar	0.5
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	60.80%
Training Result	79%
Overall Result	**77.18%**
Mean Memory Antibodies	429.1

Music Test 4 - C1 vs C2 vs C3 vs C4 vs C5	
iAIRS best test result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.29
Clonal rate	4
Hyper mutation rate	4
Distance threshold	0.2
Distance threshold scalar	0.5
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	**62.40%**
Training Result	77.02%
Overall Result	75.56%
Mean Memory Antibodies	440.6

Table 4.27: iAIRS classification results-Music Test 4

4.3.3 WDBC Test Data Set Results

The two versions of the algorithm were also tested as classifiers that discriminate benign from malignant breast lumps. For that purpose, the WBCD (Wisconsin Breast Cancer Diagnosis) dataset is employed.

This data set is widely utilized for this kind of application because it has a large number of instances and is virtually noise-free. However, the two classes are slightly unbalanced with the 'bening' class having slightly more data instances than the 'malign' class. The WDBC data set has been previously discussed in section 4.2.2.

WDBC Test - Bening vs Malign	
AIRS best overall result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.15
Clonal rate	4
Hyper mutation rate	4
Stimulation threshold	0.9680
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	97.01%
Training Result	97.95%
Overall Result	**97.86**%
Mean Memory Antibodies	469.3

WDBC Test - Bening vs Malign	
AIRS best test result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.28
Clonal rate	4
Hyper mutation rate	4
Stimulation threshold	0.9570
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	**97.18%**
Training Result	97.27%
Overall Result	97.26%
Mean Memory Antibodies	274.1

Table 4.28: AIRS classification results-WDBC Test

WDBC Test - Bening vs Malign	
iAIRS best overall result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.14
Clonal rate	4
Hyper mutation rate	4
Distance threshold	0.8
Distance threshold scalar	1
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	96.89%
Training Result	98.32%
Overall Result	**98.18%**
Mean Memory Antibodies	285.4

WDBC Test - Bening vs Malign	
iAIRS best test result	Value
Input Values	
Training data percentage	0.9
Affinity threshold scalar	0.33
Clonal rate	4
Hyper mutation rate	4
Distance threshold	0.4
Distance threshold scalar	0.5
Total resources	20
Nearest Neighbors	5
Classification Results	
Test Result	**97.53%**
Training Result	96.52%
Overall Result	96.63%
Mean Memory Antibodies	92.5

Table 4.29: iAIRS classification results-WDBC Test

4.4 Discussion and Comparative Analysis

In this section, the results presented in the previous section 4.3 are thoroughly discussed and evaluated. The focus of this discussion will be on three important features of the AIRS algorithms: classification accuracy, data reduction and algorithmic efficiency.

4.4.1 Classification accuracy

One of the most important features of AIRS algorithms is its competitive classification accuracy. Therefore, in this section, the best configuration classification results for both the testing sets and the entirety of the data set presented in section 4.3 are compared in order to assess any significant difference between the two version of the AIRS algorithm regarding classification accuracy. Table 4.30 presents the best overall accuracies, achieved by both versions of AIRS on all the benchmark data sets, while table 4.31 presents the best testing accuracies.

Overall Data-Classification Results Comparison			
Test ID	AIRS	iAIRS	+/-
Synthetic Test 1	92.20%	92.70%	+0.50%
Synthetic Test 2	93.15%	95.65%	+2.50%
Synthetic Test 3	89.67%	90.30%	+0.63%
Synthetic Test 4	88.03%	88.92%	+0.89%
Synthetic Test 5	87%	88.04%	+1.04%
Music Test 1	95%	96.40%	+1.40%
Music Test 2	82.17%	85.77%	+3.60%
Music Test 3	78.95%	81.05%	+2.10%
Music Test 4	73.52%	77.18%	+3.66%
WDBC	97.86%	98.18%	+0.32%

Table 4.30: Overall Data-Classification Results Comparison

Testing Set-Classification Results Comparison			
Test ID	AIRS	iAIRS	+/-
Synthetic Test 1	92%	92.50%	+0.50%
Synthetic Test 2	94%	96.50%	+2.50%
Synthetic Test 3	88.33%	90%	+1.67%
Synthetic Test 4	86.50%	88.50%	+2%
Synthetic Test 5	86.20%	88.40%	+2.20%
Music Test 1	94.50%	95%	+0.50%
Music Test 2	74.33%	75.67%	+1.33%
Music Test 3	68.50%	69.50%	+1%
Music Test 4	61%	62.40%	+1.40%
WDBC	97.18%	97.53%	+0.35%

Table 4.31: Testing Set-Classification Results Comparison

It can be noted that the alternative version iAIRS had better classification results than the original AIRS in all testing scenarios, with both versions optimally configurated. The gains appear to be greater in multi-class scenarios and the more difficult classification problems with fuzzy and overlapping boundaries. In some cases, these differences appear to be greater than 2%, therefore noting that the changes introduced and discussed in section 3.3 have indeed improved the classification accuracy of the algorithm.

However, the results presented so far were achieved by finding the configuration that led to the best accuracy result for each data set through grid searching. Therefore, in order to assess the consistency of iAIRS classification ability in comparison to AIRS, the average and the standard deviation of the fifty best testing set results as emerged from the grid searching testing process

are presented in table 4.32. To allow for comparison, all hyper-parameters except Stimulation Threshold and Affinity Threshold Scalar were set to the same fixed values.

Best Fifty Testing Set Results			
Test ID	AIRS	iAIRS	+/-
Synthetic Test 1	90.61% (0.48%)	90.87% (0.46%)	+0.26% (-0.02%)
Synthetic Test 2	94% (2*10^{-14}%)	96% (8*10^{-14}%)	+2% (+6*10^{-14}%)
Synthetic Test 3	86.85% (0.7%)	87.25% (0.62%)	+0.40% (-0.08 %)
Synthetic Test 4	84.75% (0.64%)	85.56% (1.03%)	+0.81% (+0.37 %)
Synthetic Test 5	83.40% (1.02%)	84.16% (1.54%)	+0.76% (+0.52%)
Music Test 1	93.26% (0.35%)	93.34% (0.24%)	+0.08% (-0.11%)
Music Test 2	73.19% (0.81%)	73.32% (0.75%)	+0.13% (-0.06 %)
Music Test 3	67.57% (0.38%)	67.69% (0.61%)	+0.12% (+0.13%)
Music Test 4	60.59% (0.2%)	61.28% (0.59%)	+0.69% (+0.39%)
WDBC	96.95% (0.14%)	96.88% (0.19%)	-0.07% (+0.05%)

Table 4.32: Best Fifty Testing Set Results

Here it can be observed that, the iAIRS version holds a slight advantage in all the testing scenarios except the WDBC data set regarding the average classification accuracy, while standard deviation appears marginally higher in iAIRS.

To delve deeper, investigations were undertaken to determine what affect altering both the Stimulation Threshold and the Affinity Threshold Scalar have on the classification accuracy of the two algorithms. Table 4.33 and table 4.34 show average testing classification results on a synthetic data set (Table 4.5) and WDBC data set respectively, achieved through tuning Affinity Threshold Scalar in the range [0.01,0.6] with the step equal to 0.01, while all other-hyper-parameters values are fixed. To maintain comparability, Stimulation Threshold is calculated as the average of the stimulation thresholds of each training antigen in iAIRS as illustrated in 3.7.

