ARTIFICIAL IMMUNE SYSTEMS BASED COMMITTEE MACHINE FOR CLASSIFICATION APPLICATION

A thesis submitted for the degree of

Doctor of Philosophy

by

Jamal Al-Enezi

Electronic and Computer Engineering
School of Engineering and Design

Brunel University

June 2012

Abstract

A new adaptive learning Artificial Immune System (AIS) based committee machine is developed in this thesis. The new proposed approach efficiently tackles the general problem of clustering high-dimensional data. In addition, it helps on deriving useful decision and results related to other application domains such classification and prediction.

Artificial Immune System (AIS) is a branch of computational intelligence field inspired by the biological immune system, and has gained increasing interest among researchers in the development of immune-based models and techniques to solve diverse complex computational or engineering problems. This work presents some applications of AIS techniques to health problems, and a thorough survey of existing AIS models and algorithms.

The main focus of this research is devoted to building an ensemble model integrating different AIS techniques (i.e. Artificial Immune Networks, Clonal Selection, and Negative Selection) for classification applications to achieve better classification results. A new AIS-based ensemble architecture with adaptive learning features is proposed by integrating different learning and adaptation techniques to overcome individual limitations and to achieve synergetic effects through the combination of these techniques.

Various techniques related to the design and enhancements of the new adaptive learning architecture are studied, including a neuro-fuzzy based detector and an optimizer using particle swarm optimization method to achieve enhanced classification performance. An evaluation study was conducted to show the performance of the new proposed adaptive learning ensemble and to compare it to alternative combining techniques. Several experiments are presented using different medical datasets for the classification problem and findings and outcomes are discussed. The new adaptive learning architecture improves the accuracy of the ensemble. Moreover, there is an improvement over the existing aggregation techniques. The outcomes, assumptions and limitations of the proposed methods with its implications for further research in this area draw this research to its conclusion.

Acknowledgements

First of all, I thank Almighty God whose wisdom enlightened my mind and whose inspiration alone helped me navigate the complex pathways of applied science and engineering in developing this thesis. All glory to His Name.

During the research and writing, I have had the opportunity of working with extraordinary colleagues to whom I am greatly indebted. I am especially grateful to Dr. Maysam Abbod, my thesis supervisor, as well as my co-supervisor, Dr. Salah Al-Sharhan, for their thought-provoking discussions, stimulating insights, suggestions for improvements and continuous guidance during this journey.

Dr. Maysam provided me with wise counsel at all times, challenging my ideas and helping me stay focused; I am sincerely grateful to him. Dr. Salah's mentoring helped reinforce my passion for research and kept my efforts from faltering. Both of them served as distinguished role models for me and I am truly thankful to them.

Finally, I am extremely grateful to my family, especially my father and my mother, my wife and my children, all of whom have patiently supported and encouraged me, gladly surrendering quality family time to my academic work. I dedicate this thesis to all these wonderful people whose unwavering love and support has sustained me over these years.

List of Publications

1- Al-Enezi J. R., Abbod M. F. and Al-Sharhan S., 2009. Advancement in Artificial Immune Systems: A Perspective of Models, Algorithms and Applications. The 5th IEEE GCC Conference and Exhibition, March, Kuwait.

2- Al-Enezi J. R., Abbod M. F. and Alsharhan S., 2010. Artificial Immune Systems - Models, Algorithms and Applications. International Journal of Research and Reviews in Applied Sciences (IJRRAS), Vol. 3, No. 2, pp. 118- 131.

3- Al-Enezi J. R., Abbod M. F. and Al-Sharhan S., 2011. Artificial Immune Systems Applications in Cancer Research. Proceedings - 4th International Conference on Developments in eSystems Engineering, DeSE 2011, pp. 168- 171.

Table of Contents

List of Figures

List of Tables

List of Nomenclature

AIS	Artificial Immune Systems
PSO	Particle Swarm Optimization
BIS	Biological Immune System
ICS	Immunity Clonal Strategy
IMSA	Immunity Monoclonal Strategy Algorithm
IPSA	Immunity Polyclonal Strategy Algorithm
ACS	Adaptive Clonal Selection
AICSA	Adaptive Immune Clonal Strategy Algorithm
RCSA	Real-Coded Clonal Selection Algorithm
PMU	Phasor Measurement Unit
DICSA	Differential Immune Clonal Selection Algorithm
CCAA	Clonal Chaos Adjustment Algorithm
ANSC	Artificial Negative Selection Classifier
V-Detector	Real-Valued Negative Selection Algorithm with Variable-Sized Detectors
AINE	Artificial Immune Network
ARB	Artificial Recognition Ball
RLAIS	Resource Limited Artificial Immune System
SSAIS	Self-Stabilizing Artificial Immune System
aiNet	Artificial Immune Network Algorithm
MLAIS	Multi-layered Artificial Immune Systems
AISEC	Artificial Immune System for E-mail Classification
IPD	Iterated Prisoner’s Dilemma
RIN	Reactive Immune Network
AINIDS	Artificial Immune Network Intrusion Detection System
LNNAIS	Local Network Neighbourhood Artificial Immune System
ALC	Artificial LymphoCyte
AINC	Artificial Immune Network Classification
KNN	K-Nearest Neighbour
TSAIN	Tree Structured Artificial Immune Network

CIN	Chaos Immune Network
AIKCN	Artificial Immune Kernel Clustering Network
FaiNet	Fuzzy Artificial Immune Network
MST	Minimal Spanning Tree
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
MV	Majority Voting
WA	Weighted Average
MV_E	Majority Voting based Ensemble
WA_E	Weighted Average based Ensemble
WA_PSO	Weighted Average with Particle Swarm Optimization Ensemble
ROC	Receiver Operating Characteristics
ANFIS	Adaptive Network based Fuzzy Inference System
ANN	Artificial Neural Networks
FL	Fuzzy Logic
WBC	Wisconsin Breast Cancer dataset
BC	Bladder Cancer dataset
HS	Haberman’s Survival dataset
PID	Pima Indians Diabetes dataset
NFS	Neuro-Fuzzy System
WA_NFS	Weighted Average with Neuro-Fuzzy System Ensemble
WA_PSO_NFS	Weighted Average with Particle Swarm Optimization and Neuro-Fuzzy System based Ensemble

CHAPTER 1: Introduction

1.1. Introduction

This thesis presents research work that proposes, develops and assesses a new adaptive learning Artificial Immune System (AIS) based Ensemble. The new proposed approach efficiently tackles the general problem of clustering high-dimensional data. In addition, it helps on deriving useful decision and results related to other application domains such classification and prediction. Decisions about clustering high-dimensional data are traditionally difficult to derive using individual techniques or based on conventional algorithms. This is due to the fact that conventional and individual clustering algorithms divide the data into clusters based on certain performance measures related to the similarity between data points. However, the problem scope is complex, broad, and consists of high-dimension search space. Moreover, the selection of the similarity and performance measures is a challenge and often adds a new dimension to the problem complexity. These aspects, in addition to the sparse nature of the data, can lead to qualitatively poor performance of the conventional algorithms and the individual techniques. In this thesis, this fact is demonstrated by presenting a case study about the performance of three popular artificial immune algorithms and then proposes two solutions to overcome these problems. First, a new ensemble based on artificial immune algorithms is presented. The proposed ensemble has a unique architecture that is based on adaptive learning detector to enhance the performance. Second, the proposed ensemble is further enhanced based on the Particle Swarm Optimization (PSO) to improve the overall performance of the architecture. These innovative solutions are combined together in an effective, computationally efficient architecture. Different samples of high-dimensional data are used to evaluate the performance of the proposed solution; the results demonstrate that the performance of the new proposed system outperforms the conventional AIS based algorithms.

The remainder of this chapter is organised as follows: the following section (1.2) presents the background and motivation of this dissertation. It describes the problem of the high-dimensional data clustering problem and finding the network relationship between the data points. Sections 1.3 and 1.4 introduce the biological immune systems and the artificial immune systems (respectively). Section 1.4 further presents the appealing features of the AIS in the computational intelligence paradigm. Sections 1.5 and 1.6 present the objectives and main contribution of the thesis. Finally, an outline of the dissertation structure is presented in Section 1.7.

1.2. Background and Motivation

Clustering is a popular approach for exploratory data analysis and mining. One of the main goals of clustering research is to design scalable and efficient algorithms for high dimension datasets (Zhang et al., 1996). At the present time, technological advances have made data collection easier and faster, resulting in larger, more complex and high dimensional data. Therefore, adaptations to existing algorithms are required to maintain cluster quality and speed as the datasets become larger and more varied. Various clustering algorithms can handle data with low dimensionality, but as the dimensionality of the data increases, these algorithms tend to fail.

The main goal of clustering is to partition a given set of data points in a multidimensional space into clusters, such that the points within a cluster are similar to one another (Fern and Brodley, 2006). In high dimensional data, there are two challenges facing clustering algorithms. First, the presence of irrelevant dimensions can mislead the clustering process by hiding clusters in noisy data. The second challenge that many clustering algorithms are facing with high dimensional data sets is the curse of dimensionality, which means that the data tend to be sparse in high dimensional space. As the number of dimensions in a dataset increases, the difference in distance between a given point and its nearest neighbour and other points in the data set often becomes negligible, making it difficult if not impossible to identify any clustering structure in the data based on distance measures (Fern and Brodley, 2006). Hence, the performances of clustering algorithms are often directly influenced by the dimensionality used for calculating the chosen distance metric (McCallum et al. 2000; Aggarwal et al., 2003; Aggarwal et al., 2004; Fern and Brodley, 2006).

Ensemble models offer a solution to challenges arising from high dimensional data used in clustering applications. Ensembles can provide robust and stable solutions by eliminating the limitations of the individual members, which may have a great impact on the final decision, hence leading to poor performance. In this study, a novel clustering biological based ensemble model is introduced with adaptive learning approaches to address these problems. The effectiveness of the proposed architecture will be demonstrated by running experiments with several real datasets, including high dimensional data set, and investigate the issue of diversity and accuracy in the ensemble model.

1.3. Biological Immune System

Immunity refers to the biological state that describes the defence mechanisms and techniques in an organism against foreign pathogens, known as antigens, which cause infectious diseases. It is the role of the Biological Immune System (BIS), which is composed of many interdependent cell types, to protect the body from a wide variety of pathogens such as viruses, bacteria, parasites and fungi.

Immune systems have many characteristics such as uniqueness, autonomous, recognition of foreigners, learning, memory, distributed detection, and noise tolerance (de Castro and Zuben, 1999). The immune system has great pattern recognition capability that may be used to distinguish between foreign cells entering the body (non-self or antigen) and the body cells (self).

The immune response is incited by the recognition of an associated molecule called an antigen. The immune system usually works according to two mechanisms called innate and adaptive immunity. Innate immunity is directed against general pathogens that enter the body while adaptive or acquired immunity allows launching an attack against any invader that the innate system cannot remove. The innate immune system plays an important role in the initiation and regulation of immune responses, including adaptive immune responses. Adaptive immunity includes immunologic memory as a significant, distinguishing characteristic.

Inspired by biological immune systems, Artificial Immune Systems (AIS) have emerged during the last decade. Many researchers have designed and built immune-based models for a variety of application domains. AIS can be defined as a computational paradigm that is inspired by theoretical immunology, observed immune functions, principles and mechanisms (de Castro and Timmis, 2003).

1.4. Artificial Immune Systems

Researchers have explored the main features of the AIS mechanisms and exploited them in many application areas. Based on their aspects, some AIS techniques have been found to be more suitable for certain application areas compared to other AIS approaches. It has been found that negative selection models and algorithms were widely used in fault detection and computer security applications utilizing the self/non-self-recognition aspect. Alternatively, the artificial immune network approaches were used in clustering, classification, data analysis and data mining applications. The clonal selection models were used mostly for optimization problems (Al-Enezi et al., 2009).

Although AIS models have achieved great successes in various application domains, there are still some theoretical issues that need to be further explored, such as the development of unified frameworks, convergence and scalability. The developments of the artificial immune systems would benefit not only from the inspiration of biological immune principles and mechanisms, but also hybridization with other soft computing paradigms, such as neural networks, fuzzy logic, and genetic algorithms. They could also be further studied and applied to more challenging application areas and to solve complex real-world problems.

1.5. Aim and Objectives

This research is devoted to discussing the advancements of the AIS as one of the emerging fields in the bio-inspired computational intelligence area. The aim is to develop an AIS based classifier system with high classification accuracy and good generalisation. This study also presents an overview of the biological immune systems, including a theoretical background on the main functions, components, and immunological mechanisms and their relation to the development of computational models for problem solving.

The main objectives of this research are:

- To provide an overview of the AIS field, including a theoretical background on the main ideas and concepts of AIS and recent advances in research literature.
- To build an ensemble model using AIS algorithms.
- To integrate different learning and adaptation techniques to overcome individual limitations and to achieve synergetic effects through the combination of these techniques.

The approach adopted focuses on building an ensemble model integrating different AIS techniques (i.e. Artificial Immune Networks, Clonal Selection and Negative Selection) for classification applications. The research followed a straightforward approach involving initial research, followed by implementation, testing, and a dissertation phase. The core of this research was its emphasis on building an ensemble model for classification applications with an optimal performance. This development was based on: (a) combining three different types of artificial immune systems (i.e., artificial immune networks models, clonal selection algorithms, and negative selection algorithms) in an ensemble architecture; (b) exploring the possibility of utilizing other soft computing and optimization techniques to further enhance the overall ensemble performance; (c) examination of the applicability of the proposed committee machine approach for classification problems; (d) comparison of the proposed model with the existing classification models; and (e) performance quantification of the developed classification ensemble model based on a number of statistical tests, benchmarks and empirical studies.

1.6. Contribution to Knowledge

The goal of this dissertation is to develop a new adaptive learning artificial immune system based ensemble to tackle the general problem of clustering high-dimension data. Some of the challenges inherent to clustering were overcome by designing new approaches for clustering ensembles through the integration of AIS models with other learning and optimization methods. A summary of the main contributions follows:

- The research provides a survey on the different AIS computational paradigms and introduces different AIS models and techniques developed in the literature since Dasgupta’s work (Dasgupta et al., 2003).

- A new biological based ensemble model is introduced integrating different AIS approaches and techniques for classification application to achieve better performance results. An empirical review has been conducted to compare the proposed ensemble model with other classification techniques.

- This work also suggests a new technique to measure the confidence level for the base classifiers in the proposed ensemble architecture. The major focus here is on assigning the weights for the base classifier on the basis of its competence in order to achieve the maximum performance for the ensemble system.

- An adaptive learning detector approach using neuro-fuzzy system to further enhance the performance has been introduced to the proposed ensemble. The suggested neuro-fuzzy detector assigns weights to the individual classifiers outputs based on their overall accuracy results before being fed to the aggregation procedure through a learning process.

- This dissertation proposes also a new optimizer based on the Particle Swarm Optimization technique as an additional improvement to the unique AIS architecture aiming for an optimum performance. The PSO based optimizer refines the weights generated from the neuro-fuzzy detector and accordingly new optimized weights will be used in the final stage of the ensemble model. The integration of all of these innovative solutions has resulted in an effective, computationally efficient ensemble model.

- Several experiments have been conducted for evaluating the effectiveness of the suggested adaptive learning AIS based ensemble approach on different medical datasets. Four medical datasets including high-dimension data were used for testing the classification problem to demonstrate the capability of the new ensemble technique and how it can be employed in dealing with real-world problems in health and cancer research.

1.7. Structure of the Thesis

The rest of this thesis is organized as follows: Chapter 2 presents a critical review of the existing AIS models and algorithms expounded in existing literature. The chapter discusses in brief the various AIS models developed based on clonal selection, negative selection and immune network theories and highlights the applications of these models in the fields of science and engineering. Hybrid intelligent systems developed based on the integration between AIS and other soft computing techniques are presented at the end.

Chapter 3 introduces the biological immune systems and highlights the role of the various organs and immune cells during immune response. The main immune system principles and mechanisms that inspire the design and developments of AIS are also discussed in the chapter. Furthermore, the chapter presents immunity-based systems and case study to test three of the well-known artificial immune systems application for cancer research.

In Chapter 4, some of the basic ideas of ensemble systems are discussed and the commonly used methods for combining classifiers in an ensemble are introduced. In addition, the chapter introduces new biological based ensemble architecture for classification problem and a new technique to measure the confidence level for the base classifiers is suggested. Another enhancement is proposed in this chapter to the AIS based ensemble using particle swarm optimization method. The chapter finally presents a case study to test the performance of the new ensemble models using a real cancer dataset.

A detector-based architecture as a main modification to the AIS ensemble is introduced in Chapter 5 using neuro-fuzzy approach to further improve the system performance. Additionally, the chapter introduces a new adaptive learning AIS based ensemble architecture as the main contribution of this thesis. The core components of the proposed adaptive learning ensemble architecture are discusses in details.

In Chapter 6, an empirical review is presented to evaluate the performance of the new adaptive learning AIS ensemble systems and compare it to alternative combining techniques. Several experiments are carried out using different real medical datasets for the classification problem to demonstrate the effectiveness of the proposed ensemble architecture and the findings and outcomes of this are briefly discussed.

Finally, Chapter 7 highlights directions for future work and forms the conclusion of this research.

CHAPTER 2: Artificial Immune Systems – A Survey[1]

2.1. Introduction

During the past decade, artificial immune systems have attracted a lot of interest from researchers aiming to develop immune-based models and techniques to solve complex computational or engineering problems. Many AIS-based algorithms have been introduced in the literature. These AIS algorithms have been developed based on the emulation of different sets of immune system principles. Among these, three main immunological principles have been considered while developing the AIS techniques: the clonal selection, the negative selection and the immune network theories. This chapter presents a survey of existing AIS models and algorithms.

Many application areas have been addressed by the AIS algorithms, including anomaly detection, pattern recognition, data mining, computer security, adaptive control and fault detection (de Castro and Zuben, 1999; Dasgupta et al., 2003; Hart and Timmis, 2005).

Table 2.1 shows a chronological list of some AIS models and techniques developed in the literature since Dasgupta's work (Dasgupta et al., 2003). A brief description for each model or technique, the aspect of the biological immune systems modelled, the type of representation used and the application area to which AIS has been applied are included in the table.

[1] Published in the 2009 5th GCC IEEE Conference and the International Journal of Research and Reviews in Applied Sciences 2010.