Synthetic Test 1-Accuracy Sensitivity Comparison			
Stimulation Threshold / Distance Threshold	AIRS	iAIRS(DTS=1)	iAIRS(DTS=1.5)
ST=0.9958/DT=0.05	88.84% (1.53%)	**89.27% (1.47%)**	89.09% (1.66%)
ST=0.9917/DT=0.1	88.42% (1.84%)	89.14% (1.19%)	**89.18% (1.55%)**
ST=0.9833/DT=0.2	87.53% (2.31%)	88.62% (1.51%)	**88.63% (1.44%)**
ST=0.9750/DT=0.3	87.55% (1.64%)	**88.71% (1.19%)**	88.28% (1.34%)
ST=0.9666/DT=0.4	87.88% (1.55%)	**88.80% (1.51%)**	88.19% (1.57%)
ST=0.9583/DT=0.5	88.02% (1.84%)	**88.71% (1.46%)**	88.28% (1.68%)
ST=0.9499/DT=0.6	87.91% (1.71%)	88.32% (1.94%)	**88.62% (1.58%)**

Table 4.33: Synthetic Test 1-Accuracy Sensitivity Comparison

WDBC Test-Accuracy Sensitivity Comparison				
Stimulation Threshold / Distance Threshold	AIRS	iAIRS(DTS=0.5)	iAIRS(DTS=1)	iAIRS(DTS=1.5)
ST=0.9880/DT=0.05	95.969% (0.93%)	95.89% (0.91%)	**95.972% (0.85%)**	95.90% (0.72%)
ST=0.9780/DT=0.1	95.87% (0.75%)	95.81% (0.90%)	95.86% (0.74%)	**95.98% (0.80%)**
ST=0.9680/DT=0.2	95.96% (0.76%)	95.84% (0.87%)	95.85% (0.76%)	**96.09% (0.65%)**
ST=0.9570/DT=0.3	96.06% (0.83%)	95.93% (0.8%)	95.94% (0.69%)	**96.11% (0.65%)**
ST=0.9466/DT=0.4	95.85% (0.81%)	95.80% (1.11%)	95.98% (0.86%)	**96.15% (0.48%)**
ST=0.9360/DT=0.5	95.75% (0.79%)	95.59% (1.13%)	95.85% (0.83%)	**96.14% (0.54%)**
ST=0.9250/DT=0.6	95.61% (0.68%)	95.31% (1.49%)	95.64% (0.83%)	**95.89% (0.74%)**

Table 4.34: WDBC Test-Accuracy Sensitivity Comparison

According to the results shown in tables 4.33 and 4.34, it appears that as the stopping criterion becomes more relaxed, the differences in classification accuracy between the two algorithms grow wider in favor of iAIRS. This trend is further explored in regard to the data reduction capabilities of AIRS in the following section.

4.4.2 Data Reduction

In the previous section, the main focus was on the classification accuracy of the two algorithms under various circumstances. It can be seen that the changes introduced to AIRS offer a small, but in some cases substantial, improvement to classification accuracy. However, authors (Watkins and Boggess, 2002) argue that aside from high classification accuracy another significant feature of the AIRS algorithm is its ability to reduce the number of data instances needed to characterize a given class of data from the original training data to the evolved set of memory cells. Therefore, it is essential to focus our comparison to the data reduction capabilities of the two algorithms and how they affect the classification accuracy tests presented in the previous section.

To begin with, table 4.35 and 4.36 show the size of the evolved set of memory cells used for classification on the best configuration results presented in tables 4.30 and 4.31, respectively.

Overall Data-Data Reduction Comparison			
Test ID	AIRS	iAIRS	+/-
Synthetic Test 1	155.6	128.1	-27.5(-17.67%)
Synthetic Test 2	180	176.9	-3.1(-1.70%)
Synthetic Test 3	241.9	207.3	-34.6(-14.30%)
Synthetic Test 4	303.4	293.5	-9.9(-3.20%)
Synthetic Test 5	405.7	369.5	-36.2(-8.92%)
Music Test 1	152.5	136.9	-15.6(-10.22%)
Music Test 2	266.2	269.4	+3.2(+1.20%)
Music Test 3	354.8	353	-1.8(-0.51%)
Music Test 4	444.1	429.1	-15(-3.34%)
WDBC	469.3	285.4	-183.9(-39.18%)

Table 4.35: Overall Data-Data Reduction Comparison

Testing Set-Data Reduction Comparison			
Test ID	AIRS	iAIRS	+/-
Synthetic Test 1	44.9	38.6	-27.5(-14.03%)
Synthetic Test 2	180	179.8	-0.2(-0.11%)
Synthetic Test 3	65.5	80.3	+14.8(+22.59%)
Synthetic Test 4	101.5	87.8	-13.7(-13.49%)
Synthetic Test 5	115.6	120.6	+5(+4.32%)
Music Test 1	155.7	70.1	-84.9(-54.53%)
Music Test 2	262.7	246.6	-16.1(-6.12%)
Music Test 3	336.1	335.9	-0.2(-0.06%)
Music Test 4	426.2	440.6	+14.4(+3.26%)
WDBC	274.1	92.5	-181.6(-66.25%)

Table 4.36: Testing Set-Data Reduction Comparison

As it can been seen, in the best configuration, the iAIRS version exhibits greater data reduction as well as slightly better classification results for most of the data sets tested against. Hence, iAIRS seems to achieve better results than the original AIRS in terms of efficiency.

In the previous section, table 4.32 presented the best fifty testing set results in terms of classification accuracy for each algorithm, where the iAIRS version tended to have slightly higher

classification results than original AIRS. Table 4.37 shows the average number of memory cells and standard deviation for the results presented previously in table 4.32.

Best 50 Testing Set Results-Data Reduction			
Test ID	AIRS	iAIRS	+/-
Synthetic Test 1	61.28 (42.83)	56.11 (49.44)	-5.17 (-8.44%)
Synthetic Test 2	180(0)	179.99(0.0424)	-0.01(-0.005%)
Synthetic Test 3	87.25(62.66)	97.67(54.36)	+9.60 (+11%)
Synthetic Test 4	122.06 (70.22)	118.61 (72.44)	-3.45 (-2.82%)
Synthetic Test 5	161.53 (97.15)	135.56 (86.27)	-25.97 (-16.08%)
Music Test 1	176.62 (7.04)	174.08 (10.59)	-2.54 (-1.43%)
Music Test 2	268.60(4.02)	255.75((16.79)	-12.85 (-4.48%)
Music Test 3	352.28 (9.26)	346.42 (19.73)	-5.86 (-1.66%)
Music Test 4	447.05 (3.1)	435.03 (19.25)	-12.02 (-2.67%)
WDBC	393.65 (123.52)	298.40 (134.35)	-95.25 (-24.19%)

Table 4.37: Best 50 Testing Set Results-Data Reduction

These results show that iAIRS tends to exhibit consistently greater data reduction than AIRS, while also slightly improving accuracy. This indicates that the revisions made to AIRS allow for greater efficiency in general.

Additionally, in the previous section, experimental results regarding the affect of altering the stopping criterion and the Affinity Threshold Scalar, which is the hyper-parameter that provides a cut-off value for memory cell replacement and therefore significantly affects the size of the evolved set of memory cell, on classification accuracy were presented. In this section, experimental results, in correspondence to those presented in table 4.34, regarding the affect of altering the stopping criterion and the Affinity Threshold Scalar are presented but focusing on the number of memory cells instead of classification accuracy. The experimental tests were conducted on the WDBC data set.