Table 2-1: A timeline of AIS works (2003-present)

Reference	Model or technique description	Aspects of the BIS modelled	Type of representation used	Applications
Neal, 2003	Meta-stable memory immune system for multivariate data analysis.	Immune Networks	Real-valued	Data analysis
Rouchen et al., 2003	An Immunity Clonal Strategy Algorithm (ICS) to solve multi-objective optimization task.	Clonal Selection	Real-valued vectors	Optimization
Zuo and Li, 2003	A Chaos Artificial Immune Algorithm (CAIF) by integrating of chaotic search and CLONALG	Clonal Selection	Real-valued vectors	Optimization
Nasraoui et al., 2003	Techno – streams model for detecting an unknown number of evolving clusters in a noisy data stream	Immune Networks	Real-valued	Clustering
Secker et al., 2003	An artificial immune system for e-mail classification (AISEC)	Immune Networks	Two-part words vector	Classification
Garrett, 2004	An Adaptive Clonal Selection (ACS) algorithm that suggests some modifications to the CLONALG	Clonal Selection	Real-valued vectors	Optimization
Gonzalez and Canady, 2004	A self adaptive negative selection algorithm for anomaly detection.	Negative Selection	Binary strings, real-valued	Anomaly Detection
Yu and Hou, 2004	An improved Clonal selection algorithm based in CLONALG	Clonal selection Ag-Ab binding	Binary Strings	Machine Learning
Liu et al., 2004	An Adaptive Immune Clonal Strategy Algorithm (AICSA)	Ag-Ab binding Clonal Selection	Real-valued vectors	Numerical Optimization problems
Bentley and Timmis, 2004	A Fractal immune networks model combining the ideas of fractal proteins with immune networks.	Immune Networks	Real-valued	Classification, Clustering
Luh and Lin, 2004	A Reactive Immune Network (RIN) for mobile robot learning navigation strategies within unknown environments	Immune Networks	Real-valued	Robots
Ji and Dasgupta 2004.	A Real-Valued Negative Selection Algorithm with Variable-Sized Detectors V-Detector	Negative Selection	Binary strings, real-valued	Anomaly Detection
Campels et al., 2005	A Real-coded Clonal Selection Algorithm (RCSA) that enables the treatment of real valued variables for optimization problems.	Clonal Selection	Real-valued vectors	Electromagnetic design optimization
Franca et al., 2005	A modified algorithm named dopt-aiNet as an improved version of opt-aiNet to deal with time varying fitness functions	Immune Networks	Real-valued vector	Optimization
Xian et al., 2005	A novel unsupervised Fuzzy K-Means (FKM) clustering anomaly detection algorithm based on clonal selection algorithm.	Clonal Selection	Numeric characteristic variables	Computer Security

Reference	Model or technique description	Aspects of the BIS modelled	Type of representation used	Applications
Cutello et al., 2005	Immunological algorithm for continuous global optimization problems named OPI-IA	Clonal Selection	Binary String	Optimization
Cutello et al., 2006	An improved version of OPT-IA called Opt-IMMALG	Clonal Selection	Real-code	Optimization
Qiao and Jianping, 2006	An Immune based Network Intrusion Detection System (AINIDS)	Immune Networks	Rules	Computer Security
Bian and Qiu, 2006	An adaptive clonal algorithm that suggests some modifications to the CLONALG	Clonal selection, receptor editing	Binary strings	Optimization
Karakasis et al., 2006	A hybrid model which combines clonal selection principles and gene expression programming	Clonal selection	Symbol Strings	Data Mining
Tian et al., 2006	A modified algorithm of aiNet to solve function optimization problems	Immune Networks	Real-valued vector	Optimization
Hao and Cai-Xin, 2007	Artificial immune network classification algorithm (AINC) for fault diagnosis of power transformer.	Immune Networks	Real-valued	Classification
Zhang and Yi, 2007	A Tree structured artificial immune network (TSAIN) model for data clustering and classification.	Immune Networks, Clonal Section	Real-valued	Classification, Clustering
Fu et al., 2007	A hybrid artificial immune network that uses swarm learning	Immune Networks	Real-valued	Optimization
Lv, 2007	A chaos immune network algorithm combines chaos idea with immune network to improve its ability of searching peaks.	Immune Networks,	Real-valued	Optimization
Zeng et al., 2007	A feedback negative selection algorithm (FNSA) for anomaly detection.	Negative Selection	Real-valued	Anomaly Detection
Huang and Jiao, 2007	An artificial Immune Kernel Clustering Network (IKCN) for unsupervised image segmentation.	Immune Networks	Real-valued, Image features sets	Clustering
Gan et al., 2007	A technique that combines gene expression programming with clonal selection algorithm for system modelling and knowledge discovery.	Clonal selection	Symbol Strings, Binary String	System Modelling
Graaff and Engelbrecht, 2007	A local network neighbourhood artificial immune system (LNNAIS) model for data clustering	Immune Networks	Real-valued	Clustering
Gong et al., 2007a	An improved clonal selection algorithm based on CLONALG with a novel mutation method, self-adaptive chaotic mutation.	Clonal Selection	Real-valued	Optimization

Reference	Model or technique description	Aspects of the BIS modelled	Type of representation used	Applications
Gong et al., 2007b	A differential immune Clonal selection algorithm (DICSA) combining the mechanism of Clonal selection & differential evolution	Clonal selection	Real-valued	Optimization
Zhengbing et al., 2008	A novel anomaly detection algorithm based on real-valued negative selection system	Negative Selection	Real-valued vectors	Anomaly Detection
Dabrowski and Kubale, 2008	A parallel clonal selection algorithm for solving the graph colouring problem	Clonal selection	Real-valued	Optimization
Danzhen et al., 2008	A fuzzy artificial immune network (FaiNet) algorithm for lead classification that includes three parts: AIN learning algorithm, MST algorithm and fuzzy C-means algorithm.	Immune Networks	Real-valued vectors	Classification
Lu and Zhichun, 2008	A Clonal Chaos Adjustment Algorithm (CCAA) that improves the search efficiency of CLONALG	Clonal Selection, Immune Networks	Real-valued	Multi-modal function optimization
Igawa and Ohashi, 2008	Artificial Negative selection Classifier (ANSC) that combines the negative selection algorithm with clonal selection mechanism.	Negative selection, clonal selection	Real-valued	Multi-class Classification

2.2. Clonal Selection Based Algorithms

The clonal selection principle is the whole process of antigen recognition, cell proliferation and differentiation into memory cell (Burnet, 1959). Several artificial immune algorithms have been developed imitating the clonal selection theory.

de Castro and Zuben (2002) proposed a clonal selection algorithm named CLONALG for learning and optimization, CLONALG generates a population of N antibodies, each specifying a random solution for the optimization process. At each iteration some of the best existing antibodies are selected, cloned and mutated in order to construct a new candidate population. New antibodies are then evaluated and certain percentage of the best antibodies is added to the original population. Finally, a percentage of worst antibodies of previous generations is replaced with new randomly created ones.

Rouchen et al. (2003) introduced an Immunity Clonal Strategy (ICS) algorithm that included Immunity Monoclonal Strategy Algorithm (IMSA) and Immunity Polyclonal Strategy

Algorithm (IPSA). ICS is used to solve multi-objective optimization tasks. Zuo and Li (2003) proposed a chaos artificial immune algorithm for function optimization problems. It uses chaotic variables to perform local searches and explore solution spaces.

Garrett (2004) introduced an Adaptive Clonal Selection (ACS) algorithm as a modification of CLONALG. This included some modifications of CLONALG, based on an analysis of the operators for selecting the amount of mutation and number of clones to overcome the drawbacks of the latter, such as the several parameters used and binary representation. An Adaptive Immune Clonal Strategy Algorithm (AICSA) was proposed for solving numerical optimization problems by Liu et al. (2004). It dynamically assigns the immune memory unit and antibody population according to the Ab-Ab and Ab-Ag affinities. It also integrates the local search with the global search.

Yu and Hou (2004) presented an improved clonal selection algorithm based in CLONALG algorithm. A learning operator was introduced to enhance the learning mechanism of CLONALG and to improve the detection efficiency. Campels et al. (2005) proposed a Real-Coded Clonal Selection Algorithm (RCSA) for electromagnetic design optimization. Some modifications were made to the clonal selection algorithm to enable the treatment of real valued variables for optimization problems. It has some features such as the number of clones, mutation range and the fraction of the population selected each generation. Cutello et al. (2005) devised an immunological algorithm for continuous global optimization problems named OPT-IA. The main features of the proposed algorithm include a cloning operator that explores the neighbourhood at each point within the search space and the inversely proportional hypermutation operator used in the algorithm, where the number of mutations is inversely proportional to the fitness value. Furthermore, the aging operator is used to remove the oldest candidate solution from the current populations to introduce diversity and avoid local minima during the search process.

An adaptive clonal algorithm was proposed by Bian and Qiu (2006) for optimal Phasor Measurement Unit (PMU) placement. It adjusts the number of the cycle supplement population and the probabilities of hypermutation and recombination operators of the CLONALG algorithm. These modifications can enhance the optimization process and help to avoid locally optimal traps. Cutello et al. (2006) introduced an improved version of OPT-IA called opt-IMMALG. The main modifications in this algorithm are the replacement of the binary string representation by a real-coded one and the introduction of a new inversely proportional hyper mutation operator.

Gong et al. (2007a) presented an improved clonal selection algorithm based on CLONALG with a novel mutation method, self-adaptive chaotic mutation. The main modifications are that the new algorithm adopts the logistic chaotic sequence to generate the initial antibody population, while the hypermutation adopts self-adaptive chaotic mutation. Gong et al. (2007b) later proposed a Differential Immune Clonal Selection Algorithm (DICSA) to solve the global optimization problems. It combines the clonal selection theory and differential evolution and employs three operators: a clone operator, a differential mutation crossover mutation and a standard selection operator.

A parallel clonal selection algorithm for solving the Graph Coloring Problem was presented by Dabrowski and Kubale (2008). It uses an island model wherein every processor works on its own pool of antibodies to improve the performance. Lu and Zhichun (2008) proposed a Clonal Chaos Adjustment Algorithm (CCAA) for Multi-modal Function Optimization. In order to enhance the global convergence performance of CLONALG, it takes advantages of the ergodic and dynamic properties of chaos system, and introduces the chaotic search mechanism into the CLONALG to improve its search efficiency.

Many other clonal selection based algorithms were introduced in previous studies, such as those of Jiao and Li (2005), Li et al. (2005), Jin et al. (2006), Xiu-li and Yu-qiang (2006), Halavati et al. (2007), He and Jian (2007), Hu et al. (2007), Chen (2007), Zhang et al. (2007), Li et al. (2008), Qiao et al. (2008) and Yang et al. (2008).

2.3. Negative Selection Based Algorithms

Negative selection is one of the mechanisms of the natural immune system that has inspired the developments of most of the existing artificial immune systems. In the T-cell maturation process (refer to Chapter 3) of the immune system, if a T-cell in thymus recognizes any self cell, it is eliminated before deploying for immune functionality. Similarly, the negative selection algorithm generates detector set by eliminating any detector candidate that matches elements from a group of self samples.

Negative selection based algorithms have been used in different applications areas, notably anomaly detection. Forrest (1994) proposed a negative selection algorithm whose main idea is to generate a set of detectors by first randomly making candidates and then discarding those that recognize training self-data, and then these detectors can later be used to detect anomalies.

Ayara et al. (2002) presented the NSMutation algorithm. This uses somatic hyper-mutation, eliminates redundancy and possesses tunable parameters. It consists of three phases: define self-data, generate candidate detector and compare the generated detector with self-data based on affinity threshold. Gonzalez and Cannady (2004) presented a self-adaptive negative selection approach for anomaly detection. It uses self-adaptive techniques for parameter tuning. The main two phases of the algorithm comprise the generation of the initial population and the evolution of this generated population.

A Real-Valued Negative Selection Algorithm with Variable-Sized Detectors named V-Detector was developed by Ji and Dasgupta (2004). It has many notable characteristics, such as a simple generation strategy and detector scheme, variable-sized detectors, coverage estimate and boundary-aware technique to interpret the training data set as a whole, and not as independent points.

Igawa and Ohashi (2008) proposed a new negative selection algorithm named Artificial Negative Selection Classifier (ANSC) for multi-class classification. It introduces a cutting method to reduce the effect of noise. It combines the negative selection algorithm with clonal selection mechanism to solve issues that prevent negative selection algorithms from being applied to classification problems. These issues include random searching, overfitting and incomplete information. Some other researchers proposed negative selection algorithms (Xia et al., 2007; Zeng et al., 2007; Zhengbing et al., 2008).

2.4. Artificial Immune Network Based Models

Based on the immune network theory proposed by Jerne (1974), discussed in the next chapter, many researchers developed models that use ideas and concepts from the immune network theory to solve problems in different application areas. The pioneering work of Ishiguro et al. (1994) and Hunt and Cooke (1996) inspired the development of several models by subsequent researchers. Following the work of Dasgupta et al. (2003), some of the existing immune network models are summarized below, with the specific focus on the last five years.

Timmis et al. (2000) proposed an Artificial Immune NEtwork (AINE) to perform data analysis. It uses Artificial Recognition Ball (ARB) to represent identical B-cells. Two B-Cells are linked together if the affinity between two ARBs is below a Network Affinity Threshold (NAT). Timmis and Neal (2001) developed a Resource Limited Artificial Immune System (RLAIS) based on AINE. The main enhancements in their model are the fixed total number of B-cells presented in ARBs with centralized control, whereby each ARB competes to allocate resources from the pool. The ARBs with no resources are removed from the network. The cloning and mutation process and the interactions of B-Cells are done at the ARB level.

The Self-Stabilizing Artificial Immune System (SSAIS) presented in Neal (2001) is based on RLAIS for continuous analysis of time-varying data. Unlike RLAIS, there is no limited number of resources and the control is decentralized to the level of ARBs. Castro and Zuben (2000) presented the aiNet model for data analysis tasks. The network of antibodies generated according to the Euclidean distance. It shares some features of AINE, but differs in that the immune network structure is not a part of the antibody cloning and selection process.

de Castro and Timmis (2002a) proposed the Hierarchy of aiNets model based on aiNet. The main improvements to the aiNet model were the proposed stopping criterion for the network interactive process and the introduction of an automatic hierarchical method to generate a tree of aiNets capable of detecting clusters with less-uniform characteristics. de Castro and Timmis (2002b) presented the opt-aiNet model for multimodal function optimization based on the aiNet model. The main characteristics of this model are automatic determination of the population size, combination of local with global search, well-defined stopping criterion and capability of locating and maintaining stable local optima solutions. Knight and Timmis

(2002) proposed a Multi-layered Artificial Immune Systems (MLAIS) inspired by the clonal selection theory, incorporating a feedback mechanism much like the co-stimulation in the immune network theory. It incorporates the idea of a primary immune response to deal with the event of unknown data being presented to the system.

Neal (2003) proposed a modified version of SSAIS named Meta-Stable Memory Immune System for multivariate data analysis. The model uses the stimulation function and resource allocation mechanism similar to SSAIS. It differs in that the system employs the cloning process in a primary response which is mediated by the affinity threshold, but it does not consider mutation operator. Nasraoui et al. (2003) introduced the TECNO-STREAMS model for detecting an unknown number of evolving clusters in a noisy data stream. It can model clusters with arbitrary shapes, since multiple B-cells can represent a single cluster.

An Artificial Immune System for E-mail Classification (AISEC) was presented by Secker et al. (2003). It is capable of continuous learning for the purpose of two-class classification, and is used for the task of electronic mail sorting. Alonso et al. (2004) proposed an approach based on the aiNet model to model an agent that plays the Iterated Prisoner's Dilemma (IPD). The agent structure consists of two immune networks: recognition AIN and a decision AIN. The main improvement to the aiNet is introduced in the mechanism the network uses to add B-cell to the memory.

Bentley and Timmis (2004) introduced the Fractal Immune Network combining the ideas of fractal proteins with immune networks. The model maps data items to fractal antigens, creates fractal recognition spaces similar to ARBs in dynamic networks and forms all network links by emission and reception of fractal cytokines. The system provides desirable clusters and data classification regardless of the data. Luh and Liu (2004) developed a Reactive Immune Network (RIN) for mobile robot learning navigation strategies within unknown environments. In their approach, a modified virtual target method is integrated to solve local minima problem. Franca et al. (2005) proposed a modified algorithm termed dopt-aiNet (opt-aiNet for dynamic environments) to deal with time-varying fitness functions as an improved version of opt-aiNet. The main improvements presented in their approach are the use of separate memory subpopulation, a line search procedure, two new mutation operator schemes, a cell line suppression mechanism, and a limited population size.

Qiao and Jianping (2006) developed an Artificial Immune based Network Intrusion Detection System (AINIDS). It consists of five components: a data collector, a packet head parser and feature extraction, antibody generation and antigen detection, co-stimulation and report and rule optimization components. Tian et al. (2006) proposed a modified algorithm of aiNet to solve function optimization problems. The main improvements presented in this algorithm are: 1) the searching radius is a variable parameter depending in the number of the generations in which a cell survives; 2) the capability to reserve the cell with the largest fitness (elitist strategy); and 3) the expanding rate is controlled to maintain the diversity of the network.

Graaff and Engelbrecht (2007) introduced a Local Network Neighbourhood Artificial Immune System (LNNAIS) model for data clustering. Compared to the existing AIS models, LNNAIS uses the concept of Artificial Lymphocyte (ALC) neighbourhood to determine the network links between the ALCs. There is no network affinity threshold in this model that determines whether two ALCs should be linked to form a network. The lymphocytes neighbours are determined by their individual indexes and they interact and learn from one another to have a better local representation of patterns.

Hao and Cai-Xin (2007) proposed an Artificial Immune Network Classification algorithm (AINC) for fault diagnosis of power transformer. The algorithm consists of three steps: classifying the fault samples into training and testing antigens sets and initialize them; using AINC to train the antigens set to obtain memory antibodies; and calculating the Euclidean distance among the test antigens set and memory antibodies, and classifying fault samples using the K-Nearest Neighbour (KNN) approach.

Zhang and Yi (2007) proposed a Tree Structured Artificial Immune Network (TSAIN) for data clustering and classification. In this model, a topological link is set up between two antibodies immediately after one has reproduced by another, with no need to set a threshold for this connection. It consists of four phases: the clonal section, the antibody cooperation, the antibody suppression and the topology updating phases. The first two phases provide the network with self-organizing ability. The suppression and topology updating ensures the consistency of network topology with distribution of clusters.

In Lv (2007), a Chaos Immune Network (CIN) algorithm for multimodal function optimization is discussed. The main features of this algorithm are the use of chaos variable to simulate proliferation mode of immune cells to enhance searching accuracy, the stepping criteria was improved and some relevant measures have been added to avoid pre-maturation. Huang and Jiao (2007) presented an Artificial Immune Kernel Clustering Network (IKCN) for unsupervised image segmentation. It combines the artificial immune network and the support vector domain description. In this model the image features sets are divided into subsets by the antibodies, then each subset is mapped into a hypersphere in a high dimensional feature space by a Mercer Kernel. Finally, a minimal spanning tree is used to automatically determine the final number of clusters without a predefined number of clustering. Some other proposed immune network algorithms can be found in Li et al. (2008).

2.5. Hybrid Computational Intelligent Systems

Hybrid intelligent systems development is one of the most intensively growing areas. Hybrid systems utilize various soft computing methods and techniques like artificial neural networks, fuzzy systems, artificial immune systems, evolutionary computation, and genetic algorithms. The main objective is to integrate different learning and adaptation techniques to overcome individual limitations and to achieve synergetic effects through the combination of these techniques. This has contributed effectively in the developments of a large number of new intelligent system designs in recent years.

In this context, researchers have explored combining AIS with other computational models and techniques, especially with soft-computing methods. Some of the earlier work that combined AIS ideas with genetic algorithms was developed by Hajela et al. (1997). They used immune networks to improve the convergence of genetic algorithms for design. Dasgupta (1997) pointed out the similarities and the differences between AIS and artificial neural networks. Nasraoui et al. (2002) proposed the Fuzzy AIS model, which uses a fuzzy set to model the area of influence of each B-cell, which makes it more robust to noise. Vergas et al. (2003) presented an immune learning classifier network named CLARINET for autonomous navigation by combining the strengths of learning classifier systems, evolutionary algorithms, and an immune network model. Xian et al. (2005) proposed a novel intrusion detection method that optimizes the objective function of unsupervised fuzzy k-means clustering based on clonal selection algorithm.

Karakasis and Stafylopatis (2006) introduced a hybrid technique for data mining tasks which combines clonal selection principles and gene expression programming. Fu et al. (2007) proposed a hybrid artificial immune network which uses the swarm learning of particle swarm optimization to speed up the convergence of artificial immune system. Gan et al. (2007) proposed a technique that combines the simple representation method of gene expression programming and the advantage of clonal selection algorithms. Danzhen et al. (2008) introduced a fuzzy artificial immune network (FaiNet) algorithm for load classification. It consists of three parts: the artificial immune network learning algorithm, the minimal spanning tree algorithm, and the classification algorithm based on fuzzy C-means algorithm.

2.6. Summary

This chapter gives an overview of the various models and algorithms of the artificial immune systems as well as their applications in the real-world problems. The chapter briefly introduces and discusses the models and algorithms that have been developed based on various computational aspects of the immune system. Furthermore, hybrid intelligent systems combining artificial immune systems with other soft computing techniques have been highlighted.

AIS models and algorithms have been applied in various application domains. For example, ClonalG and opt-aiNet algorithms were used for optimization problems. Furthermore, aiNet algorithm can be categorized within the clustering algorithms, while negative selection algorithms such as V-Detector were used mainly in anomaly detection applications. AIS algorithms can be classified into population-based and network-based categories. Thus, the negative and clonal selection based algorithms are included in first category and immune network models are considered in the second category (de Castro and von Zuben, 2002).

Although AIS models have achieved great successes in various application domains, there are still some theoretical issues that need to be further explored. As mentioned by Dasgupta et al. (2003), these open issues include:

- Most of the existing AIS algorithms have been exploratory, and they do not scale.
- The efficiency of the AIS algorithms needs to be improved.
- Enhancement of the representation.
- Development of a unified framework that can integrate several AIS models.

The developments of the artificial immune systems would benefit not only from the inspiration of biological immune principles and mechanisms, but also from hybridization with other soft computing paradigms, such as neural networks, fuzzy logic, and genetic algorithms. They could also be further studied and applied to more challenging application areas and to solve complex real world problems.