WDBC Test-Data Reduction Sensitivity Comparison				
Stimulation Threshold / Distance Threshold	AIRS	iAIRS(DTS=0.5)	iAIRS(DTS=1)	iAIRS(DTS=1.5)
ST=0.9880/DT=0.05	318.58 (164.50)	303.89 (169.45)	310.32 (167.62)	318.31 (165.48)
ST=0.9780/DT=0.1	309.07 (162.98)	284.5 (171.96)	297.47 (169.35)	314.43 (162.52)
ST=0.9680/DT=0.2	293.5 (159.03)	245.72 (169.79)	273.01 (169.47)	307.64 (164.09)
ST=0.9570/DT=0.3	268.72 (147.87)	208.6 (157.53)	250.89 (165.65)	302.71 (161.43)
ST=0.9466/DT=0.4	238.34 (129.17)	175.97 (138.53)	230.70 (158.74)	298.86 (158.13)
ST=0.9360/DT=0.5	205.57 (106.69)	146.59 (113.635)	211.82 (150.92)	295.04 (153.51)
ST=0.9250/DT=0.6	172.95 (83.96)	121.72 (86.36)	196.22 (141.75)	290 (149.32)

Table 4.38: WDBC Test-Data Reduction Sensitivity Comparison

Some interesting deductions can be drawn from table 4.38 regarding the data reduction and classification capabilities of the proposed iAIRS algorithm and the underlying connection between the two. At first, it is clear that as the stopping criterion becomes more relaxed the number of evolved memory cells decreases for both algorithm. Furthermore, it can be seen that as the Distance Threshold Scalar increases so does the size of the set of evolved memory cells as well. On the other hand, as it was mentioned in the previous section, it also appears that as the stopping criterion becomes more relaxed, while AIRS classification accuracy is diminishing, iAIRS can still achieve competitive classification results.

These conclusions seem perplexing at first glance but they merely point to the different nature of the two formulations of the AIRS algorithm. If we take a closer look to the results of table 4.38, it can be seen that, in many cases, iAIRS exhibits greater data reduction while achieving similar classification results to AIRS. In the following figures, the effect of altering the Affinity Scalar

Threshold is illustrated. There, it can be seen that while AIRS might achieve better results when a larger number of memory cells is involved, iAIRS performs better while employing fewer number of cells. This is also demonstrated in figure 4.7 where the best fifty test results of each version on the WBDC data set are shown.

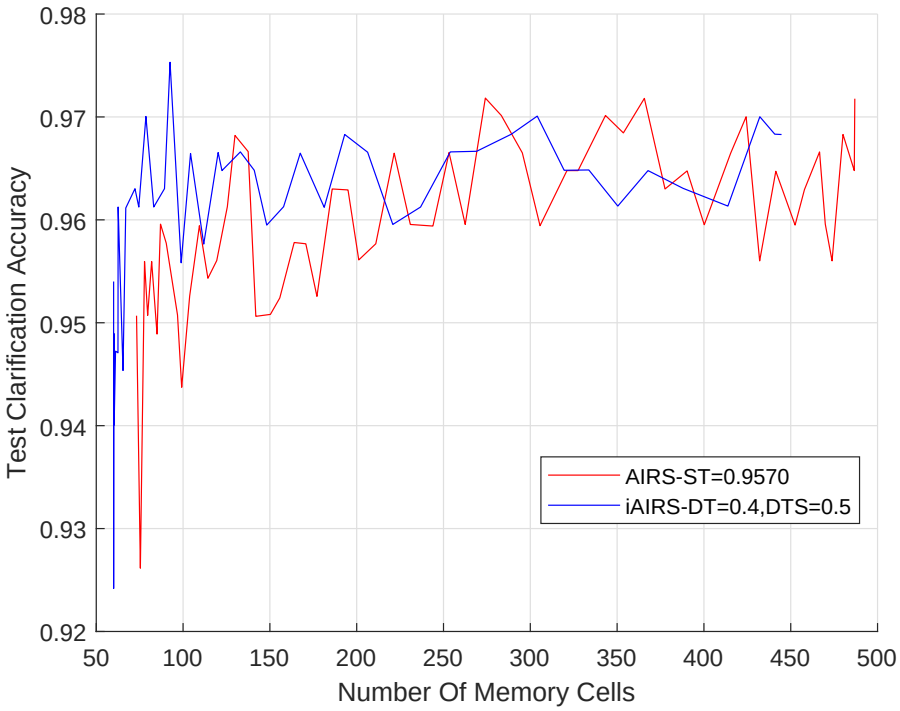

Figure 4.3: Effect of altering the Affinity Scalar Threshold-WDBC Data Set-Example 1

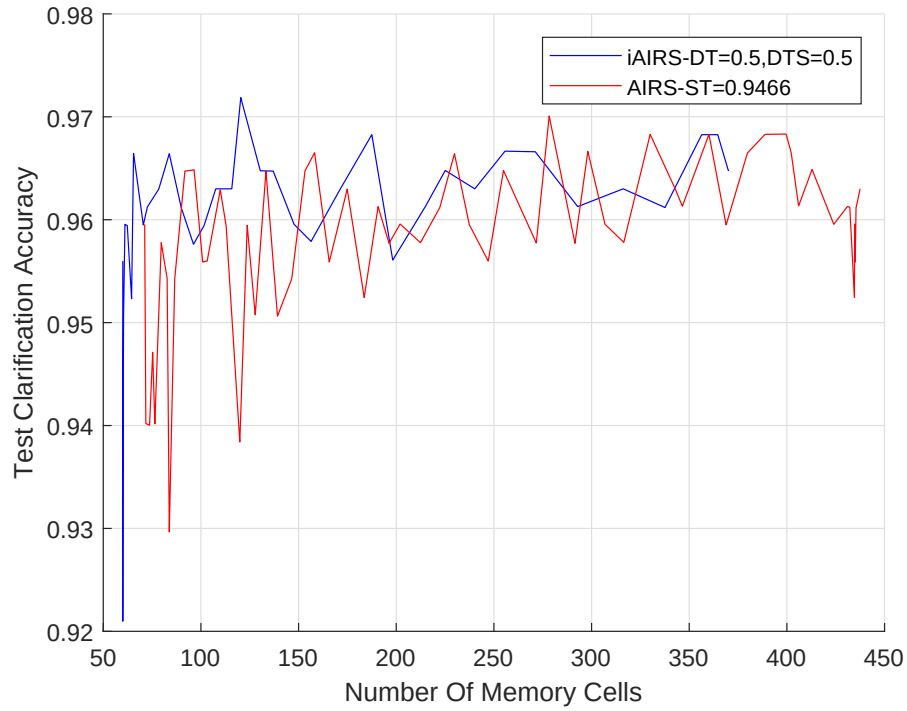

Figure 4.4: Effect of altering the Affinity Scalar Threshold-WDBC Data Set-Example 2