The following chapter presents a theoretical background of the biological immune system including the main components of the immune system and the immunological principles that inspired the development of the AIS field. The chapter also introduces immunity-based systems, discussing three popular AIS algorithms in detail, with case studies based on actual cancer datasets.

CHAPTER 3: Biological Based Computational Systems

3.1. Introduction

This chapter introduces the Biological Immune System and particularly discusses the roles of various organs and immune cells once an antigen invades the human body. It also discusses important principles and mechanisms of immune systems in order to describe the behaviour and the immunological processes of the BIS during immune responses. The chapter discusses three immune mechanisms that inspired and are primarily used in the development of AIS computational methods, namely immune network theory, clonal selection principles and negative selection mechanisms.

Artificial Immune Systems incorporate many properties of natural immune systems, including diversity, distributed computation, error tolerance, dynamic learning and adaptation, and self-monitoring. AIS utilizes the biological immune system's remarkable pattern-matching ability, used to distinguish between foreign cells entering the body (referred to as non-self, or antigen) and the cells belonging to the body (referred to as self). Furthermore, this chapter presents several AIS models and algorithms and case studies to test AIS application for cancer research by validation against actual cancer dataset. Three popular AIS algorithms inspired by the immunological principles are considered for the case studies, including the CloanlG (de Castro and Zuben, 2002), V-Detector (Ji and Dasgupta, 2004a), and aiNet (de Castro and von Zuben, 2001), which are discussed in detail in the following sections, since they are used as the comparison foundation in this work.

3.2. Immunity – Theoretical Background

3.2.1. The Biological Immune System

Immunity refers to the biological state that describes the defence mechanisms and techniques of an organism against foreign pathogens, known as antigens, which cause infectious diseases. The vertebrate immune system is composed of diverse sets of cells and molecules that work together with other systems (such as the neural and endocrine), to maintain homeostasis. The primary function of the immune system is to protect the body from infectious agents, such as viruses and bacteria, commonly known as pathogens. In this section, a general overview of the immune system is presented to introduce the reader to its anatomy and the main cells that are responsible for defence responses. Detailed information on the functional elements of the immune system can be found in previous studies (de Castro and Zuben, 1999; Hofmeyr, 2000; Parkin and Cohen, 2001; Dasgupta and Nino, 2009).

3.2.2. Cells of the Immune System

All immune cells are generated as immature stem cells in the bone marrow. Some of the immature stem cells develop through a maturation process within the marrow, whereas others leave the bone marrow and migrate into the thymus. The immune cells respond to different cytokines and other chemical signals to grow into specific immune cell types, such as T cells, B cells, or phagocytes. There are several types of immune cells; however, the focus of this study concerns the major cell types, which are lymphocytes and phagocytic/dendritic cells.

- **Lymphocytes**

Lymphocytes are white blood cells produced in the bone marrow specialized mainly in the recognition of pathogens (de Castro and Timmis, 2003). There are two broad sub-types of lymphocyte known as B cells and T cells. All of them originate in the bone marrow, but T cells undergo a process of maturation in the thymus gland. These two types of mature cells are of similar appearance, but they differ in the way they identify antigens. B and T cells circulate in the blood and through body tissues.

B cells are specialized white blood cells produced and developed in the bone marrow. It gives rise to plasma cells which generate substances called antibodies in the body's fluids, which bind to foreign antigens circulating in the bloodstream. Each B cell is programmed to make one specific antibody. When a B cell encounters an antigen that triggers it to become active, it gives rise to many large cells known as plasma cells, which produce antibodies. Unlike B cells, T cells have to migrate to the thymus where the maturation process occurs. T cells can only recognize antigens that are presented by other accessory cells. T cells contribute to immune system defences in two major ways (Dasgupta and Nino, 2009):

1- Directing and regulating immune responses.
2- Directly attacking infected or malignant cells.

There are five different types of T cells produced in the thymus: Delayed hypersensitivity, Helper, Cytotoxic, Memory and Suppressor T cells (Dasgupta and Nino, 2009). Each of these cells has important functions in an immune response. *The delayed hypersensitivity T cell (T_{DH})* secretes cytokines that can mediate cellular immunity and activate phagocytic cells for more effective immune response. *The helper T cell (T_h)* produces signals by releasing cytokines that prompt the proliferation of both T cells and B cells. T_h cells also secrete cytokines that enhance the immune response. *Memory T cells* are maintained in the body to remember previous antigens so that when the antigen is encountered again, it quickly respond by giving rise to additional helper, memory, and cytotoxic T cells. *The cytotoxic T cells (T_C)*, also called killer T cells, are the true effecter cells in cell-mediated immunity. These cells recognize the infected self-cells, tumour cells and other cells carrying certain foreign or abnormal molecules on their surfaces and then launch an attack to kill and destroy the infected cell. *Suppressor T cell or Regulatory T cell (T_{reg})* is a T cell that reduces or suppresses the immune response of B cells or of other T cells to an antigen. These cells are involved in shutting down immune responses after a successful elimination of invading organisms in order to prevent excessive reactions (de Castro and Zuben, 1999; Hofmeyr, 2000; Dasgupta and Nino, 2009).

• Phagocytic and Dendritic Cells

A *phagocyte* is a large white immune cell that engulfs and kills microbes and other foreign particles. Phagocytes also play an important role in the disposal of dead or dying cells caused by tissue injury. Three types of white blood cells can act as phagocytes: neutrophils, eosinophil and monocytes. Phagocytes circulate in the bloodstream waiting for chemical signals from dying cells, which allows them to detect their decline. Upon receiving these signals, the white blood cell phagocytes migrate to the site of infection and ingest the dying cells through a process called phagocytosis.

Dendritic Cells (DCs) get their name from their surface projections and are found in the parts of lymphoid organs where T cells also exist. Dendritic cells in lymphoid tissues perform two main functions: they display antigens to T cells and help stimulate T cells during an immune response. Once activated, DCs migrate to the lymph nodes where they interact with T cells and B cells to initiate and shape the adaptive immune response (Parkin and Cohen, 2001; Dasgupta and Nino, 2009).

3.2.3. The Organs of the Immune System

Each organ of the immune system plays a different role in defending the body against pathogens. There is no central organ that controls how the immune system functions. The components of the human immune system are shown in Figure 3.1.

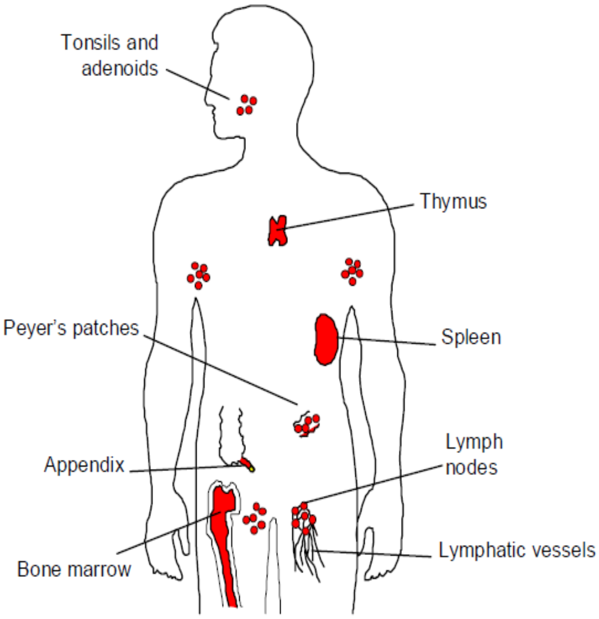


Figure 3-1: Anatomy of the Immune System (de Castro and Zuben, 1999)

3.2.3.1. Bone Marrow

The bone marrow is the place where all the immune system cells are initially generated and formed through a process called hematopoiesis. During this process, the generated stem cells differentiate into either mature cells of the immune system or into precursors of cells that migrate and develop outside the bone marrow. The bone marrow produces B cells, natural killer cells, granulocytes and immature thymocytes, in addition to red blood cells and platelets (Dasgupta and Nino, 2009).

3.2.3.2. Thymus

The thymus is an organ located behind the breastbone, which is responsible for the maturation of T cells in the immune system. Immature thymocytes, also known as prothymocytes, leave the bone marrow and migrate into the thymus. During the maturation process, T cells that are containing receptors capable of recognizing self-antigens and may cause an autoimmune response are excluded from the population of T cells. The mature T cells are then released into the bloodstream (Hofmeyr, 2000; Parkin and Cohen, 2001).

3.2.3.3. Spleen

The spleen is an organ located in the upper-left portion of the human abdomen. It is made up of B cells, T cells, macrophages, dendritic cells, natural killer cells and red blood cells. The spleen acts as an immunologic filter of the blood. It removes and destroys old red blood cells, called erythrocytes, from the blood supply and removes, stores and produces white blood cell lymphocytes (B cells). These stored B cells are activated in the spleen and produce antibodies that will assist in removing microbes and other debris from the blood supply (Dasgupta and Nino, 2009).

3.2.3.4. Lymph Nodes

Lymph nodes filter the lymphatic fluid and store special cells that can eliminate antigens, bacteria or tumour cells that are travelling through the body in the lymph fluid. The lymph nodes are located in different places in the human body, and they are made up of T cells, B cells, dendritic cells and macrophages. Lymph nodes are critical for the immune responses and are principal sites where many immune reactions are initiated (Hofmeyr, 2000; Dasgupta and Nino, 2009).

3.2.4. Immune System Mechanisms

The immune response is incited by the recognition of an associated molecule called antigen. Immune system usually works according to two mechanisms: Innate and Adaptive Immunity. Innate immunity is directed against general pathogens that enter the body, while adaptive or acquired immunity allows launching an attack against any invader that innate system cannot remove. The adaptive immunity includes immunological memory as a significant, distinguishing characteristic. For more information about the immune system mechanisms, the reader can refer to previous studies (de Castro and Zuben, 1999; de Castro and Timmis, 2003; Timmis et al. 2004).

- **Innate Immunity**

Vertebrates are born with this immunity, which plays a vital role in the initiation and regulation of immune responses. Specialized cells have evolved to recognize and bind to common molecular patterns of micro-organisms. However, it does not provide blanket protection, as it is primarily static in nature (de Castro and Zuben, 1999; Dasgupta and Nino, 2009).

- **Adaptive Immunity**

Adaptive immunity is directed towards specific invaders, and is modified by exposure to invaders - either those previously encountered by the body's immune system, or novel antigens. It mainly consists of lymphocytes (white blood cells, more specifically B and T type) that aid the process of recognizing and destroying specific substances, and are antigen-specific (de Castro and Zuben, 1999; Dasgupta and Nino, 2009).

3.3. Artificial Immune Systems

Artificial immune systems are computational paradigms that belong to the computational intelligence family and are inspired by the biological immune system. During the past decade, they have attracted a lot of interest from researchers aiming to develop immune-based models and techniques to solve complex computational or engineering problems. Similarly to other bio-inspired computing paradigms, the AIS intends to capture some of the immune system principles and processes previously described within a computational perspective. The main objective is to utilize the appealing features of the natural immune system including pattern recognition, learning, memory and self-organisation. Hence, the field of immunology inspires computer scientists and creates much scope for work within the area of AIS (Somayaji et al. 1997; Delahunty and Callaghan, 2004; Timmis et al., 2008).

Three well-known AIS algorithms have been selected in this work to test their applicability for the classification problem and then used on building the AIS based committee machine. The base of selection was due to the difference on the learning process for each algorithm and since they all are unsupervised learners. Features of these three AIS algorithms are described below. Greater details may be found in the original papers cited.

3.3.1. ClonalG Algorithm

ClonalG algorithm is based on clonal selection theory (Burnet, 1959). The theory is used to explain the basic response of adaptive immune system to antigenic stimulus. It establishes the idea that only those cells capable of recognizing an antigen will proliferate while other cells are ignored. Clonal selection operates on both B and T cells. B cells, when their antibodies bind with an antigen, are activated and differentiated into plasma or memory cells. Prior to this process, clones of B cells are produced and undergo somatic hyper mutation. As a result, diversity is introduced into the B cell population. Plasma cells produce antigen-specific antibodies that work against antigens. Memory cells remain with the host and promote a rapid secondary response (de Castro and Timmis, 2003).

de Castro and Zuben proposed a clonal selection algorithm named ClonalG for learning and optimization (de Castro and Zuben, 2002). They considered the main features of the clonal selection theory while developing their algorithm, including: maintenance of a specific memory set, selection and cloning of the most stimulated antibodies, death of non-stimulated antibodies, affinity maturation and re-selection of the clones proportionally to their antigenic affinity, and generation and maintenance of diverse set of antibodies. The main steps of the ClonalG algorithm are described in algorithm 3.1.

Algorithm 3.1: ClonalG Algorithm adopted from (de Castro and Zuben, 2002)

```
Input: Ab , gen, n, d, L, β;

// Ab: available antibody repertoire
// gen: no of generations
// n: no of antibodies to select for cloning
// d: lowest affinity antibodies to be replaced
// L: bit string length for each antibody
// $\beta$: cloning factor

Output: Ab, f

1.  for t = 1 to gen,

        1.1  f := decode ( Ab ); f  vector containing the affinity of all antibodies with
relation        to antigen

        1.2  $Ab_n$ := select ( Ab, f, n );

        1.3  C := clone ( Abn, $\beta$, f );

        1.4  C* := hypermut  ( C, f );

        1.5  f := decode ( C* );

        1.6  $Ab_n$ := select ( C*, f, n );

        1.7  Ab := insert ( Ab, $Ab_n$ );

        1.8  $Ab_d$ := generate ( d, L );  Randomly generate d antibodies of  length  L

        1.9  Ab := replace (Ab, $Ab_d$, f );

end;

2.  f := decode ( Ab );   Function decode is supposed to decode Ab and evaluate for
   these   decoded values.
```

The affinity between an antibody and an antigen can be defined using different techniques such as matching rules and distance measures. One of the commonly used techniques is the Euclidean distance, which is suitable when using a real-valued vector representation (Dasgupta and Nino, 2009). To explain how the ClonalG works using the Euclidean distance, let $\boldsymbol{Ag} = \{\boldsymbol{Ag}_1, \boldsymbol{Ag}_2, \ldots, \boldsymbol{Ag}_N\}$ and $\boldsymbol{Ab} = \{\boldsymbol{Ab}_1, \boldsymbol{Ab}_2, \ldots, \boldsymbol{Ab}_N\}$denote the antigens and antibodies sets respectively, where N is their common order. Then, the matching degree d between $\boldsymbol{Ag}$ and $\boldsymbol{Ab}$ can be calculated based on the Euclidean distance:

$$d=\sum_{i=1}^{N}(\mathbf{Ag_i}-\mathbf{Ab_i})^2 \qquad (3.1)$$

d is then compared with a preset threshold $\boldsymbol{\lambda}$, and the matching error E obtained by:

$$E = d - \lambda \qquad (3.2)$$

If E > 0, then we consider that the two vectors are not matching, therefore the antigen has not been recognized by the antibodies. If E <= 0, we conclude that ***Ag*** matches ***Ab***, hence the antigen has been recognized by the antibodies. The common range for the affinity calculation is varies between 0 and 1.

For the affinity maturation, ClonalG algorithm assumed that the *n* highest affinity antibodies were sorted in ascending order, and the amount of clones generated for all these *n* selected antibodies is given by:

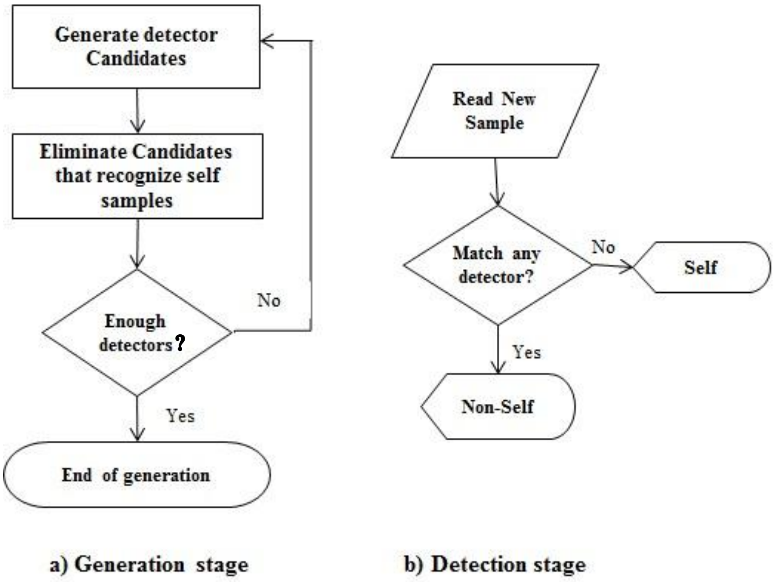
$$Nc = \sum_{i=1}^{n} \text{round}\left(\frac{\beta . N}{i}\right) \qquad (3.3)$$

where Nc is the total amount of clones generated for each of the antigens, β is a cloning factor specifies the scaling factor for the number of clones generated for the selected antibodies and its common values $\beta \in (0,1]$, N is the total amount of antibodies and round() is the operator that rounds its argument towards the closest integer. Each term of this sum corresponds to the clone size of each selected antibody, e.g. for N = 100 and β = 1, the highest affinity antibody (i = 1) will produce 100 clones, while the second highest affinity antibody produces 50 clones, and so on (de Castro and Zuben, 2002).

3.3.2. V-Detector Algorithm

V-Detector algorithm is based on the negative selection mechanism, which aims at the protection of the body against self-reactive lymphocytes. Negative selection is a biological process by which the natural immune system generates non-self detectors that do not detect self structures. Similarly, the negative selection algorithm generates detector set by eliminating any detector candidate that match elements from a group of self samples. Negative selection based algorithms have been used in different applications areas, such as anomaly detection.

The AIS algorithms developed based on the negative selection mechanism basically consist of two phases. In the first phase, the detector set is generated randomly as part of the training or generation stage. Then, the new sample is examined using the detector set obtained during the training phase and classified as either self or non-self sample. On the other hand, if the new sample is recognized by any detector in the detector sets, then it is classified as a non-self sample (represent class 1 of the data). On the other hand, if it is not matching any of the detectors, then it is considered as a self sample (represent class 0 of the data). Figure 3.2 illustrates the basic idea of negative selection algorithms.



a) Generation stage b) Detection stage

Figure 3-2: Negative Selection Main Idea

Ji and Dasgupta introduced a real-valued negative selection algorithm with variable-sized detectors named V-Detector (Ji and Dasgupta, 2004a). It suggests a simple detectors generation strategy and matching scheme and includes a new variable parameter, which is the radius of each detector. Several versions exist of the V-Detector algorithm. The earlier version called point-wise V-Detector that treats each training data point (self sample) individually (Ji and Dasgupta, 2004b). In a later version, a new feature of negative selection algorithm was introduced that enables the V-Detector to detect the boundary of self region. This version was named a boundary-aware V-detector (Ji Z., 2004). The authors subsequently proposed a statistical mechanism to analyze the detector coverage namely a quantitative measurement of a detector set's capability to detect non-self sample (Ji and Dasgupta, 2005).

The V-Detector algorithm randomly generates detector candidates one at a time instead of generating a full set of detectors. Each individual candidate is then examined using the

Euclidean distance matching rule. If the distance to the nearest self sample is less than r_s, a threshold value which represents the radius of this self sample, the detector is eliminated and a new candidate is generated. If the minimum distance to any self sample is greater than the r_s, then the detector is retained and its radius is kept as the minimum distance to the nearest self sample. The generation phase finishes when a preset number of detectors are obtained.

Several key running parameters exist in the V-Detector algorithm: p is the target coverage of the non-self region by all the existing detectors for hypothesis testing, α is the significant level for hypothesis testing, n is the sample size, r_s is the self radius, z is the standard score for z score using "Central Limit Theorem", and z_α is the z score for a confidence level of 1-α. The V-detector algorithm is shown in Algorithm 3.2, which is used in the case study.

Algorithm 3.2: V-Detector Algorithm adopted from (Ji and Dasgupta, 2005)

```
V-Detector - Set ( $S$, $T_{max}$, $r_s$, $c_o$ )
$S$ : set of self samples
$T_{max}$ : maxium number of detectors
$r_s$ : self radius
$c_o$ : expected coverage
1: D $\leftarrow \emptyset$
2: Repeat
3:      t $\leftarrow$ 0
4:      T $\leftarrow$ 0
5:      $r \leftarrow \infty$
6:      $x \leftarrow$ random sample from $[1, 0]^n$
7:      Repeat for every $d_i$ in D = {$d_i$, i = 1, 2, …}
8:          $d_d \leftarrow$ Euclidean distance between x ($d_i$) and $x$, where x($d_i$) is the location of $d_i$
9:          if $d_d \leq$ r($d_i$) then, where r($d_i$) is the radius of detector $d_i$
10:                 $t \leftarrow t + 1$
11:                 if $t \geq 1 / (1 - c_o)$ then return D
12:                 go to 4:
13:      Repeat for every $s_i$ in $S$
14:          d $\leftarrow$ Euclidean distance between $s_i$ and $x$
15:          if d - $r_s \leq r$ then $r \leftarrow$ d - $r_s$
16:      if $r > r_s$ then D $\leftarrow$ D $\cup$ {< $x$, $r$ >}, where < $x$, $r$> is a detector with location $x$ and
          radius $r$
17:      else T $\leftarrow$ T + 1
18:      if T > 1 / ( 1- maximum self coverage ) exit
19: Until | D | = $T_{max}$
```

```
20: return  D
```

3.3.3. aiNet Algorithm

The main idea of the immune networks theory is that the immune system maintains an idiotypic network of interconnected B cells for antigen recognition. These cells interconnect with each other in certain ways that lead to the stabilization of the network. Two B cells are connected if the affinities they share exceed a certain threshold, and the strength of the connection is directly proportional to the affinity they share (Jerne, 1974).

Inspired by immunological principles, including clonal selection, affinity maturation, and immune network theory, the artificial immune network model aiNet was presented by de Castro and von Zuben (de Castro and von Zuben, 2001). The aiNet model has the capability of reducing redundancy and describing immune network structure, including data distribution and clustering.

The learning procedure for the aiNet algorithm consists of two main steps. First, the clonal selection principle and affinity maturation interactions are applied, whereby the antibodies go through the cloning and mutation processes in order to recognize the antigens. This stage corresponds to the clonal selection algorithm (ClonalG) originally proposed by de Castro and von Zuben, which was outlined in the previous section (de Castro and von Zuben, 2002). The raw training data is explored and compressed by the aiNet, leading to an antibody network by extracting the most relevant information from the data for clustering purposes.

The second step of the aiNet includes the immune network interactions and introduction of diversity. The Minimal Spanning Tree (MST), one of the hierarchical and graph-theoretical clustering techniques, is used in this stage to define the final network structure. This method is built on the antibody network, whereby the inconsistent edges are identified and removed, which can transform the network by separating the data into clusters (de Castro and von Zuben, 2001). The aiNet adaptation procedure is described in Algorithm 3.3.

Algorithm 3.3: aiNet Algorithm adopted from (de Castro and von Zuben, 2000)

```
$\boldsymbol{X}$: data set composed of $N_p$ patterns of dimension $p$;
$\boldsymbol{C}$: matrix containing all the $N_t$ network cells ( $\boldsymbol{C} \in \Re^{N\text{txp}}$ );
$\mathbf{M}$: matrix of the $N$ memory cells, ($\mathbf{M} \subseteq \boldsymbol{C}$);
$Nc$: total number of clones generated by each stimulated cell;
$\mathbf{D}$: dissimilarity matrix with elements $d_{ij}$ (Ag-Ab);
$\mathbf{S}$: similarity matrix with elements $s_{ij}$ (Ab-Ab);
$n$: $n$ highest affinity cells selected to clone and mutate;
$\zeta$: percentage of the matured cells to be selected; and
$\sigma_{\mathrm{d,s}}$: natural death and suppression threshold, respectively.
1. At each iteration step, do:
   1.1 For each antigen $i$, do:
       1.1.1  Determine its affinity, $d_{ij}$, to all the network cells according to
              a distance metric;
       1.1.2  Select the $n$ highest affinity network cells;
       1.1.3  Reproduce (clone) these $n$ selected cells. The higher the cell affinity,
              the larger $Nc$;
       1.1.4  Apply Equation ( $\boldsymbol{C} = \boldsymbol{C} - \alpha\ (\boldsymbol{C} - \boldsymbol{X})$ ) to these $Nc$ cells;
       1.1.5  Determine $\mathbf{D}$ for these improved cells;
       1.1.6  Re-select $\zeta$ % of the highest affinity cells and create a partial $\mathbf{M}_p$
              memory cell matrix;
       1.1.7  Eliminate those cells whose affinity is inferior to threshold $\sigma_{\mathrm{d}}$,
              yielding a reduction in the size of the $\mathbf{M}_p$ matrix;
       1.1.8  Calculate the network Ab-Ab affinity, $s_{ij}$;
       1.1.9  Eliminate $s_{ij} < \sigma_{\mathrm{s}}$ (clonal suppression);
       1.1.10 Concatenate $\boldsymbol{C}$ and $\mathbf{M}_p$, ($\boldsymbol{C} \leftarrow [\boldsymbol{C}\ ;\ \mathbf{M}_p]$);
   1.2 Determine $\mathbf{S}$, and eliminate those cells whose $s_{ij} < \sigma_{\mathrm{s}}$ (network suppression);
   1.3 Replace r % of the worst individuals;
2. Test the stopping criterion.
```

Compared to the other AIS algorithms, the ClonalG algorithm is low in complexity and has a small number of user parameters. However, the main issue with ClonalG is scalability, since the number of clones per generation which increases in proportion to the number of antibodies may reaches unlimited value. While the V-Detector algorithm is more effective in using smaller number of detectors because of their variable sizes. But like other negative selection algorithms, V-Detector algorithm has the limitations of one-class classification and specific matching rules (Ji and Dasgupta, 2009). On the other hand, aiNet has the capability of reducing redundancy and describing immune network structure, including data distribution and clustering; however, it also has some drawbacks, including its high number of parameters and high time complexity.

3.4. Case Studies

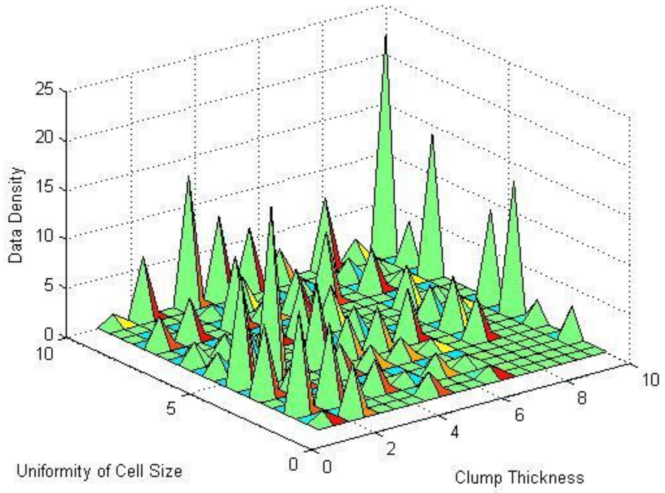
Some experiments were carried out in order to test three of the well-known AIS algorithms and to explore their capabilities. The ClonalG, V-Detector, and aiNet algorithms were chosen for this case study and tested on a cancer data set for classification purposes. The Matlab software was used for all the codes, running on a Windows 7 (64-bit operating system) machine with Intel core processor i7-2.20 GHz CPU and 8 GB RAM.

3.4.1. Data Set

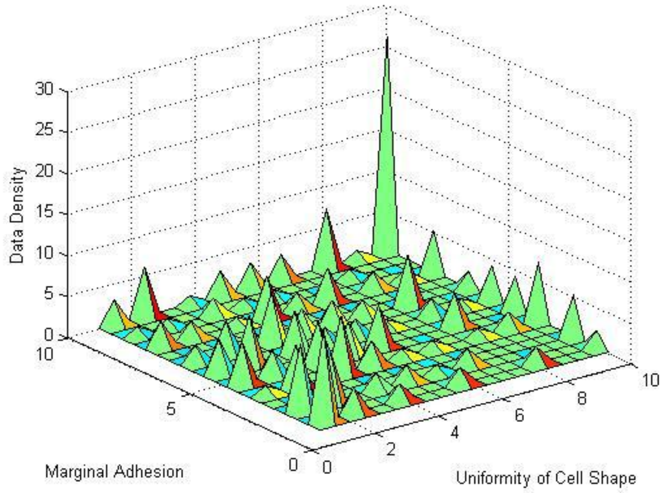
The dataset used in the case studies is the Wisconsin breast cancer data set, obtained from the University of Wisconsin Hospitals (Frank and Asuncion, 2010). It has 699 samples with 9 attributes, where each instance has one of two possible classes: benign or malignant. Table 3.1 shows a snapshot from this dataset. Figure 3.3 shows the data distribution graphs for the Wisconsin breast cancer data set.

Table 3-1:Sample of Wisconsin breast cancer data set

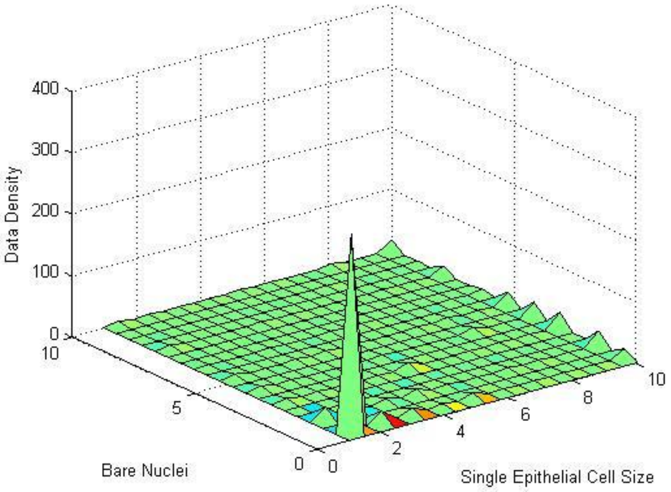
Clump Thickness	Uniformity of Cell Size	Uniformity of Cell Shape	Marginal Adhesion	Single Epithelial Cell Size	Bare Nuclei	Bland Chromatin	Normal Nucleoli	Mitoses	Class
5	3	3	3	2	3	4	4	1	2
1	1	1	1	2	3	3	1	1	1
8	7	5	10	7	9	5	5	4	2
7	4	6	4	6	1	4	3	1	2
4	1	1	1	2	1	2	1	1	1
4	1	1	1	2	1	3	1	1	1
10	7	7	6	4	10	4	1	2	2
6	1	1	1	2	1	3	1	1	1
7	3	2	10	5	10	5	4	4	2
10	5	5	3	6	7	7	10	1	2


Data Density
0
5
10
15
20
25
10
5
0
Uniformity of Cell Size
0
2
4
6
8
10
Clump Thickness

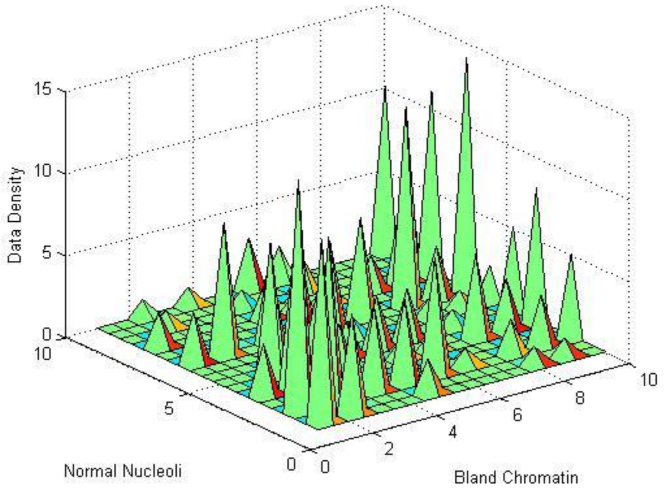
(a)


Data Density
0
5
10
15
20
25
30
10
5
0
Marginal Adhesion
0
2
4
6
8
10
Uniformity of Cell Shape

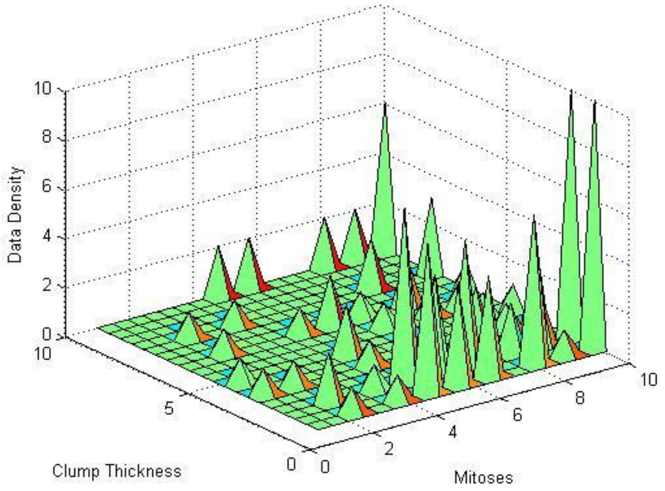
(b)


Data Density
0
100
200
300
400
10
5
0
Bare Nuclei
0
2
4
6
8
10
Single Epithelial Cell Size

(c)



(d)



(e)

Figure 3-3: Data distribution graph for WBC dataset

Initially, the dataset was normalised to unity before being fed to the algorithms. Once the data was normalised, a five percent of the samples were chosen at random and removed from the data set. These then became the testing samples, and the rest were used for the training phase. The training and the testing phases were repeated 100 times (was chosen to test the reliability of the system) for each experiment, using different samples chosen randomly.

3.4.2. Performance Measures

Some testing measures were used, such as accuracy, sensitivity, and specificity to evaluate the performance of the AIS algorithms. The predictive accuracy of the classifier measures the proportion of correctly classified instances; sensitivity measures the fraction of actual positive examples that are correctly classified; and specificity measures the fraction of actual negative examples that are correctly classified.

All these measures can be calculated by measuring true and false positives (TP, FP) in addition to the true and false negatives (TN, FN) as follows (Gambino, 2006; Sokolova et al., 2006):

$$Accuracy = \frac{(\mathrm{TP} + \mathrm{TN})}{(\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN})} \qquad (3.4)$$

$$Sensitivity = \frac{\mathrm{TP}}{(\mathrm{TP}\ +\ \mathrm{FN})} \qquad (3.5)$$

$$Specificity = \frac{\mathrm{TN}}{(\mathrm{TN}\ +\ \mathrm{FP})} \qquad (3.6)$$

Accordingly, the overall performance of the algorithm can be calculated using the following equation:

$$\boldsymbol{Total} = Accuracy + Sensitivity + Specificity \qquad (3.7)$$

3.4.3. Experiment Results

This section presents a comparative analysis of the three algorithms described above. For the ClonalG algorithm, the stopping criterion is set as a fixed number of generations: gen = 3. Initially, all of the parameters were changed in values to test their effects on the accuracy results and the dependency between them. These variables were varied on the following way without significant effect in the results except for the test tolerance:

- The number of detectors for the training and testing stage was varied from 300 to 2000 detectors.
- The number of clones was varied from 2 to 20.
- The test tolerance value was varied from 0.01 to 1.0.

It was found that better performance results were obtained using number of detectors = 700, number of clones = 5 and test tolerance = 0.6. Several test runs were conducted in which the training and the testing samples were generated randomly from the original breast cancer data set.

Similar to the ClonalG algorithm experiment, several runs were carried out using combinations of the key control parameters in the V-Detector algorithm to reach an acceptable balance. The typical values for these parameters were as follows:

- Radius of real-valued self samples: $r_s = 0.3$
- Estimated coverage rate: $c_o = 99.9\%$
- Significant level for hypothesis testing: $\alpha = 0.001$
- Maximum number of detectors: $T_{max} = 1000$

The statistics obtained in this case study are based on 100 repeated runs with the same control parameters. V-Detector algorithm could use different distance measures, however only Euclidean distance is used in the results reported here. Furthermore, the performance of the V-Detector algorithm was discovered to be sensitive to several parameters, such as the self radius.

Lastly, the same approach used previously in the CloanlG and V-Detectors tests was applied in the aiNet experiment. For the purposes of training, the aiNet parameters were set as follows:

- The suppression threshold = 0.1
- The pruning threshold = 1.0
- Number of best matching cells to be selected (n) = 4
- Clone number multiplier (N) = 20
- Percentile amount of clones to be re-selected = 10%
- The stopping criterion is a fixed number of generations = 30.

The comparison of performance results for the CloanlG, V-Detector, and aiNet algorithms are shown in Table 3.2. The table highlights the performance results for some test runs that include the accuracy, sensitivity, and specificity of the three algorithms. The outcome of this experiment has shown that the V-Detector algorithm has achieved a very promising classification performance results against the Wisconsin breast cancer dataset. Furthermore, the ClonalG algorithm also achieved very good performance results compared to the V-Detector algorithm. However, the results obtained from the aiNet algorithm still were unacceptably low, indicating the need for more improvements. Figure 3.4 depicts the overall performance of the ClonalG, V-Detector and aiNet algorithms respectively.

It is worth mentioning that the ClonalG algorithm is relatively low in complexity and has a small number of variables compared to other AIS techniques. Furthermore, aiNet has the capability of reducing redundancy and describing immune network structure, including data distribution and clustering; however, it also has some drawbacks, including its high number of user-defined parameters and its high computational cost per iteration O(m^2), with relation to the number of memory antibodies (m) (de Castro and von Zuben, 2001).

Table 3-2: Performance Results for the ClonalG, V-Detector and aiNet algorithms (Testing)

Round #	Accuracy			Sensitivity			Specificity		
	ClonalG	V-Detector	aiNet	ClonalG	V-Detector	aiNet	ClonalG	V-Detector	aiNet
1	0.7647	0.9412	0.5000	0.6111	0.8889	0.9444	0.9375	1.0000	0.0000
2	0.9118	0.9412	0.5294	0.8571	0.8571	1.0000	0.9500	1.0000	0.2000
3	0.8286	0.9429	0.4000	0.5385	0.8462	1.0000	1.0000	1.0000	0.0455
4	0.7714	0.8857	0.3429	0.4167	0.6667	1.0000	0.9565	1.0000	0.0000
5	0.8824	0.9412	0.3529	0.6667	0.8333	1.0000	1.0000	1.0000	0.0000
6	0.8529	0.9412	0.3235	0.5833	0.8333	0.9167	1.0000	1.0000	0.0000
7	0.8571	0.9143	0.3143	0.5455	0.7273	1.0000	1.0000	1.0000	0.0000
8	0.8571	0.9714	0.2857	0.7000	0.9000	1.0000	0.9200	1.0000	0.0000
9	0.9118	0.8824	0.3529	0.7000	0.7000	1.0000	1.0000	0.9583	0.0833
10	0.8857	0.9714	0.2571	0.5556	0.8889	1.0000	1.0000	1.0000	0.0000

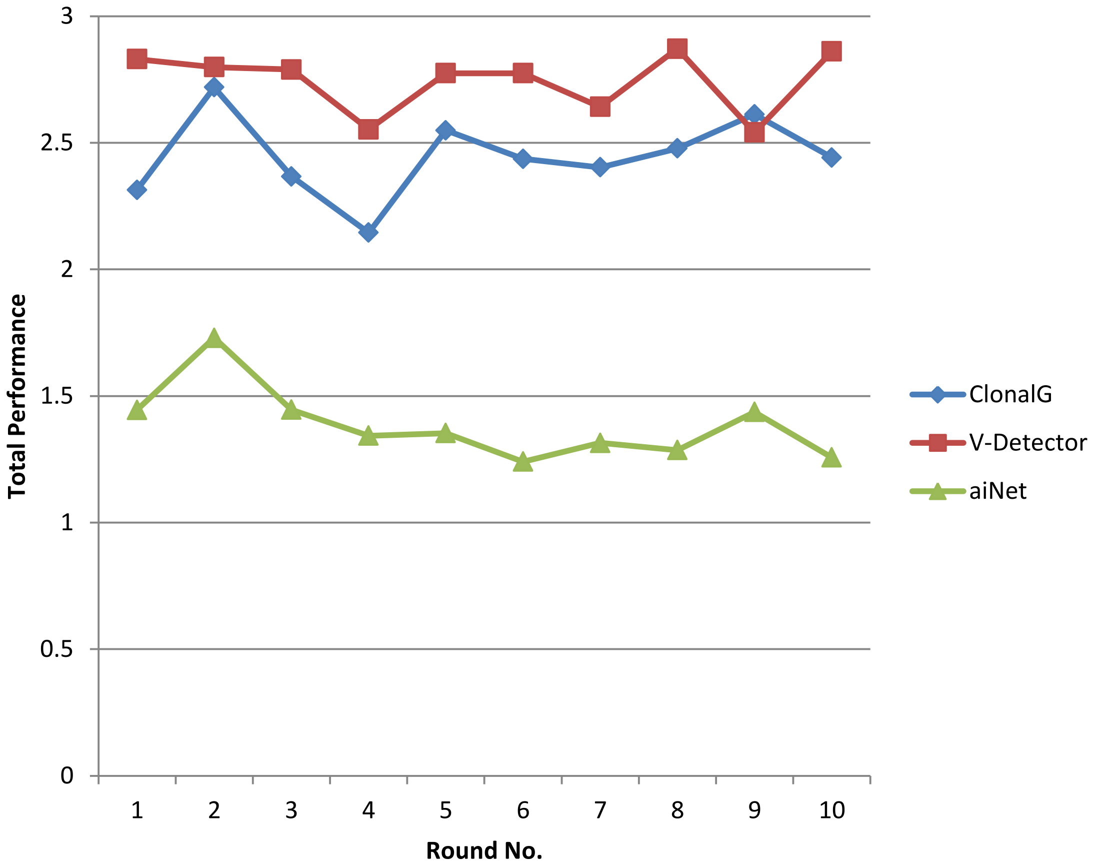


Figure 3-4: The overall performance of ClonalG, V-Detector and aiNet algorithms

3.5. Summary

An overview of the Artificial Immune Systems field, including a theoretical background on the main ideas and concepts of the biological immune system, is presented in this chapter. Three immune mechanisms are primarily used in the development of AIS methods. These include the immune network theory, clonal selection principles and negative selection mechanisms.