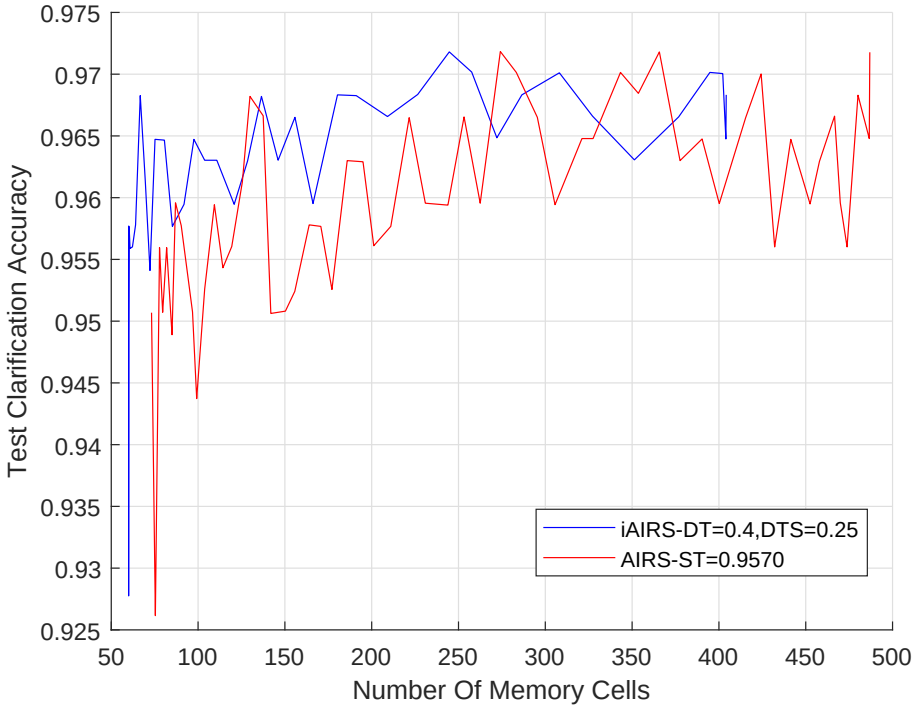

Figure 4.5: Effect of altering the Affinity Scalar Threshold-WDBC Data Set-Example 3

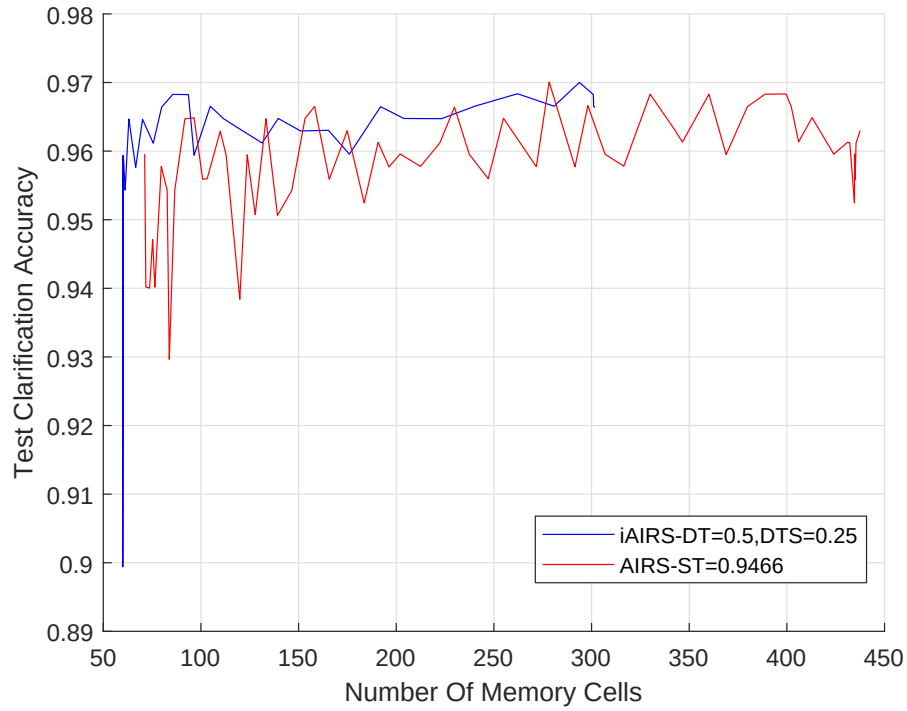

Figure 4.6: Effect of altering the Affinity Scalar Threshold-WDBC Data Set-Example 4

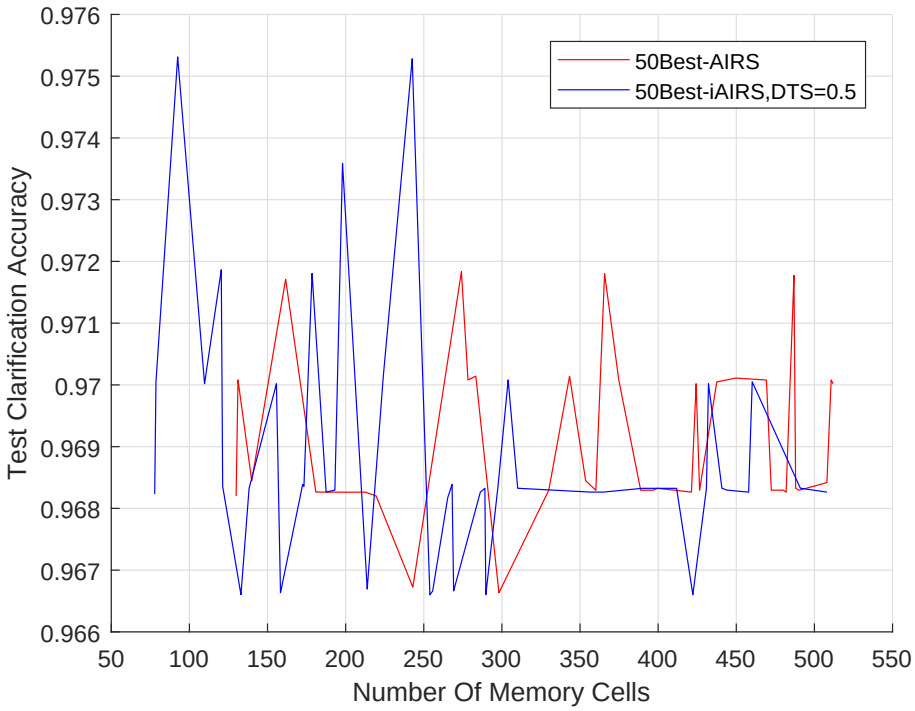

Figure 4.7: Best 50 Test Results-WDBC Data Set

4.4.3 Distance Threshold and Distance Threshold Scalar

At this point, a more thorough experimental evaluation of the two hyper-parameters, Distance Threshold and Distance Threshold Scalar, introduced to AIRS as described in section 3.3.1, is essential to the better understanding of the alternative machine learning paradigm presented in this thesis, iAIRS. In this section, the focus will be on the affect of the modifications made on the stopping criterion of the original algorithm and how the new concept of the stopping criterion, incorporated in iAIRS, really works.

Previously, in presenting test results about the classification accuracy and the data reduction capabilities of iAIRS, some patterns regarding the affect of the Distance Threshold (DT) and the Distance Threshold Scalar on the classification accuracy and the data reduction capability of iAIRS emerged. In order to properly address this matter, more experimental tests were conducted on the WDBC data set. Table 4.39 shows testing set results of both versions of AIRS on the WDBC data set for various levels of the Distance Threshold (iAIRS) and the average Stimulation Threshold (AIRS). DTS was set to zero, so as to study the affect of Distance Threshold more clearly, while ATS was set to 0.01. All other hyper-parameter values were set as shown in table 4.1.