Some of the AIS models and techniques developed in the literature are discussed, with particular reference made to their application in the health sector. A case study was presented to test three of the most well-known AIS algorithms using a dataset specific to cancer. The case study clearly demonstrates how AIS approaches can be employed in dealing with real world problems in health and cancer research. The three experiments conducted to test the ClonalG, V-Detector and aiNet algorithms respectively against the cancer dataset yielded mixed results. A better performance result was achieved in the experiment especially with the V-Detector algorithm by detecting successfully the number and the clusters for the tested dataset. This outcome leads to the conclusion that some of the AIS techniques are found to be more suitable for cancer research than other AIS approaches.

The main concepts and various methods for combining classifiers in ensembles are discussed in the following chapter. Furthermore, a new biological based ensemble model is proposed and a method for measuring the confidence level for the base classifiers is suggested. A modification to the new AIS based ensemble is introduced also in the chapter using optimization techniques and compared to alternative combining techniques. A case study is then presented to test the performance of the proposed AIS based ensemble systems against real cancer dataset.

CHAPTER 4: Classifiers Ensembles Combining Methods

4.1. Introduction

Ensemble is a well established method for obtaining highly accurate classifiers by combining different algorithms. It has received increasing attention from researchers in the pattern recognition community. Ensemble learning can be generally defined as a machine learning system consisting of a set of individual learner models and a decision fusion strategy to combine their outputs, which produces single answers for given problems. The basic idea here is to combine a mixture of experts and to effectively make use of the results produced by each expert within the ensemble. The ensemble model should combine the strengths and weaknesses of the individual members and effectively apply the fusion strategy to reach enhanced results. By combining the results of each model, a final result with improved performance can be achieved.

Recently, the need for a combination of diverse classification algorithms has been widely recognized. Classifier combination is expected to outperform individual classifiers, whose performance is limited due to different factors such as the imperfection of feature extraction and learning algorithms, and the inadequacy of training data. It has been shown by many researchers that the classification accuracy of a combination of multiple classifiers is higher than that of the best individual classifier (Xu et al., 1992; Ho et al. 1994; Kittler et al., 1998).

In this chapter, the basic ideas of ensemble learning and important steps to implement such methods are introduced. Furthermore, a new biological based ensemble is proposed for a classification problem and a new technique to measure the confidence level for the base classifiers is suggested. A further enhancement to the AIS based ensemble is introduced by using particle swarm optimization approach. Finally, a case study is presented to evaluate the performance of the two suggested AIS based ensemble systems and their results are compared to other artificial immune systems algorithms.

4.2. Methods for Combining Classifiers in Ensemble

Many researchers have investigated the technique of combining multiple classifiers to produce a single classifier with an enhanced performance to solve complex recognition problems (Kittler and Roli, 2000; Kuncheva, 2004). Ensemble systems can be classified in a variety of ways. The basic categorization of ensemble systems is based on how the multiple classifiers are arranged. The serial and parallel architectures are the two basic categories in this regard. In the parallel expert architecture, a set of classifiers are arranged in parallel and the decisions of the various experts are combined in parallel by the fusion module. Alternatively, the serial architecture consists of a set of classifiers arranged in series. It is appropriate to deal with situations where an expert can be undecided on the input patterns and information is then passed to the next expert in the sequence. In this case, the experts should have a varying ability of generalizations. There are also other complicated combinations of these two architectures in the literature. However, parallel architectures are the most popular schemes.

According to the output information of member classifier, classifier combination can be categorized into three levels: abstract level, rank level and measurement level (Xu et al., 1992). In the abstract level, combination methods combine simple class labels. In the ranked level, combination methods combine ranked lists of class labels ordered according to the degree of membership of the input pattern. In the measurement level, combination methods combine values provided by individual classifiers as a measure of the degree of membership of the input pattern to each class (Xu et al., 1992). Among the three categories, the combination of classifiers at the measurement level is expected to be the most effective, since it uses all information available.

Xu et al. (1992) highlighted the two key challenges in the development of multiple classifier systems: the issue of how to create the individual classifiers that should be used for a specific application; and how to combine the results from different existing classifiers so that a better result can be obtained (a question facing various applications). Some of the commonly used approaches are introduced below for both challenges.

4.2.1. Creating Classifier Ensembles

The main objective in ensemble systems is to create many classifiers and combine their outputs, so that the combination improves upon the performance of a single classifier. To achieve this, there is clearly no gain in combining identical classifiers, and therefore the need is to have diverse classifiers which generalize differently and whose decision boundaries are adequately different from those of others.

Classifiers' diversity can be achieved in several ways. There are a number of training parameters which can be manipulated with this goal in mind: initial conditions, the training data, the typology of the classifiers, and the training algorithm (Duin, 2002; Wanas and Kamel, 2002). A set of different classifiers may be generated in the following ways:

- *Varying the set of initial random weights:* A set of classifiers can be created by varying the initial random weights from which each classifier is trained, while maintaining the same training data.
- *Varying the architecture:* A set of classifiers can be created by varying the architecture, while using the same training data.
- *Varying the algorithm employed:* The algorithm used to train the classifiers can be varied, while holding the data constant.
- *Varying the data:* The diversity in the classifiers is typically achieved by using a different training data set for each classifier, which then allows each classifier to generate different decision boundaries.

The most frequent methods used for the creation of ensembles are those which involve altering the training data. This can be done using several ways, including sampling data, disjoint training sets, boosting and adaptive re-sampling, different data sources, and pre-processing, or a combination of these techniques (Wanas and Kamel, 2002). In this context, the two most common algorithms to create an ensemble by training the classifiers on different samples of the training data are bagging and boosting (Bootstrapped Aggregating). The Bagging algorithm (Breiman, 1996) randomly samples the data set with replacement to create different training sets for each ensemble classifier. In AdaBoost algorithm (Freund and Schapire, 1996), which is the commonly used version of boosting, the training sets are adaptively re-sampled, so that the weights in the re-sampling are increased for those cases which are most often misclassified.

4.2.2. Combining Classifier in Ensembles

After creating classifiers, the second key component in building any ensemble system is employing a strategy for combining classifiers. Several approaches have been proposed for combining multiple classifiers (Battiti and Colla, 1994; Jacobs R., 1995). Kuncheva investigated a variety of combination techniques in detail, including a discussion of voting approaches as one of the most commonly used techniques in the literature (Kuncheva, 2004).

One of the popular voting approaches is the majority vote (Xu et al., 1992; Battiti and Colla, 1994; Lam and Suen, 1995; Ji and Ma, 1997; Waterhouse and Cook, 1997; Kuncheva, 2004). Other voting schemes include the minimum, maximum, median (Kuncheva, 2002), average (Munro and Parmanto, 1997; Taniguchi and Tresp, 1997), and product (Tax et al., 2000) methods. The weighted average approach (Jacobs, 1995; Hashem, 1997; Heskes, 1997; Merz and Pazzani, 1997; Kuncheva, 2004) introduces weights for the various classifiers used prior to averaging. The weights determine the relative importance of the classifiers outputs on the average. In the Behaviour-Knowledge Space (BKS) approach (Woods, 1997), the best classifier in some region of the input space is selected, the selection is based on its prediction accuracy. Other classifiers combination techniques include the rank-based methods such as the Borda count (Ho et al.1994), the Bayes approach (Xu et al., 1992; Lam and Suen, 1995), the Dempster-Shafer theory (Xu et al., 1992; Rogova, 1994; Denouex, 1995; Le Hegarat-Mascle et al., 1998), the fuzzy integral (Tahani and Keller, 1990; Grabisch, 1994; Cho and Kim, 1995; Grabisch, 1995; Gader et al 1996; Mirhosseini et al. 1998), fuzzy connectives (Kuncheva, 1997), fuzzy templates (Kuncheva et al. 1998), and probabilistic schemes (Kang et al., 1997; Kittler et al., 1997a; Kittler et al., 1997b; Kittler et al., 1998).

The following subsections outline the majority vote and weighted average approaches commonly used in combining various classifiers.

a) Majority Vote

In majority vote, each classifier provides a vote to a class to which the input pattern belongs. The correct class is the one most often chosen by the classifiers. Three consensus patterns for majority vote were introduced by Kuncheva (2004):

1. Unanimity - 100% agree on choice to be returned
2. Simple Majority - 50% + 1 agree on choice to be returned
3. Plurality - Choice with the most votes is returned

Assume that the label outputs of the classifiers are given as c-dimensional binary vectors $[d_{i,1}, d_{i,2}, \ldots, d_{i,c}]^T \in \{0, 1\}^c$, $i = 1, \ldots, L$, where $d_{i,j} = 1$ if D_i labels x in class j, and 0 otherwise. The majority vote results in an ensemble decision for class k if

$$\sum_{i=1}^{L} d_{i,k} = \max_{j=1}^{c} \sum_{i=1}^{L} d_{i,j} \qquad (4.1)$$

In majority voting, ties are resolved arbitrarily. The plurality vote is often called the majority vote, and it is the same as the simple majority when there are two classes ($c = 2$) (Kuncheva, 2004).

b) Weighted Average

The weighted average approach is similar to the average combining strategy. However, the weighted average approach introduces weights to the outputs of the different classifiers prior to averaging. The average approach calculates the support **μ** (x) for class j by:

$$\mu_j\,(\mathrm{x}) = \frac{1}{L} \sum_{i=1}^{L} d_{i,j}(x) \qquad (4.2)$$

where L represents the number of classifiers and $d_{i,j}(x)$ represents the output of the ith classifier for the jth class for the input x. In the weighted average approach, the overall accuracy of each classifier is used for generating the weighting parameters, where, a classifier is assigned a higher weight if the resulting classification accuracy is high. On the other hand, if the overall accuracy of a classifier is low, the classifier is assigned a lower weight.

If w_i is the weight assigned to the ith classifier, the support **μ** (x) for class j can be calculated by (Kuncheva, 2004):

$$\mu_{\boldsymbol{j}}\ (\mathrm{x}) = \frac{1}{L}\sum_{i=1}^{L} w_i d_{i,j}(x) \qquad (4.3)$$

where

$$\sum_{i=1}^{L} w_i = 1 \qquad (4.4)$$

4.2.3. Architectures for Combining Classifiers

Different multiple classifiers combining methods exist (as mentioned above) with the aim of achieving optimum performance accuracy. It has been argued that in many domains an ensemble of classifiers outperforms any of its single components. Three possible architectures can be used for combining different single classifiers: cascaded, parallel and hierarchical (Lu, 1996). In a cascaded system, the classification results generated by a classifier are often used to direct the classification processes of successive classifiers. On the other hand, the main problem in this method is that errors made by previous classifiers are not recoverable by the successive classifiers (Lu, 1996). On the other hand, the classification result is generated independently in a parallel system from different sources, then a decision strategy is used to form the final decision. If the decision process is well designed, the overall system may reach peak performance (Lu, 1996). Finally, in a hierarchical system, the control strategy is a combination of cascaded and parallel processing. Figure 4.1 illustrates these different classifiers combining configurations.

In general, the architectures depicted in Figure 4.1 work based on two architectural methodologies: multi-expert and multistage (Alpaydin, 1998). In multi-expert method, the classifiers work in parallel where they all trained on all patterns and then give their decisions in which a separate combiner computes the final decision using a decision combining strategy. Examples of this method are voting, mixture of experts and stacked generalization. Alternatively, the multi-stage method uses a serial approach where the next classifier is trained (consulted) only for patterns rejected by the preceding classifier(s); examples include boosting and cascading.

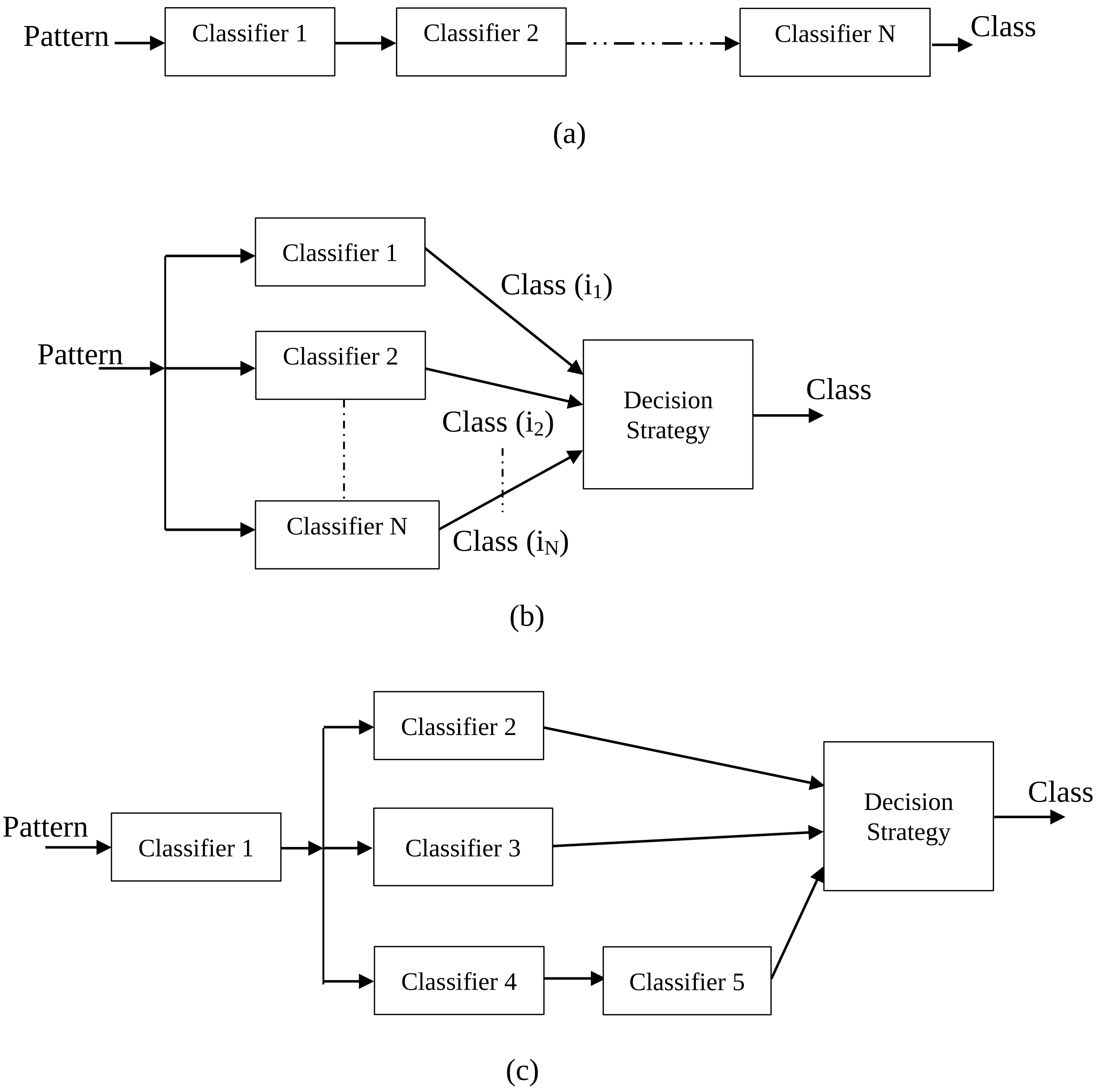


Figure 4-1: Architectures of multiple classifiers system: (a) cascading, (b) parallel, and (c) hierarchy

4.3. AIS Based Ensemble System

Various approaches presented in the literature for combining multiple classifiers have been reviewed by previous sections. A new artificial immune system based ensemble is proposed in this section, in which three of the popular AIS algorithms are combined in an ensemble for the classification application, using the majority voting and weighted average combination techniques (presented in section 4.2.2). The three AIS algorithms are: ClonalG, V-Detector, and aiNet models (discussed in Chapter 3). Figure 4.2 highlights the block diagram of the new proposed ensemble system.

The proposed AIS architecture introduces a new method for calculating the confidence level on the predicted output $\boldsymbol{O} = [\boldsymbol{O_1}, \boldsymbol{O_2}, \boldsymbol{O_3}]$ for the base classifiers by extracting some features during the training stage and accordingly a confidence value will be assigned to each predicted output. The confidence levels for the base classifiers are represented by vector $\boldsymbol{C} = [\boldsymbol{C_1}, \boldsymbol{C_2}, \boldsymbol{C_3}]$. Then, the predicted output along with the confidence measure will be used to perform the aggregation procedure for achieving the final decision.

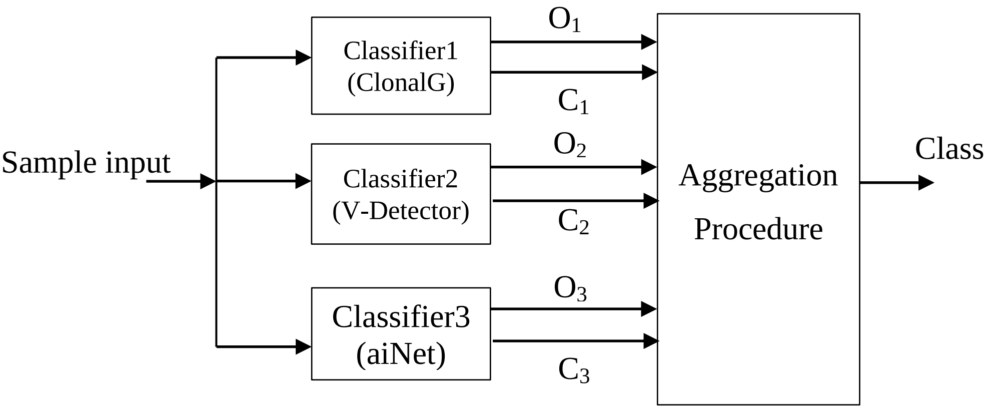


Figure 4-2: New AIS based ensemble system

4.3.1. Method for Measuring Confidence Level

Improving the confidence level of making right decisions is one of the main objectives of ensemble systems. This can be achieved by weighing various opinions obtained from the base classifiers and combining them through some aggregation procedures to reach a final decision. The major focus here is on assigning the weights for the base classifier on the basis of its competence in order to achieve the maximum performance for the ensemble system.

This work introduces a new method to measure the confidence level of the AIS classifier's predicted output in the ensemble. Initially, the classifiers of the ensemble are trained independently on the training data set to generate sets of detectors. The detector set associated with each base classifier is used then to extract the output class and the confidence level for a given input sample x_i. The outputs of the classifiers are interpreted as a decision output O_i and a confidence value C_i which will be used finally by the aggregation scheme. An illustrative example of the proposed method to measure the confidence level values is shown in Figure 4.3.

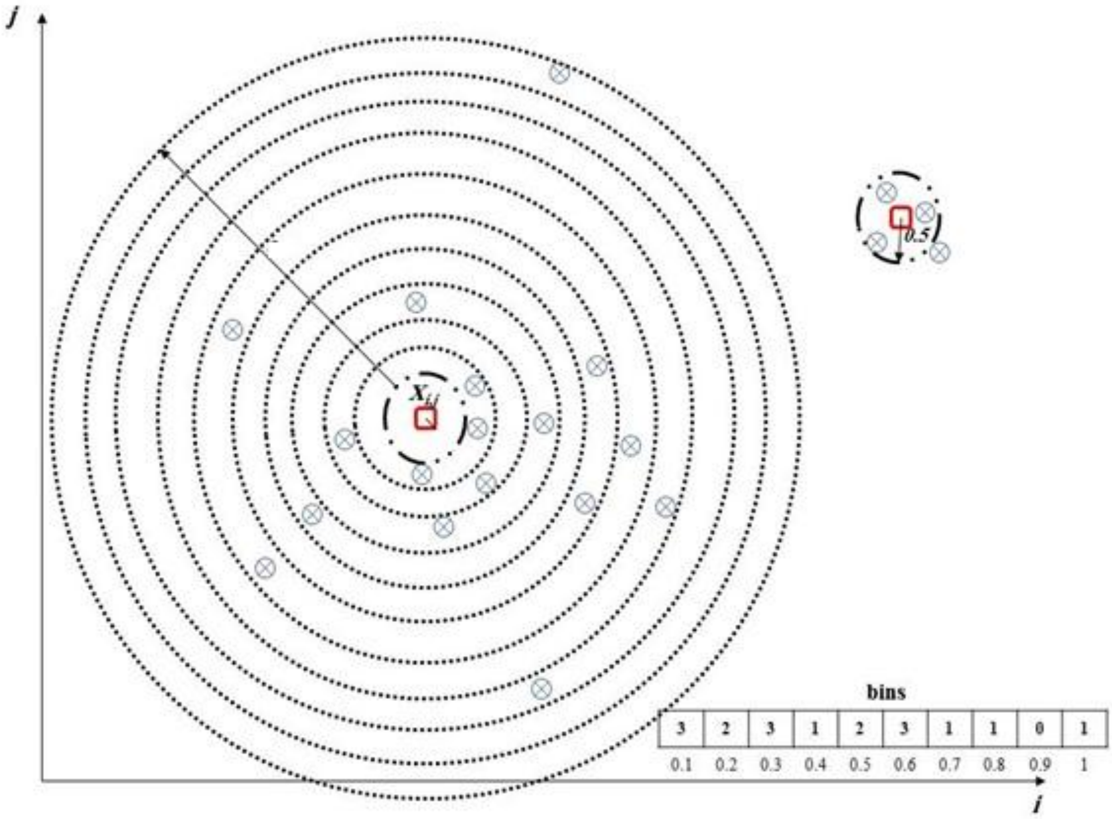
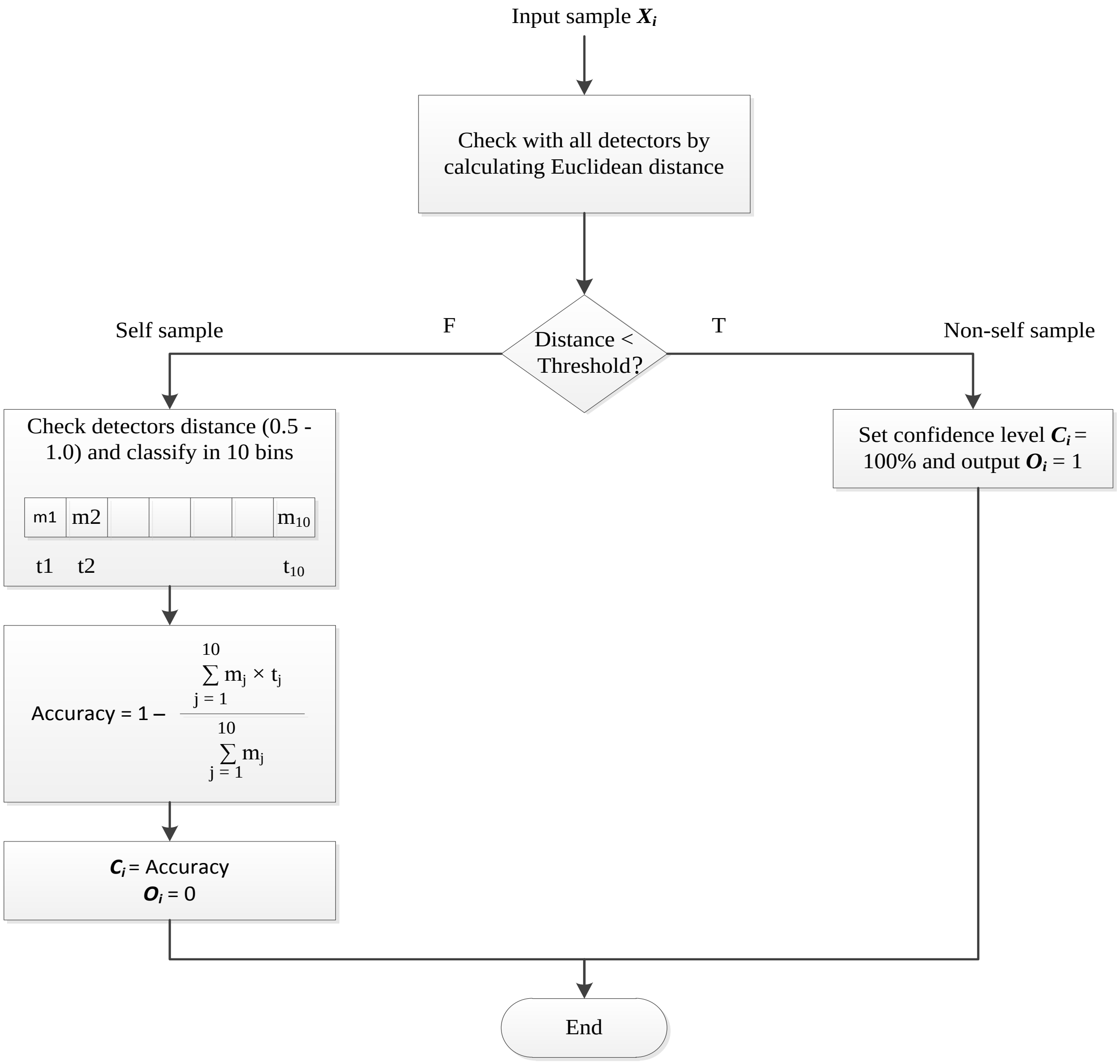


Figure 4-3: An example for calculating the confidence level for the input sample $x_{i,j}$

Figure 4.3 shows an example of using the ClonalG algorithm as the base classifier, where the new sample $x_{i,j}$ will be tested against the detectors set generated during the training phase of the algorithm. The suggested method for measuring the confidence level works in two main steps. The first step in this method is to find the predicted class for the input sample by testing it against all the detectors using the matching rules such as Euclidean distance and a preset threshold value. If the input sample $x_{i,j}$ matches any of the detectors, then it is considered as a non-self sample where the predicted class O_i will be set equal to one, and accordingly the confidence level C_i will be set to a high value (100%). Otherwise, for self sample, where the predicted class will be set equal to zero and the confidence level will be calculated in the next step. In the second step, the matching region around the input sample will be expanded to search for more detectors near this sample. The expanded region will be divided into 10 bins, then all of the detectors discovered within the region will be classified in bins. Let $\boldsymbol{M} = [m_1, m_2, \ldots, m_i]$ represent the number of detectors discovered in each bin, where i is the total number of bins and $\boldsymbol{T} = [t_1, t_2, \ldots, t_i]$ represent the bin index, then the confidence level of the predicted output will be calculated by:

$$Confidence\ Level\ (C_i) = 1 - \frac{\sum_{i=1}^{10} m_i \times t_i}{\sum_{i=1}^{10} m_i} \qquad (4.5)$$

Figure 4.4 (below) outlines the flowchart of the proposed method for calculating the confidence level.



Input sample X_i
Check with all detectors by calculating Euclidean distance
Distance < Threshold?
F
T
Self sample
Non-self sample
Check detectors distance (0.5 - 1.0) and classify in 10 bins
m1
m2
m10
t1
t2
t10
Accuracy = 1 − $\frac{\sum_{j=1}^{10} m_j \times t_j}{\sum_{j=1}^{10} m_j}$
C_i = Accuracy
O_i = 0
Set confidence level C_i = 100% and output O_i = 1
End

Figure 4-4: Flowchart of the proposed method for calculating the confidence level

4.4. AIS Based Ensemble System with PSO Optimizer

Weighing the predicted outputs of base classifiers is an important task in ensemble methods to avoid poor combination performance. Hence, the drivers outputs of classifiers are transformed into unified measures that represent the confidence level of decisions made respectively in order to achieve enhanced performance. This section introduces an optimized version of the AIS based ensemble architecture presented earlier by adjusting the weights assigned to the base classifiers using particle swarm optimization technique. A brief introduction of the PSO algorithm and the proposed AIS based ensemble with PSO optimizer is discussed in the following subsections.

4.4.1. Particle Swarm Optimization Algorithm

Particle Swarm Optimization is a swarm intelligence global optimization technique introduced by Eberhart and Kennedy that mimics the social behaviours of birds flocking or fish schooling (Eberhart and Kennedy, 1995; Kennedy and Eberhart, 1995; Kennedy and Eberhart, 1997). In PSO, a member in the swarm is called a particle that represents a potential solution in the search space. Each particle has a position and velocity. The PSO algorithm adaptively updates the velocities and positions of the members of the swarm by learning from the good experiences. This is a stochastic adaptation process that depends on both the memory of each individual as well as the knowledge gained by the whole swarm.

While flying through the search space, the particle remembers the best position it has seen, and communicates this position to the other particles. Accordingly, the members of the swarm will adjust their own positions and velocity based on this information. The interconnection between particles can be common to the whole population, or be divided into local neighbourhoods of particles.

During the operation of the PSO algorithm, each particle remembers the best position or best candidate solution it has achieved thus far, referred to as *pbest*. The difference between the best candidate solution *pbest* found so far and the particle current position is added to the current velocity, causing the redirection of position around that point.

Furthermore, the PSO algorithm keeps the best global position or best global candidate solution achieved among all particles in the swarm, called *gbest*. The difference between the global best position *gbest* and the individual's current position is also added to its velocity, causing the particle to move to the best region the swarm has found so far.

The new velocity for each particle in the swarm is calculated based on its previous velocity and the two best positions (*pbest* and *gbest*) as shown in the equation below:

$$v(t+1) = wv(t) + \varphi_1 r_1 (pbest(t) - x(t)) + \varphi_2 r_2 (gbest(t) - x(t)) \tag{4.6}$$

where *v*(*t*) and *x*(*t*) represent the particle previous velocity and position respectively, *w* is the inertia weight, φ_1 and φ_2 are random positive constants called cognitive and social parameter, which weigh the influence of the two different swarm memories, and r_1 and r_2 are random numbers between 0 and 1. Once the velocity for each particle is calculated, the new position for each particle is found by applying the new velocity to the particle's previous position according to the equation below:

$$x(t+1) = x(t) + v(t+1) \tag{4.7}$$

Figure 4.5 shows the movement of the particle in the search space towards the optimal solution, where it is initialized with random positions *x*(t) and velocities *v*(t) and a fitness function is evaluated using the coordinates of particle position as input values. Algorithm 4.1 outlines the main steps of the PSO algorithm.

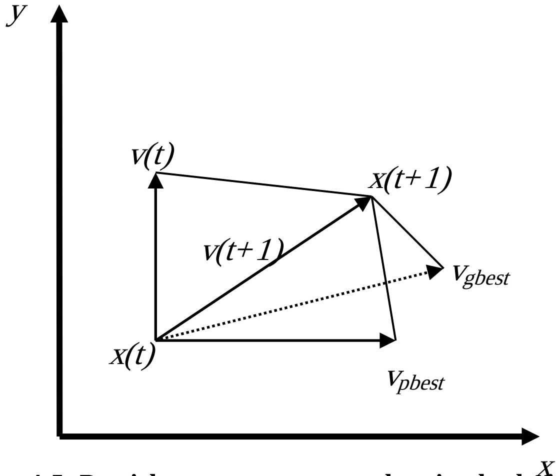


Figure 4-5: Particle movements toward optimal solution (Heo et al., 2006)

Algorithm 4.1: PSO algorithm adopted from (Poli, 2007)

```
1. Initialize a population array of particles with random positions and velocities on
   D dimensions in the problem space.
2. loop
3.   For each particle, evaluate the desired optimization fitness function in D
     variables.
4.   Compare particle's fitness evaluation with its personal best fitness $pbest^i$. If
     current value is better than $pbest^i$, then set $pbest^i$ equal to the current value, and
     $y^i$ equal to the current location $x^i$ in D-dimensional space.
5.   Identify the particle in the neighborhood with the best success so far, and
     assign its position to the variable $\hat{y}$.
6.   Change the velocity and position of the particle according to the following
     equations:
         $v_{t+1}^i = wv_t^i + \emptyset_1 \otimes (y^i - x_t^i) + \emptyset_2 \otimes (\hat{y} - x_t^i)$    (1)
         $x_{t+1}^i = x_t^i + v_{t+1}^i$    (2)
7.   If a criterion is met, exit loop.
8. end loop
```

Note: $\emptyset_i$ represents a vector of random numbers uniformly distributed in $[0, c_i]$ and $\otimes$ is component-wise multiplication.

4.4.2. Architecture

An enhanced version of the AIS based ensemble system is proposed in this section using the idea of particle swarm optimization technique in order to optimize the confidence levels (weights) of the base classifiers, which then better evaluate the competence of each member to improve the overall ensemble performance. A block diagram of the optimized AIS ensemble architecture is shown in Figure 4.6.

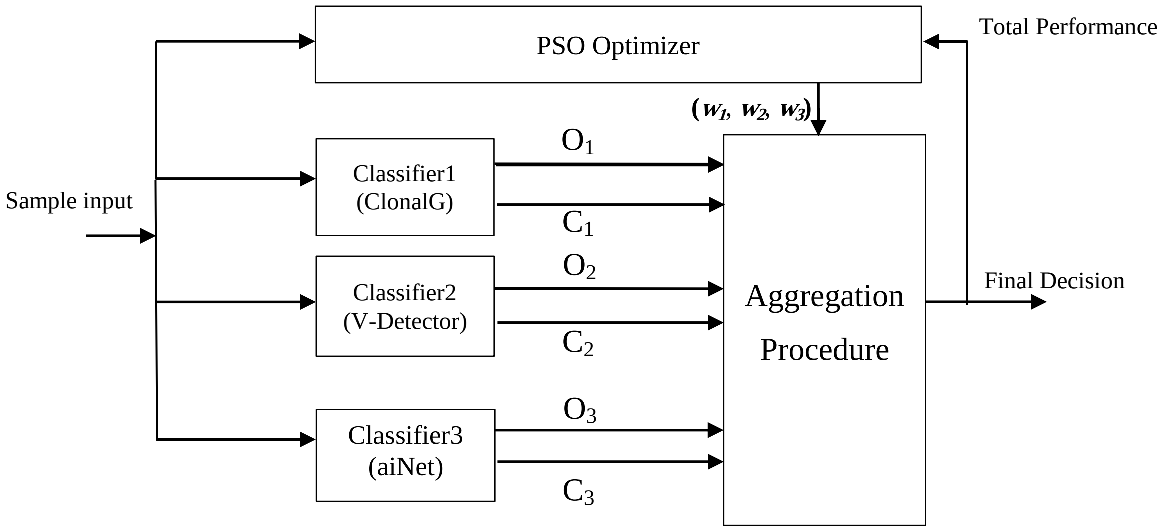


Figure 4-6: AIS based ensemble system with PSO Optimizer

In this context, the various members of the ensemble are trained on the whole training dataset in order to make a decision on an input sample. The predicted outputs of the classifiers are represented by vector [O_1, O_2, O_3] as shown in the figure. Accordingly, each one of the base classifiers will be assigned a confidence value C_i that could characterize the competence of the classifier in the ensemble based on its classification performance. These confidence levels are then optimized using PSO algorithm in order to achieve optimal weight values represented by vector [w_1, w_2, w_3]. The process of optimization continues till a predefined threshold is reached. Once the optimal weight values are achieved by PSO, the outputs of these classifiers are combined to conclude the final decision using one of the aggregation procedures such as weighted average.

4.5. Case Study: Breast Cancer

Three experiments were conducted in order to evaluate the performance of the proposed AIS ensemble system. The dataset used in these experiments is the Wisconsin breast cancer (presented in the previous chapter). The aims of these experiments are to show that the new AIS ensemble model outperforms the base AIS classifiers and to compare it with other combining methods such as majority voting and weighted average.

In the first experiment, the majority voting method was used as an aggregation procedure for combining the decisions that resulted from each algorithm. Alternately, the weighted average combination method was used in the second experiment. The proposed AIS ensemble with PSO optimizer was implemented in the last experiment to test it against the cancer dataset. The weighted average aggregation procedure was used to combine the predicted outputs resulted from the base classifiers along with the PSO optimizer.

Initially, the dataset was normalised to unity before being fed to the algorithms and splitted into 95% for training set and 5% test set. For each experiment, 100 runs were performed, in which different training and testing samples were chosen randomly in each round. The accuracy (equ. 3.4), sensitivity (equ. 3.5), specificity (equ. 3.6) and total (equ. 3.7) performance measures are used to compare between the different methods.

In the third experiment, the number of particles used in the optimization algorithm varied between 20 and 100 in each round, to test PSO optimizer capability for achieving an optimal solution. It was noticed that the computational time for achieving optimal weight values while running the PSO optimizer increases with the increase in number of particles used. However, increasing the number of particles has no direct impact in reaching the optimal solution. Table 4.1 highlights the detailed performance results for the top ten samples obtained from this experiment. The presented results show the impact of variation on the number of particles to the different performance measures. In conclusion, the proposed AIS based ensemble with PSO optimizer achieved better results than the base classifiers. However, there is still a room for improvement to the model and the ensemble performance can be further optimized.