WDBC-Distance Threshold Sensitivity Results		
Distance Threshold / Average Stimulation Threshold	AIRS Accuracy (Number of Cells)	iAIRS Accuracy (Number of Cells)
DT=0.1/ST=0.9900	96.65% (512.1)	96.65% (512.1)
DT=0.2/ST=0.9790	96.13% (512.1)	96.30% (508.7)
DT=0.3/ST=0.9680	96.48% (510.1)	96.13% (464.7)
DT=0.4/ST=0.9570	97.18% (486.9)	96.65% (361.1)
DT=0.5/ST=0.9460	96.30% (426.9)	96.48% (245.9)
DT=0.6/ST=0.9460	95.24% (373)	96.48% (165.5)
DT=0.7/ST=0.9250	95.96% (295.9)	95.96% (108)
DT=0.8/ST=0.9150	95.23% (243.4)	94.36% (71.8)

Table 4.39: Distance Threshold Sensitivity Results

It can be observed that the modified stopping criterion of iAIRS leads to greater data reduction than the original version, especially as the stopping criterion becomes more relaxed, but despite this, classification accuracy, for the most part, is not significantly sacrificed for the greater data reduction. This was expected as result of the nature of the modified stopping criterion, where training for antigens that are not very close to antigens of different class stops at an early stage allowing for greater generalization while training for antigens positioned close to antigenic patterns of different class is extensive, providing greater specification in those cases. As a result, the former group of antigens produce less evolved candidate memory cells than those produced by the latter group, which leads to a larger number of those less evolved candidate memory cells not getting introduced to the final set of memory cells, as the possibility that a more stimulated memory cell already exists is increased for those less evolved candidate cells (see this).

Following, the same test was conducted again but this time the focus was on the affect of altering the value of the Distance Threshold Scalar. Table 4.40 shows the affect of altering DTS when the Distance Threshold is equal to 0.5, while Table 4.41 presents the best configuration test results for each level of the stopping criterion and the optimal DTS value.

WDBC-Distance Threshold Scalar Sensitivity Results			
Distance Threshold Scalar	Testing Accuracy	Training Accuracy	Number of Cells
0	96.48%	97.54%	245.9
0.1	96.47%	97.81%	268.4
0.2	96.47%	97.97%	294.6
0.3	96.65%	98.01%	316.6
0.4	95.95%	97.93%	340.1
0.5	96.48%	98.07%	370.4
0.6	96.65%	97.95%	392
0.7	96.66%	97.79%	418.8
0.8	96.83%	98.05%	440.5
0.9	97.18%	97.91%	459.2
1	96.13%	97.93%	472.9
1.1	96.12%	98.11%	482.1
1.2	96.30%	97.85%	490.9
1.3	96.83%	97.79%	495.6
1.4	96.12%	97.87%	499.2
1.5	96.65%	97.93%	503.2
1.6	96.83%	97.83%	502.4
1.7	96.65%	98.01%	501.4
1.8	96.65%	97.91%	496.4
1.9	96.65%	98.01%	488.2

Table 4.40: Distance Threshold Scalar Sensitivity Results-DT=0.5

WDBC-Optimal Distance Threshold Scalar-Test Results			
Distance Threshold / Average Stimulation Threshold	Optimal DTS Value	AIRS Accuracy (Number of Cells)	iAIRS Accuracy (Number of Cells)
DT=0.1/ST=0.9900	1.9	96.65% (512.1)	97.00% (512.1)
DT=0.2/ST=0.9790	1.9	96.13% (512.1)	97.00% (512.1)
DT=0.3/ST=0.9680	0.1	96.48% (510.1)	97.01% (509.4)
DT=0.4/ST=0.9570	0.3	97.18% (486.9)	96.83% (414.2)
DT=0.5/ST=0.9460	0.9	96.30% (426.9)	97.18% (459.2)
DT=0.6/ST=0.9460	0.4	95.24% (373)	96.83% (263.2)
DT=0.7/ST=0.9250	0.2	95.96% (295.9)	96.83% (143.8)
DT=0.8/ST=0.9150	0.4	95.23% (243.4)	96.83% (142.6)

Table 4.41: Optimal Distance Threshold Scalar-Test Results

A few observations about the Distance Threshold Scalar can be drawn from these results. To begin with, it can be seen that higher DTS values associate with a larger number of memory cells up till a point when for very high DTS values (over 1.5) this trend reverses and the number of memory cells decreases again. The reason behind this pattern is that for higher values of DTS the space that candidate cells should be in order to satisfy the stopping criterion is smaller (see figure 3.7), meaning the stopping criterion becomes stricter and the evolution of antibodies becomes more focused and individualized for each antigenic pattern and thus the possibility of a memory cell, more stimulated to the currently presented antigen than the candidate cell, already existing is smaller.

Yet, for very high DTS values the possibility of a candidate memory cell getting introduced to the set becomes slightly smaller again, as in this case, the stimulation of the candidate cell to the currently presented antigenic pattern may be quite low, such that the case of a more stimulated memory cell already existing becomes quite possible again.

However, as table 4.41 shows, with the optimal combination of Distance Threshold and Distance Threshold Scalar, iAIRS can maximize classification accuracy while achieving greater data reduction at the same time when compared to AIRS. As it seems, the contrasting natures of Distance Threshold and Distance Threshold Scalar could be accordingly combined, optimally adjusting the modified stopping criterion in iAIRS and achieve competitive results of high classification accuracy and great data reduction.

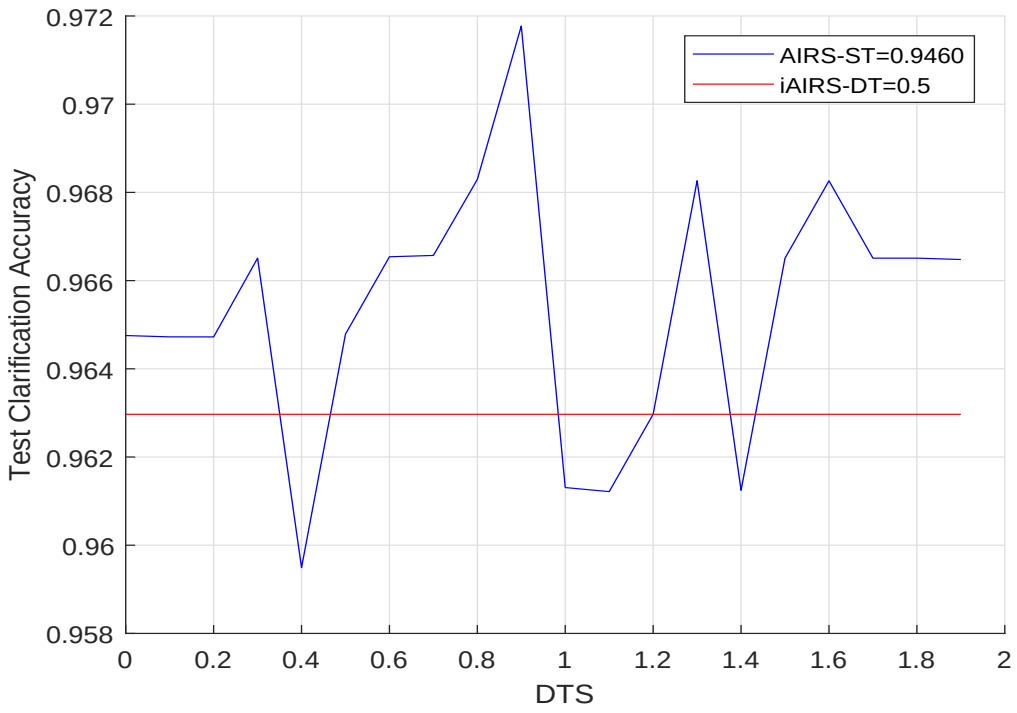

Figure 4.8: Effect of Altering DTS-Classification Accuracy

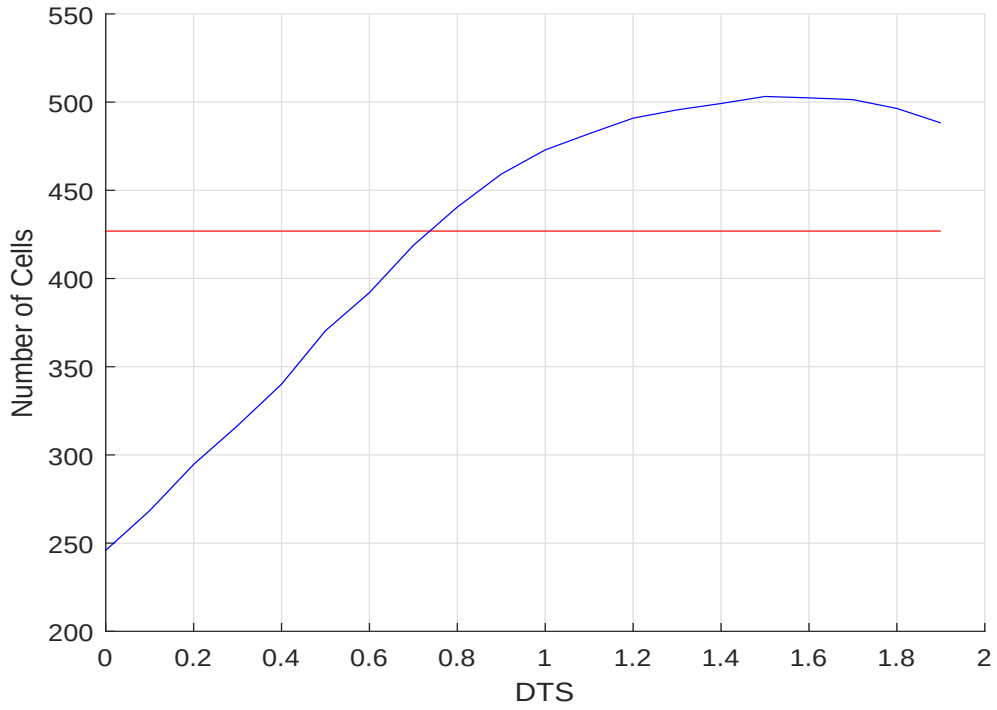

Figure 4.9: Effect of Altering DTS-Number of Cells

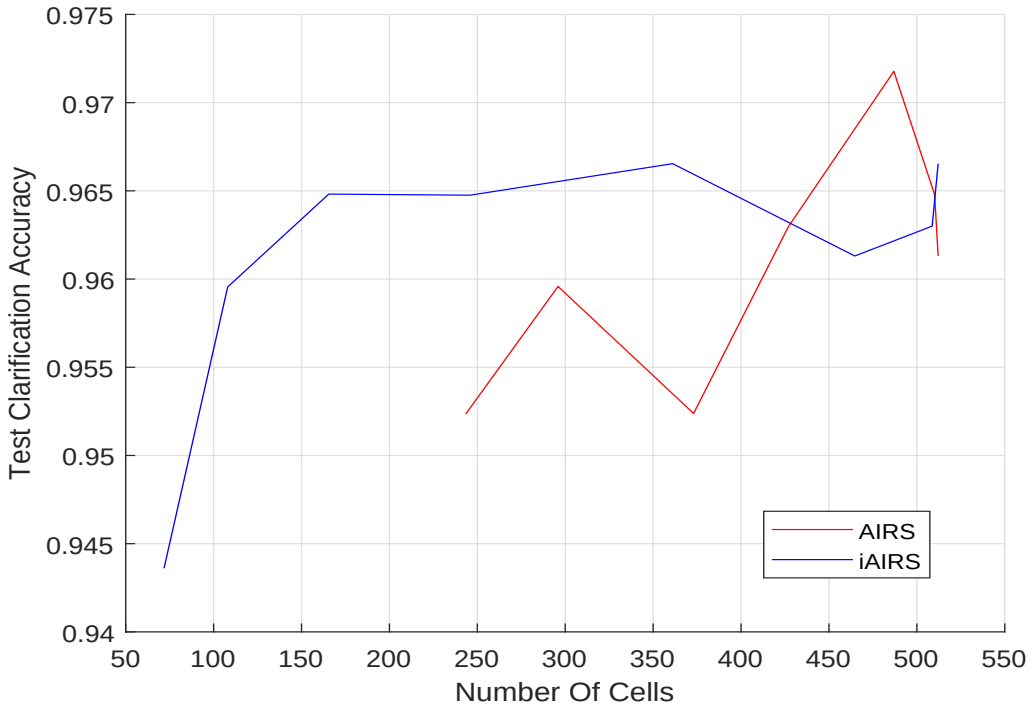

Figure 4.10: Effect of Altering DT-DTS=0

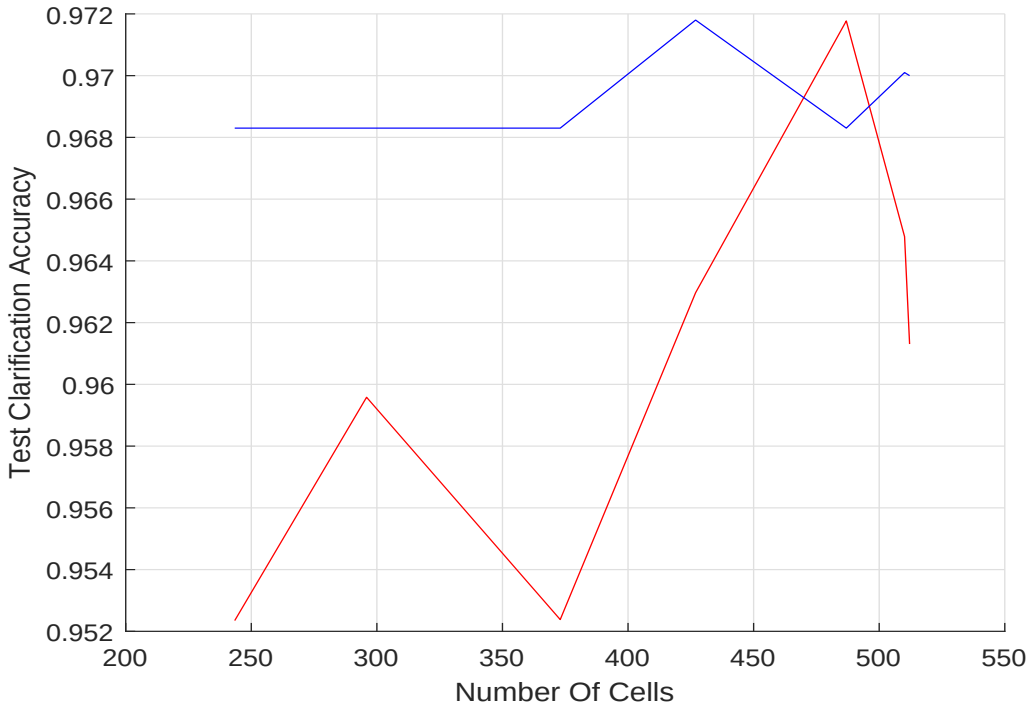

Figure 4.11: Effect of Altering DT-Optimal DTS

4.4.4 Algorithmic Efficiency

While the focus of this attempt to reformulate the AIRS algorithm has not been on the time complexity of AIRS, observations concerning the efficiency of the mutation process have been made, motivating the changes described in section 3.3.2. The objective of adopting these modifications was to decrease the time complexity of the AIRS classification system, especially when dealing with high-dimensional data sets for which the execution time of the system often increases beyond a manageable level.