Table 4-1: Performance Results for the WA_PSO Ensemble System

Sample #	# of Particles	WA_PSO Ensemble						
		Accuracy	Sensitivity	Specificity	Total	*Gbest*		
1	**20**	0.941	0.889	1.000	2.830	0.540	0.659	0.819
	30	0.958	0.875	1.000	2.833	0.670	0.759	0.232
	40	0.941	0.889	1.000	2.830	0.618	1.227	0.223
	50	0.941	0.889	1.000	2.830	0.149	1.010	0.042
	70	0.941	0.889	1.000	2.830	0.626	0.746	0.344

Sample #	# of Particles	WA_PSO Ensemble						
		Accuracy	Sensitivity	Specificity	Total	*Gbest*		
	100	0.958	0.875	1.000	2.833	0.443	0.281	0.069
2	**20**	0.971	0.929	1.000	2.899	0.726	0.779	0.572
	30	0.971	0.929	1.000	2.899	0.670	0.685	0.593
	40	0.971	0.929	1.000	2.899	0.316	0.364	1.009
	50	0.971	0.929	1.000	2.899	1.233	1.293	0.667
	70	0.971	0.929	1.000	2.899	0.373	0.449	1.462
	100	0.971	0.929	1.000	2.899	0.412	0.420	0.984
3	**20**	0.971	0.923	1.000	2.895	1.345	1.421	1.452
	30	0.971	0.923	1.000	2.895	0.828	0.969	0.073
	40	0.971	0.923	1.000	2.895	1.130	1.155	0.359
	50	0.971	0.923	1.000	2.895	0.997	1.199	0.297
	70	0.971	0.923	1.000	2.895	0.305	0.306	0.351
	100	0.971	0.923	1.000	2.895	0.897	1.078	0.221
4	**20**	0.914	0.750	1.000	2.664	1.198	1.251	1.136
	30	0.914	0.750	1.000	2.664	1.220	1.229	0.882
	40	0.914	0.750	1.000	2.664	0.841	0.870	0.721
	50	0.914	0.750	1.000	2.664	1.013	1.022	0.013
	70	0.914	0.750	1.000	2.664	1.021	1.065	1.100
	100	0.914	0.750	1.000	2.664	0.162	0.163	0.131
5	**20**	0.971	0.917	1.000	2.887	0.995	1.072	0.772
	30	0.971	0.917	1.000	2.887	0.265	0.265	1.215
	40	0.971	0.917	1.000	2.887	0.079	0.081	0.292
	50	0.971	0.917	1.000	2.887	1.095	1.178	1.057
	70	0.971	0.917	1.000	2.887	1.310	1.366	0.545
	100	0.971	0.917	1.000	2.887	1.260	1.262	0.178
6	**20**	0.971	0.917	1.000	2.887	0.192	0.198	0.377
	30	0.971	0.917	1.000	2.887	1.407	1.443	0.534
	40	0.971	0.917	1.000	2.887	1.053	1.097	0.301

Sample #	# of Particles	WA_PSO Ensemble						
		Accuracy	Sensitivity	Specificity	Total	*Gbest*		
	50	0.971	0.917	1.000	2.887	0.299	0.303	0.356
	70	0.960	1.000	0.955	2.915	1.138	1.168	0.490
	100	0.971	0.917	1.000	2.887	0.696	0.732	1.145
7	**20**	0.966	0.800	1.000	2.766	0.237	0.342	0.423
	30	0.914	0.727	1.000	2.642	0.594	1.048	0.151
	40	0.914	0.727	1.000	2.642	0.543	1.499	0.647
	50	0.966	0.800	1.000	2.766	1.070	1.089	0.107
	70	0.966	0.800	1.000	2.766	0.307	0.357	0.253
	100	0.966	0.800	1.000	2.766	0.691	0.862	0.484
8	**20**	0.971	0.900	1.000	2.871	0.971	1.266	0.833
	30	0.971	0.900	1.000	2.871	0.803	1.251	1.077
	40	0.971	0.900	1.000	2.871	0.128	1.380	0.120
	50	0.971	0.900	1.000	2.871	0.590	1.276	0.421
	70	0.971	0.900	1.000	2.871	0.429	1.018	0.258
	100	0.971	0.900	1.000	2.871	1.396	1.489	0.707
9	**20**	0.897	0.750	0.920	2.567	0.975	0.484	0.172
	30	0.963	0.750	1.000	2.713	0.840	1.164	1.826
	40	0.963	0.750	1.000	2.713	0.501	0.570	0.044
	50	0.963	0.750	1.000	2.713	0.681	0.694	0.117
	70	0.882	0.700	0.958	2.541	0.503	1.222	0.327
	100	0.882	0.700	0.958	2.541	0.393	0.647	1.415
10	**20**	0.971	0.889	1.000	2.860	0.268	0.373	0.774
	30	0.971	0.889	1.000	2.860	0.308	0.896	0.734
	40	0.971	0.889	1.000	2.860	0.730	1.477	0.709
	50	0.971	0.889	1.000	2.860	0.280	1.420	1.162
	70	0.971	0.889	1.000	2.860	0.833	1.348	1.163
	100	0.971	0.889	1.000	2.860	0.350	0.693	1.119

The experiments conducted in this chapter are a continuation of the case study performed in Chapter 3. It was shown in the case study that the V-Detector algorithm achieved the higher performance results. The performance results achieved by the individual classifiers are used in this case study in order to compare it with the results obtained from various ensemble methods. The top ten results were selected from all runs to present the overall performance of the system. Table 4.2 highlights the performance results of the three experiments of this case study. Furthermore, Table 4.3 and Figure 4.7 illustrate the total performance of the majority voting, weighted average, and the optimized AIS ensembles against the base AIS algorithms. In Figure 4.8, the ROC plots were performed for the various ensemble methods used in the three experiments to visualize their classification performance.

Table 4-2: Performance Results for the MV_E, WA_E and WA_PSO AIS ensembles

Round #	# Particles	Accuracy			Sensitivity			Specificity		
		MV_E	WA_E	WA_PSO	MV_E	WA_E	WA_PSO	MV_E	WA_E	WA_PSO
1	20	0.971	0.971	0.941	0.944	0.944	0.889	1.000	1.000	1.000
2	20	0.971	0.971	0.971	0.929	0.929	0.929	1.000	1.000	1.000
3	20	0.971	0.971	0.971	0.923	0.923	0.923	1.000	1.000	1.000
4	20	0.971	0.971	0.914	0.917	0.917	0.750	1.000	1.000	1.000
5	20	0.971	0.971	0.971	0.917	0.917	0.917	1.000	1.000	1.000
6	20	0.971	0.971	0.971	0.917	0.917	0.917	1.000	1.000	1.000
7	30	0.971	0.971	0.914	0.909	0.909	0.727	1.000	1.000	1.000
8	20	0.971	0.971	0.971	0.900	0.900	0.900	1.000	1.000	1.000
9	70	0.971	0.941	0.882	0.900	0.900	0.700	1.000	0.958	0.958
10	20	0.971	0.971	0.971	0.889	0.889	0.889	1.000	1.000	1.000

Table 4-3: The overall performance of the MV_E, WA_E and WA_PSO AIS ensembles

Round #	**Total Performance**					
	ClonalG	**V-Detector**	**aiNet**	**MV_E**	**WA_E**	**WA_PSO**
1	2.313	2.830	1.444	2.915	2.915	2.830
2	2.719	2.798	1.729	2.899	2.899	2.899
3	2.367	2.789	1.446	2.895	2.895	2.895
4	2.145	2.552	1.343	2.888	2.888	2.664
5	2.549	2.775	1.353	2.887	2.887	2.887
6	2.436	2.775	1.240	2.887	2.887	2.887
7	2.403	2.642	1.314	2.881	2.881	2.642
8	2.477	2.871	1.286	2.871	2.871	2.871
9	2.612	2.541	1.436	2.871	2.800	2.541
10	2.441	2.860	1.257	2.860	2.860	2.860
Average	**2.446**	**2.743**	**1.385**	**2.885**	**2.878**	**2.798**

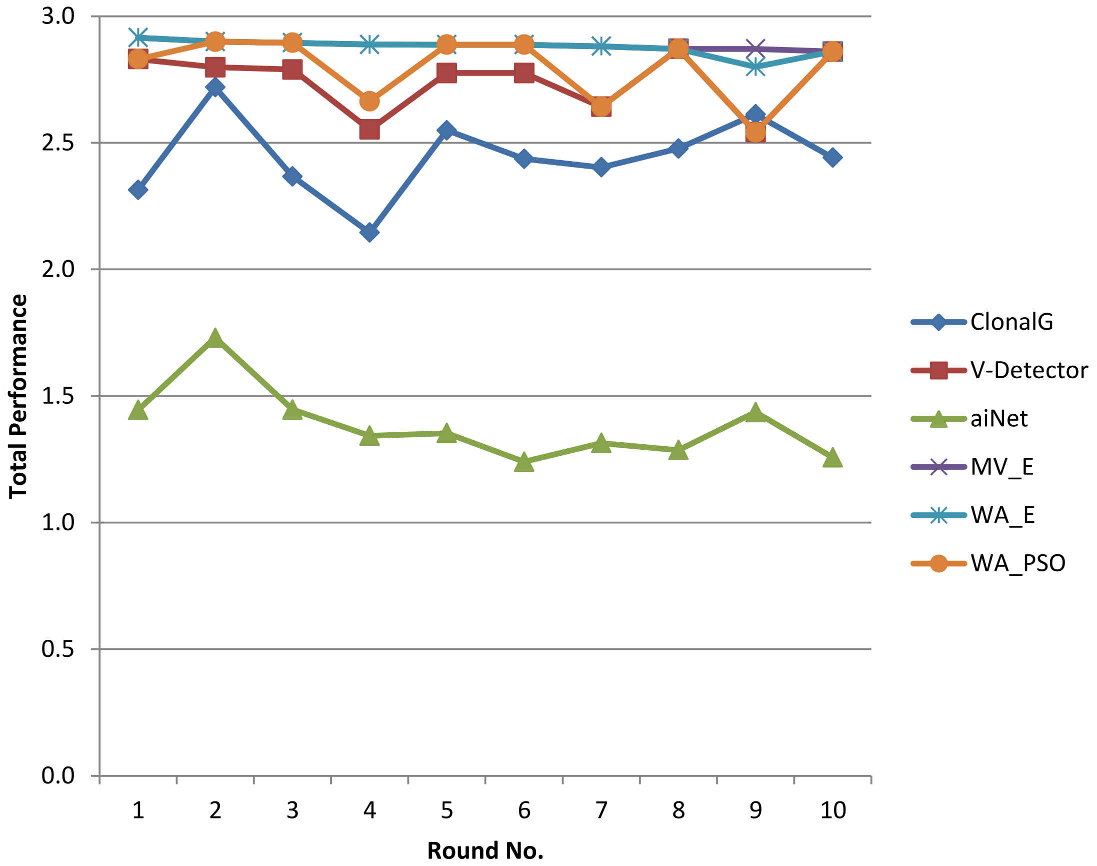


Figure 4-7: The overall performance of the MV_E, WA_E and WA_PSO AIS ensembles

The experimental results demonstrate that the proposed AIS based ensemble systems achieved the best performance in the three experiments and in all the test runs. More specifically, the classification performance of the AIS based ensembles using the majority voting, weighted average and optimized weighted average combining techniques outperform individual AIS classifiers.

However, beside the fact that the AIS ensemble with PSO optimizer model achieved better results than the base classifiers, it can be noted that the results of the third experiment that its average performance was slightly less than the other two classical combining techniques.

Finally, the results obtained from all the experiments show a slight improvement on the overall classification performance, and that the AIS ensemble system can be further enhanced. An attempt to design AIS based ensemble with the objective of reaching an optimal performance is presented in the following chapter.

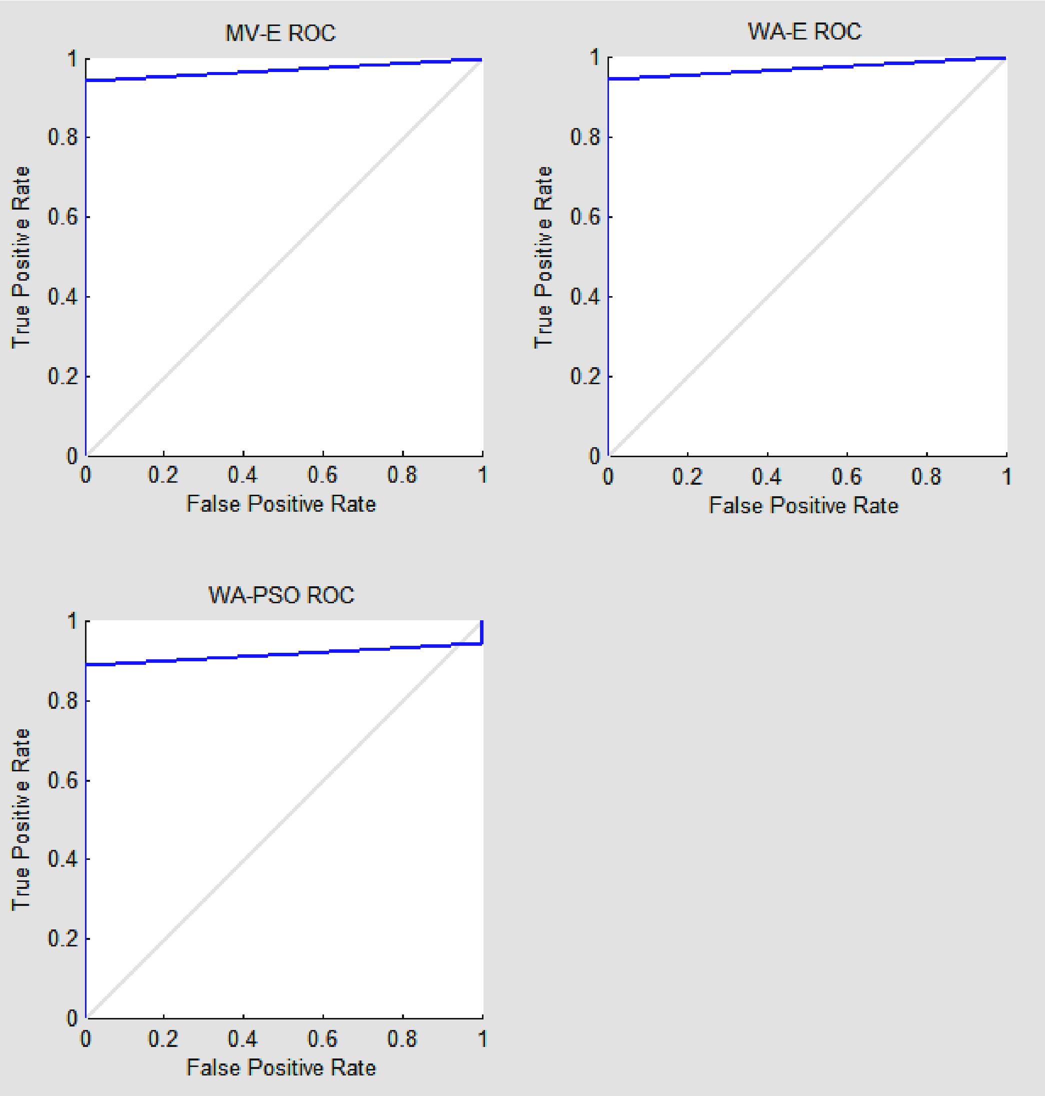


Figure 4-8: The ROC plots for the MV, WA and WA_PSO AIS ensembles

4.6. Summary

Various methods and techniques for combining multiple classifiers were presented and discussed in this chapter. In addition, the chapter highlighted the key challenges in the development of multiple classifier systems. Different strategies and architectures have been introduced to address these challenges and the relevant work in the literature has been presented.

In this chapter, a new artificial immune systems based ensemble combining three well known AIS algorithms was proposed. The majority voting and weighted average combining techniques were used to form the aggregation procedure of the suggested AIS ensemble model.

Additionally, a PSO optimizer technique was introduced to further enhance the performance of the proposed AIS based ensemble using a particle swarm optimization method.

Finally, a case study was conducted to test the performance of the proposed ensemble model against a real cancer data set for the classification problem. The experimental results show that the AIS ensemble systems achieved the best classification performance results. However, the suggested AIS ensemble model can be enhanced further to achieve even better results.

The following chapter introduces a new adaptive learning AIS based ensemble model and compares it to alternative combining techniques. The adaptive learning feature of the new model using a neuro-fuzzy based detector model is discussed in details.

CHAPTER 5: A New Adaptive Learning Artificial Immune Systems Based Ensemble

5.1.Introduction

This Chapter introduces a new AIS based ensemble with an adaptive learning capability to further enhance the accuracy of the proposed ensemble. A detector based architecture as a main modification to the AIS ensemble is introduced to further improve the system performance. The proposed architecture focuses on making the decision fusion a more adaptive process. Finally, a new adaptive learning AIS based ensemble is proposed, integrating various soft computing and optimization techniques to achieve optimal performance. The following sections discuss several variations of the adaptive learning ensemble architecture, including some theoretical background on the main components used.

The motivation for exploring the multiple classifiers combination strategies is to improve the overall system performance and robustness. In the previous chapter, the author discussed various approaches presented in the literature for combining multiple classifiers in an ensemble. These pioneering methods paved the way for developing systems with a level of performance adequate to be deployed in real-world applications. In addition, the previous chapter presented a case study which showed that by combining various artificial immune systems algorithms, the overall all performance of the AIS based ensemble can be further improved. However, more improvements on the proposed model can be achieved by using other soft computing techniques such as neuro-fuzzy systems.

5.2.Detector Based Ensemble

The main goal of the proposed detector based architecture is to provide a dynamic process for enhancing the decision fusion of the different classifiers and consequently to enhance the performance of the AIS based ensemble. This can be achieved by introducing a detector component to extract features from the base classifiers to perform the aggregation procedure efficiently. The detector allows the ensemble model to recognize and learn the changes in the input and their impact on the performance of the individual classifiers, and adjust accordingly the final decision fusion of the ensemble. This learning process provides the ensemble system with the required flexibility to adapt to changes in the input and output of the classifiers in order to improve the overall classification performance.

With this objective in mind, two main factors were taken into consideration while designing the detector based architecture. Firstly, the design of the detector depends heavily on the type of extracted features from the base classifiers. The detector requires new features for the problem that are different from those used for the classification. However, extracting such features is not a simple task, since they are not always available. Secondly, designing the base classifiers and the technique used for training them is always a challenge. Various studies have shown that classifiers applied to different problems and trained by different algorithms perform differently (Kuncheva, 2004). The variation on performance between these different classifiers is due to several reasons, such as the choices of the training and testing sets, the internal randomness of the training algorithm, and the random classification error (Kuncheva, 2004). Therefore, training the base classifiers sufficiently and appropriately is a key step to achieving an improved classification performance once the classifiers are combined.

In this Chapter, a detector component is suggested as a new contribution to further enhance the performance of the initially proposed AIS based architecture. The purpose of the new detector component is to extract more features from the base classifiers that may help in performing the aggregation procedure. Figure 5.1 illustrates the proposed detector based architecture. The following subsections explain the different components of this architecture.

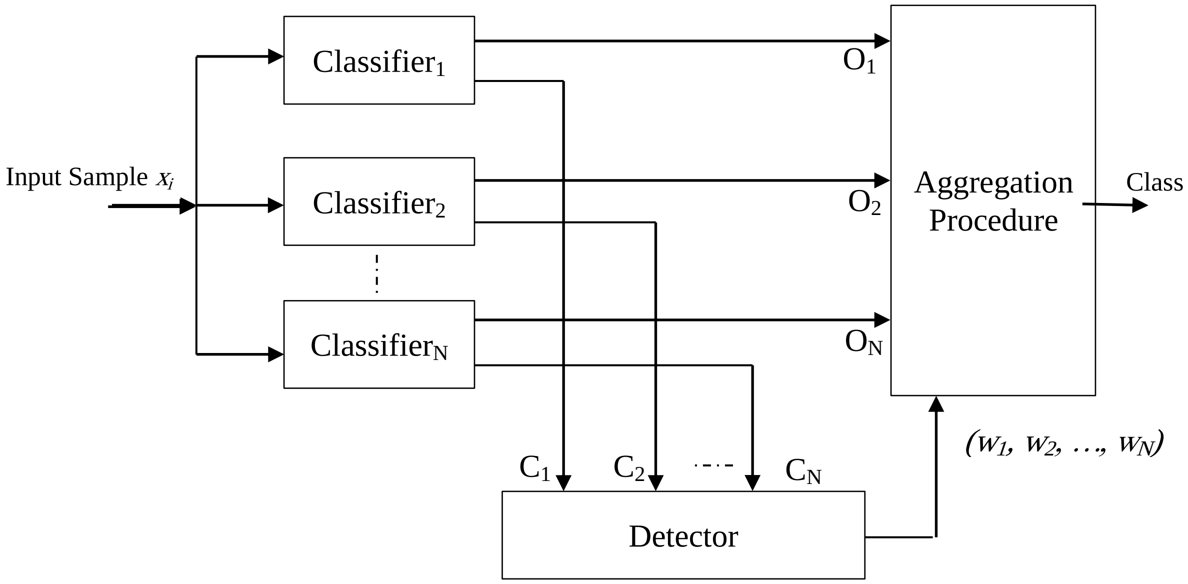


Figure 5-1: Detector Based Architecture

A. The Classifier Components

In the proposed architecture, the classifier components represent the individual classification algorithms used to form the ensemble model. The role of the classifier is to partition feature space into class-labelled decision regions. These decision regions are then used to categorize new input samples into predefined classes. Initially, the base classifiers are trained extensively in the same feature spaces and with the same training set wherein each individual classifier solves the same classification problem using different methods. The proposed architecture assumes that the classifier outputs are the predicted class labels $\boldsymbol{O} = [O_1, O_2, \ldots, O_N]$ and the confidence level of a classifier on its predicted output $\boldsymbol{C} = [C_1, C_2, \ldots, C_N]$, where N is the number of classifiers. The final output of a classifier-based ensemble system is determined by combining the outputs of the individual classifiers.

B. The Detector Components

The main function of the detector is to generate a weighting factor or confidence for each classifier. These weights are represented by the vector $\boldsymbol{W} = (w_1, w_2, \ldots, w_N)$. The weights reflect the degree of confidence in each classifier with respect to the classification accuracy performance. These weights may then be used to combine the various classifiers via the use of standard classifiers combining techniques, such as majority voting or weighted average. A detailed description of how the detector works and the main steps involved in obtaining the weights is presented in the following section.