In this section the impact of the modification made on the mutation process, as detailed in section 3.3.2, is assessed. Table 4.42 shows the execution time of the two version of AIRS for various data sets. The experiments were conducted under the same conditions and configurations for both versions.

Execution Time Results					
Test ID	Dimensions	Number of Instances	AIRS	iAIRS	+
Synthetic Test 1	2	200	2.219 sec	2.338 sec	+5.36 %
Music Test 1	30	200	19.970 sec	3.307 sec	-83.44 %
WDBC Test	30	569	32.939 sec	7.476 sec	-77.30 %
Synthetic Test 2	50	200	40.328 sec	4.078 sec	-89.88 %

Table 4.42: Execution Time Results Comparison

It is evident that the modifications made to the mutation mechanisms of AIRS, now allowing for a much more focused and efficient maturation of the memory cells, resulted in a significant improvement in the time complexity of the algorithm, particularly for high-dimensional data sets. Additionally, the difference on execution time between the two versions increases as dimensionality increases.

Further tests using very high-dimensional synthetic data sets were conducted for both algorithms. However, the testing of original AIRS with very high-dimensional data sets was practi-

cally infeasible in terms of time complexity for a high stimulation threshold (over 0.8). Table 4.43 presents the execution time of iAIRS on several very-high dimensional data sets for an average stimulation threshold equal to 0.99.

Execution Time Results-Very High-Dimensional Data		
Dimensions	Number of Instances	iAIRS Execution Time
1000	200	11.692 sec
2000	200	18.850 sec
3000	200	24.892 sec
4000	200	32.119 sec
5000	200	38.646 sec

Table 4.43: Execution Time Results-Very High-Dimensional Data

Seemingly, the continuous increasing of data dimensions leads to only marginal increasing of the execution time of iAIRS. Most importantly, the classification capabilities of the algorithm remain unaffected. This is a significant advantage of iAIRS, especially considering that overall the classification accuracy of the system was not sacrificed due to the changed nature of the mutation process, as in most cases performed better than AIRS. With very high-dimensional classification problems being ubiquitous in numerous applications, such a significant reduction in the time complexity of the algorithm when dealing with data of this sort is certainly beneficial, as the classification is usually time-consuming when the training data set is large and high-dimensional.

For example, automatic text categorization is a problem that pose the challenge of efficiently processing high-dimensional data while not affecting the quality of performance and classifying texts fast and accurately is essential (Burges, 2010). To tackle this issue, dimension reduction methods are often applied but these methods usually work only to some extent (Kim et al., 2005). Therefore, the fact that our proposed interpretation AIRS can deal faster with high-dimensional data while preserving the quality of classification accuracy is highly significant.

5 A Proposed Weighted Decision Process

Although this attempt to reformulate the AIRS algorithm is focusing mainly on the learning process of the algorithm, before the conclusion of this work, we would like to briefly discuss the decision process of the algorithm.

Following training, to classify data AIRS takes an unweighted majority vote amongst the k most stimulated memory cells and as such one could describe AIRS as an unweighted k-nearest neighbour classifier. This creates coarse decision boundaries and ignores a lot of the available information.

This is especially the case for multi-class classification where the occurrence of ties, where there is no clear winner in the majority voting, increases. In these cases, the lack of additional information in the decision process is particularly apparent and crucial. Previous investigations in literature (Marwah and Boggess, 2002) suggest that some form of weighting and handling of these ties would likely be beneficial.

To explore this idea, we modified the decision process to now take a weighted majority vote instead of unweighted one. The proposed modification is that instead of a simple vote, the memory cell stimulation level to test data will be used for the majority voting. This not only would handle ties but also take into account additional information in classifying the data.

Table 5.1 demonstrates the classification accuracy difference between the unweighted and the weighted majority vote for the best test results on the music data set shown in table 4.31. 5-NN is utilized for classification in both versions. Figure 5.1 depicts the affect of altering k-value to the classification accuracy.

Desicion Process-Music Data Set			
Test ID	Classes	Unweighted	Weighted
Music Test 1	2	95 %	95 %
Music Test 2	3	75.67 %	76 %
Music Test 3	4	69.50 %	69.75 %
Music Test 4	5	62.40 %	63 %

Table 5.1: Desicion Process-Accuracy Comparison

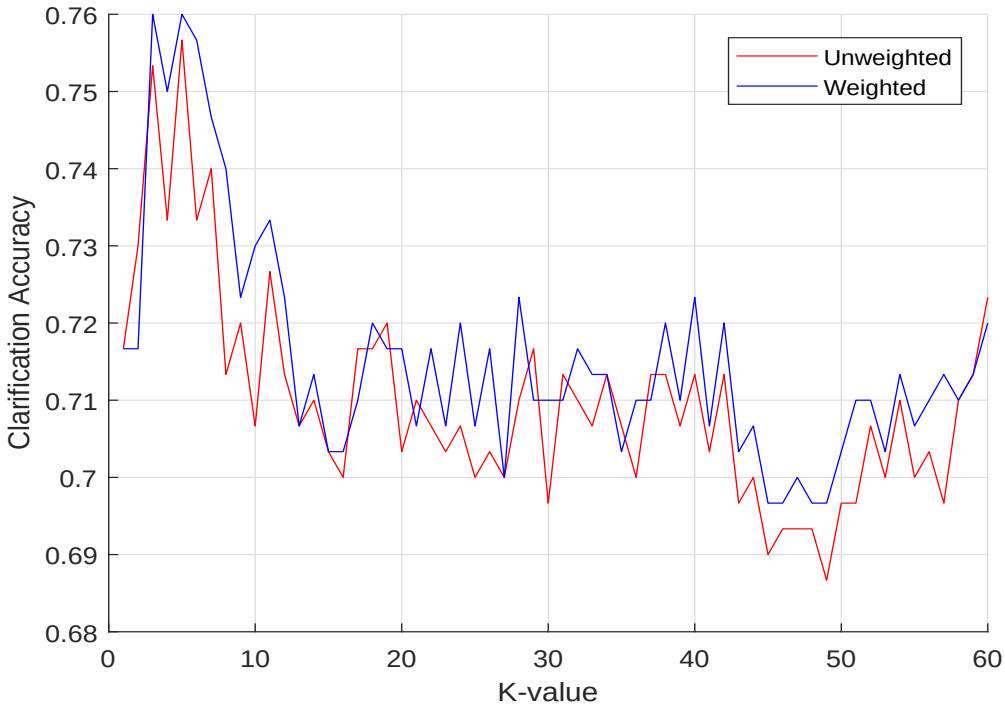

Figure 5.1: Effect of Altering K-Value-Music Test 3

It is apparent that the weighted decision process has beneficial effects on creating finer decision boundaries and handling ties, resulting in better classification accuracy regarding multi-class classification.

6 Conclusion

6.1 Summary

This thesis has presented an alternative implementation of the AIRS algorithm based on observations of the AIRS learning algorithm and other algorithms in the field of artificial immune systems. As this work focuses heavily on reformulating the AIRS algorithm, naturally follows closely from the work of Watkins et al. (2004) who originally proposed the AIRS algorithm.

Additionally, inspiration to this work was drawn from the Negative Selection algorithm proposed by Forrest et al. (1994), as one key motivation for this work was the incorporation of a censoring concept to the evolution of memory cells in AIRS.

The modifications proposed in this work include the introduction of a variant stimulation threshold and a more individualized stopping criterion adaptable to the special circumstances surrounding each presented antigenic pattern. In addition, a more directed and focused exploration of the search space was applied regarding the mutation process.

This thesis has provided a detailed discussion and presented the fundamental mechanisms of a new formulation of the AIRS algorithm, named iAIRS. Experimental results of both versions of the AIRS algorithm on synthetic and real-world data sets were presented in order to highlight the differences in classification accuracy, data reduction capabilities and algorithmic efficiency as well as evaluate the impact of the introduced modifications on the performance of the AIRS classifier.

Furthermore, we briefly discussed the decision process of AIRS and provide empirical benchmarks that suggest that a weighted majority voting would be significantly beneficial especially for multi-class classification.

Our proposed alternative formulation, iAIRS, demonstrated, in most cases, improved performance to the original AIRS algorithm in terms of classification accuracy and data reduction capability on the data sets tested for this work, thus showing a more efficient classification performance. Moreover, iAIRS managed to reduce the execution time for very high-dimensional data sets to manageable levels where the training of the original AIRS algorithm was practically infeasible.

6.2 Future Work

The exploration of an valid alternative reformulation of the AIRS algorithm is by no means concluded by the work presented here. Merely, this work pointed to the beneficial effect of some modifications being undertaken, addressing some issues of the AIRS algorithm.

However, there is still room for more modification. Some possible areas of modification could be the memory cell replacement process. Previous investigations (McEwan and Hart, 2009) have shown that the density of the set of evolved memory cells does not reflect the density of data, hindering the compression capabilities of AIRS. It have been suggested that the problem is the inflicting of a fixed threshold (Affinity Threshold) when deciding whether a candidate cell should replace the pre-existing one. It may then be possible that the replacement of Affinity Threshold for a variant threshold to better control the granularity of density representation would be of benefit.

Another area of future investigation could be on the effect of using different stimulation and affinity functions. As it stands, AIRS relies heavily on Euclidean distance as a metric for stimulation and affinity which may not be suitable for some type of data sets such as binary valued or discrete data sets. Other ambiguous processes to be investigated include the possible negative effect of the min-max attribute normalisation process at initialisation.

In addition to investigation on further modifications to the AIRS algorithm, theoretical insight into why AIRS performs as it does should clear the way for more focused and sophisticated immunological contributions to the algorithm, addressing the concerns about its functionality and further establish the validity of the algorithm.

Bibliography

Berek, C. and Ziegner, M. (1993). The maturation of the immune response. *Immunology Today*, 14:400–404.

Burges, C. J. C. (2010). Dimension reduction: A guided tour. *Foundations and Trends® in Machine Learning*, 2(4):275–365.

Castro, L. N. D. and Zuben, F. J. V. (2000). An evolutionary immune network for data clustering. In *Proceedings of the Sixth Brazilian Symposium on Neural Networks*, volume 1, pages 84–89. IEEE.

Dasgupta, D. (1998). *An overview of artificial immune systems. Artificial Immune Systems and Their Applications*, pages 3–18. Springer-Verlag.

de Castro, L. N. and Zuben, F. J. V. (2002). Learning and optimization using the clonal selection principle. *IEEE Trans. Evolutionary Computation*, 6(3):239–251.

Forrest, S., Perelson, A. S., Allen, L., and Cherukuri, R. (1994). Self-nonself discrimination in a computer. In *Proceedings of the 1994 IEEE Symposium on Security and Privacy*, SP '94, page 202, Washington, DC, USA. IEEE Computer Society.

Goodman, D., Boggess, L., and Watkins, A. (2003). An investigation into the source of power for airs, an artificial immune classification system. In *In Proceedings of the international joint conference on neural networks (IJCNN?03)*, pages 1678–1683.

Jerne, N. K. (1974). Towards a network theory of the immune system. *Annales d'immunologie*, 125C(1-2):373–389.

Kepler, T. B. and Perelson, A. S. (1993). Somatic hypermutation in b cells: An optimal control treatment. *Journal of Theoretical Biology*, 164:37–64.

Kim, H., Howland, P., and Park, H. (2005). Dimension reduction in text classification with support vector machines. *J. Mach. Learn. Res.*, 6:37–53.

Knight, T. and Timmis, J. (2002). A multi-layered immune inspired approach to data mining. In *Proc. 4th Intl. Conf. Recent Advances in Soft Computing*, pages 266–271.

Koza, J. R., Bennett, F. H., Davis, A., and Keane, M. A. (1996). Automated design of both the topology and component values of electrical circuits using genetic programming. In *Proceedings of the 1st Annual Conference on Genetic Programming*, pages 123–131, Cambridge, MA, USA. MIT Press.

Marwah, G. and Boggess, L. (2002). Artificial immune systems for classification : Some issues. In *1st International Conference in Artificial Immune Systems*, volume 1, pages 149–153. Springer.

McEwan, C. and Hart, E. (2009). On AIRS and clonal selection for machine learning. In *Artificial Immune Systems, 8th International Conference, ICARIS 2009, York, UK, August 9-12, 2009. Proceedings*, pages 67–79.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition.

Perelson, A. S. (1989). Immune network theory. *Immunological Reviews*, 110.

Sotiropoulos, D. and Tsihrintzis, G. (2017a). *Machine Learning Paradigms: Artificial Immune Systems and their Applications in Software Personalization*, chapter 7, pages 159–235. Intelligent Systems Reference Library 118. Springer International Publishing.

Sotiropoulos, D. and Tsihrintzis, G. (2017b). *Machine Learning Paradigms: Artificial Immune Systems and their Applications in Software Personalization*. Intelligent Systems Reference Library 118. Springer International Publishing.

Timmis, J. (2000). Artificial immune systems: A novel data analysis technique inspired by the immune network theory.

Timmis, J., Knight, T., de Castro, L. N., and Hart, E. (2004). *An overview of artificial immune systems. Computation in cells and tissues: Perspectives and tools for thought*, pages 51–86. Springer.

Timmis, J. and Neal, M. (2000). Investigating the evolution and stability of a resource limited artificial immune system. In Wu, A., editor, *Special Workshop on Artificial Immune Systems,*

Gentic and Evolutionay Computtion Conference (GECCO) 2000, Intelligent Robotics Group, pages 40–41, Las Vegas, Nevada, U.S.A. AAAI Press.

Timmis, J. and Neal, M. (2001). A resource limited artificial immune system for data analysis. *Knowledge Based Systems*, 14(-1):121–130.

Varela, F. J., Dupire, B., and Coutinho, A. (1988). Cognitive networks: Immune, neural and otherwise. In Perelson, A., editor, *Theoretical Immunology*, volume 2 of *SFI Series on Complexity*, pages 359–375. Addison Wesley, New Jersey.

Watkins, A. (2001). A resource limited artificial immune classifier. Master's thesis, Mississippi Sate University.

Watkins, A. and Boggess, L. (2002). A resource limited artificial immune classifier. pages 926–931. Congress on Evolutionary Computation. Part of the World Congress on Computational Intelligence, R. Ebherhart (Ed.).

Watkins, A., Timmis, J., and Boggess, L. (2004). Artificial immune recognition system (airs): An immune-inspired supervised learning algorithm. *Genetic Programming and Evolvable Machines*, 5(3):291–317.