C. The Aggregation Procedure

The aggregation procedure represents the combination strategy used to amalgamate the various outputs generated by the base classifiers, in order to conclude the final decision of the ensemble model. As shown in Figure 5.1, the aggregation procedure uses the weights resulted from the detector which represents a confidence in the output of each classifier. By using a standard combining method, all the different classification outputs are then aggregated to generate a more competent decision. Several combining methods can be used in the fusion layer; however the focus of this research is on the weighted average method presented in the previous chapter.

5.3. Neuro-fuzzy Based Detector AIS Ensemble

Fuzzy logic and fuzzy sets provide means for soft interpretation and processing of knowledge regarding a complex or ill-defined system, which is difficult to tackle using precise mathematical representation (Nauck et al., 1997; Karray and de Silva, 2004). In contrast to conventional methods, which state crisp decisions about data samples, fuzzy sets allows analysing data samples without clear or crisp boundaries or binary membership features. Fuzzy logic approaches are useful in developing expert systems. They can be employed to deal with the inexact and qualitative knowledge of experts, yet satisfactory decisions are still required (Karray and de Silva, 2004).

In this section, an overview of the neuro-fuzzy systems is presented with the emphasis on the Adaptive Network based Fuzzy Inference System (ANFIS) architecture. In addition, a neuro-fuzzy based detector is proposed, which is devoted to generating weights by learning from base classifiers and representing these weights in the form of fuzzy rules using neuro-fuzzy concepts.

5.3.1. Neuro-fuzzy Systems

Neuro-fuzzy systems represent hybrid intelligent models that inherit the features of both Artificial Neural Networks (ANN) and Fuzzy Logic (FL) systems to build advanced intelligent decision-making systems (Karray and de Silva, 2004; Vieira et al., 2004). By integrating these two intelligent approaches, the neuro-fuzzy models can benefit from the ANN advantages, including massive parallelism, robustness, and learning in data-rich environments. FL systems on the other hand offer the modelling of imprecise and qualitative knowledge, transparency as well as the transmission of uncertainty (Mitra and Hayashi, 2000). Besides these advantages, building the neuro-fuzzy systems eliminates the limitations of both ANN and FL approaches. Neuro-fuzzy systems have been used in many application domains (Karray et al., 2002; Al-Sharhan et al., 2003).

Neuro-fuzzy systems can be categorized into three main architecture types: cooperative, concurrent, and hybrid neuro-fuzzy systems (Karray and de Silva, 2004; Vieira et al., 2004). In cooperative neuro-fuzzy systems, the objective of integration is to provide the fuzzy system with the learning mechanisms of the ANN. ANN plays an initial rule of determining certain parameters of fuzzy systems such as the member functions or fuzzy rules. Alternatively, the ANN and fuzzy systems work continuously in concurrent neuro-fuzzy systems to determine the required parameters of the fuzzy system. Unlike the other two architectures, hybrid neuro-fuzzy systems have a parallel architecture, where the fuzzy logic system and the neural network work as one synchronized entity and exploit similar learning standards (Karray and de Silva, 2004).

Hybrid architectures are the most commonly known neuro-fuzzy systems, and one of the common examples is the Adaptive-Network based Fuzzy Inference System (ANFIS) proposed by Jang (1993). The ANFIS architecture is described in the following subsection.

5.3.2. ANFIS Architecture

ANFIS is a hybrid five-layer neuro-fuzzy system which implements a Takagi Sugeno fuzzy inference system (Jang, 1993). To explain how the ANFIS architecture is functioning, assume a fuzzy inference system with two inputs x and y, and one output z, and containing two if-then base rules represented as follows (Jang, 1993):

$$Rule\ 1: if\ x\ is\ A_1\ and\ y\ is\ B_1\ then\ f_1 = p_1x + q_1y + r_1 \qquad (5.1)$$

$$Rule\ 2: if\ x\ is\ A_2\ and\ y\ is\ B_2\ then\ f_2 = p_2x + q_2y + r_2 \qquad (5.2)$$

where A_1, A_2 and B_1, B_2 represent the membership functions of the two inputs, and p_i, q_i, r_i represent the parameters of the output function. Figure 5.2 illustrates the equivalent ANFIS structure of this example, where the nodes in each layer have the same functions as outlined below:

Layer 1: in this layer the input variable is mapped relatively to each membership function. For each node *i*, the output O_i^1 is defined by:

$$O_i^1 = \mu_{A_i}(x) \qquad (5.3)$$

where *x* is the input to node *i*, and A_i is the linguistic label represent the fuzzy set associated with the node function. By choosing the membership function $\mu_{A_i}(x)$ to be bell-shaped with maximum equal to 1 and minimum equal to 0, then:

$$O_i^1 = \frac{1}{1 + \left[\left(\frac{x - c_i}{a_i}\right)^2\right]^{b_i}} \qquad (5.4)$$

where {a_i, b_i, c_i} is the parameter set that changes the shapes of the membership function.

Layer 2: the node in this layer labelled as Π multiplies the incoming signals, where the output function is calculated as:

$$w_i = \mu_{A_i}(x) \times \mu_{B_i}(y), i = 1, 2. \qquad (5.5)$$

Layer 3: each node in this layer labelled as N and its output named as the normalized firing strength, which can be calculated as:

$$\overline{w}_i = \frac{w_i}{w_1 + w_2}, i = 1, 2. \qquad (5.6)$$

Layer 4: in this layer, each node I is a square node with the function:

$$O_i^4 = \overline{w}_i f_i = \overline{w}_i(p_i x + q_i y + r_i) \qquad (5.7)$$

where $\overline{w}_i$ is the output of the previous layer, and {p_i, q_i, r_i} is the parameter set denoted as consequent parameters.

Layer 5: the output layer, which calculates the total output as the summation of all the incoming input signals by:

$$O_i^5 = overall\ output = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \qquad (5.8)$$

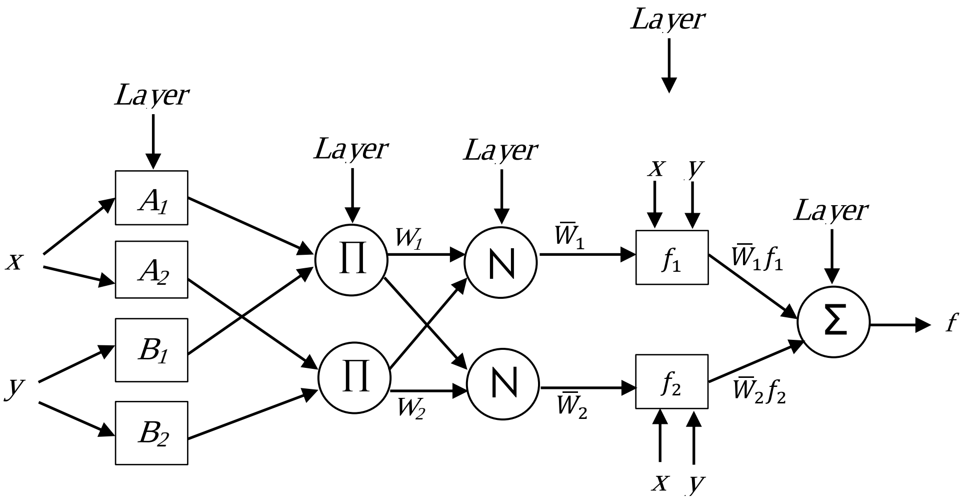


Figure 5-2: ANFIS Architecture (Jang, 1993)

5.3.3. Neuro-fuzzy Detector Development

The ANFIS methodology described above is used to develop the detector component of the proposed ensemble architecture. The ANFIS structure was derived with three inputs $A = [A_1, A_2, A_3]$ and three outputs $\boldsymbol{W} = [w_1, w_2, w_3]$, with a total of 61 rules using the clustering method. The input parameters represent the classification accuracy of the predicted output for the base classifiers while the output parameters denote the weights that will be assigned to the individual classifiers for the fusion stage, which reflects the degree of confidence in each classifier. The fuzzy inference system was constructed in this work using a Takagi-Sugeno type (Takagi and Sugeno, 1985) inference system. It is known that Takagi-Sugeno FIS is more efficient and can usually generate a better performance for accurate numerical approximation (Wu et al., 2011). It has a flexible representation capability using fewer rules to express the relation between the inputs and outputs explicitly. In addition, it is computationally effective due to its simple defuzzification process, which is based on the weighted average (Jassbi et al., 2006; Wu et al., 2011).

In the proposed system, generalized bell-shaped membership function was used to map the input values to their appropriate membership values. The membership function for each input parameter was divided into five regions, namely very low, low, medium, high, and very high to cover the full spectrum. Although several membership functions were introduced in the literature, the Gaussian and bell-shaped types are widely used for specifying fuzzy sets, due to their smoothness and concise notations (Chang and Chang, 2006). In addition to the advantage of being smooth, these curves provide nonzero membership values at all input points.

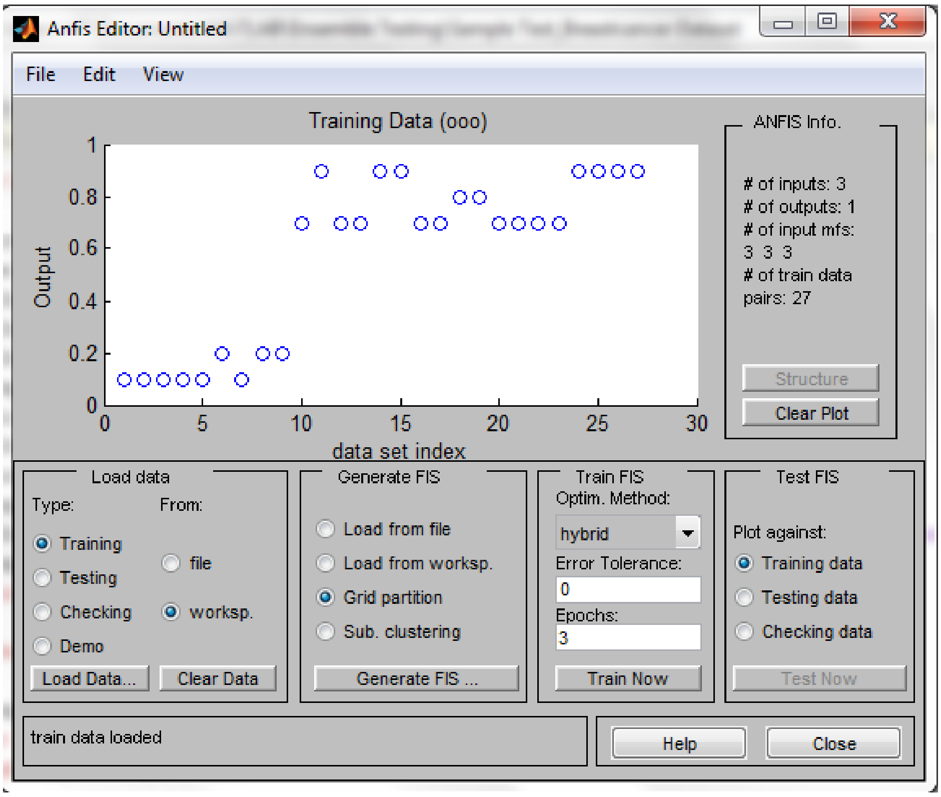
The ANFIS editor GUI in Matlab was employed to construct the ANFIS model. The first step in building the ANFIS structure is to load the data that will be used on the training stage. In this work, 27 samples were used to train the ANFIS system, referred to as training data and testing data. These samples were generated manually to represent part of the actual rules that should be generated by ANFIS. The data set used is shown in Table 5.1. Figure 5.3 shows the training data that is loaded in the ANFIS editor.

Once the training data set is loaded, the next step is to generate the new fuzzy inference system to fit the data into membership functions. The clustering partitioning method was used to generate the ANFIS network where the number of membership functions for each input parameter was set to five. Also, the generalized bell-shaped type was chosen for the membership function related to the input parameter, while the linear membership function type was chosen for the output variable.

After generating the fuzzy inference system, the next step is to train the ANFIS network. The ANFIS network was trained for 60 epochs to generate the final membership functions for all the input parameters. At the end of 60 training epochs, the network error (mean square error) convergence curve of ANFIS was derived as shown in Figure 5.4, with a final convergence value of 2.9282E−7. Finally, the ANFIS network was tested to validate the fuzzy system mapping capability. Figure 5.5 illustrates the ANFIS testing plot against the testing data.

Table 5-1: Training data for ANFIS

Sample No.	Input variables			Output variables		
	A_1	A_2	A_3	O_1	O_2	O_3
1	0.0	0.0	0.0	0.1	0.1	0.1
2	0.0	0.0	0.5	0.1	0.2	0.6
3	0.0	0.0	1.0	0.1	0.1	0.8
4	0.0	0.0	0.0	0.1	0.6	0.3
5	0.0	1.0	0.0	0.1	0.8	0.2
6	0.0	0.5	0.5	0.2	0.7	0.5
7	0.0	0.5	1.0	0.1	0.6	0.9
8	0.0	1.0	0.5	0.2	0.9	0.6
9	0.0	1.0	1.0	0.2	0.8	0.8
10	0.5	0.0	0.0	0.7	0.2	0.1
11	1.0	0.0	0.0	0.9	0.3	0.2
12	0.5	0.0	0.5	0.7	0.2	0.7
13	0.5	0.0	1.0	0.7	0.1	0.9
14	1.0	0.0	0.5	0.9	0.2	0.6
15	1.0	0.0	1.0	0.9	0.3	0.8
16	0.5	0.5	0.0	0.7	0.5	0.2
17	0.5	1.0	0.0	0.7	0.8	0.3
18	1.0	0.5	0.0	0.8	0.6	0.1
19	1.0	1.0	0.0	0.8	0.9	0.1
20	0.5	0.5	0.5	0.7	0.6	0.6
21	0.5	0.5	1.0	0.7	0.5	0.8
22	0.5	1.0	0.5	0.7	0.8	0.7
23	0.5	1.0	1.0	0.7	0.9	0.7
24	1.0	0.5	0.5	0.9	0.7	0.6
25	1.0	0.5	1.0	0.9	0.7	0.9
26	1.0	1.0	0.5	0.9	0.9	0.7
27	1.0	1.0	1.0	0.9	0.9	0.8



ANFIS Info.

# of inputs: 3
# of outputs: 1
# of input mfs:
3 3 3
# of train data pairs: 27

Load data

Type: ● Training ○ Testing ○ Checking ○ Demo

From: ○ file ● worksp.

Generate FIS

○ Load from file
○ Load from worksp.
● Grid partition
○ Sub. clustering

Train FIS

Optim. Method: hybrid

Error Tolerance: 0

Epochs: 3

Test FIS

Plot against:
● Training data
○ Testing data
○ Checking data

train data loaded

Figure 5-3: Training Samples for ANFIS Network

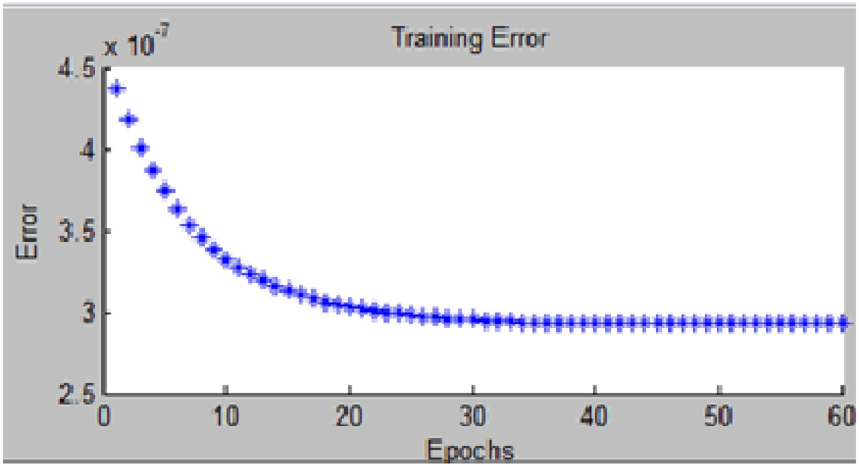
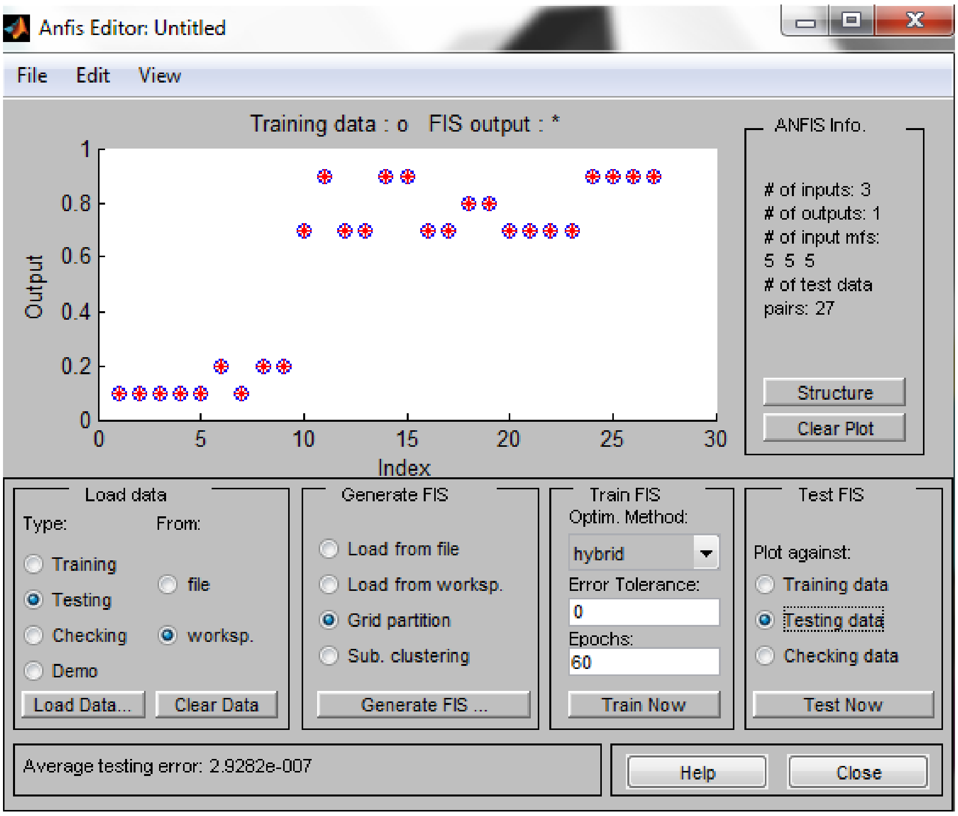


Figure 5-4: Course of error during ANFIS training


Anfis Editor: Untitled

ANFIS Info.

# of inputs: 3
# of outputs: 1
# of input mfs:
5 5 5
# of test data pairs: 27

Load data — Type: ● Testing; From: ● worksp.

Generate FIS: ● Grid partition

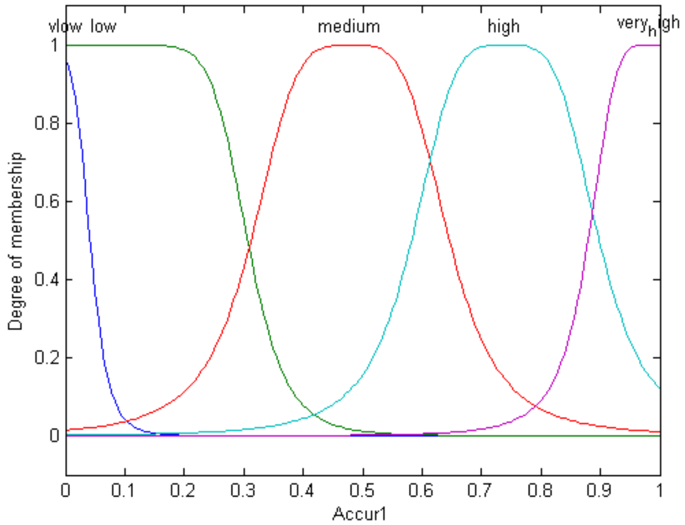
Train FIS — Optim. Method: hybrid; Error Tolerance: 0; Epochs: 60

Test FIS — Plot against: ● Testing data

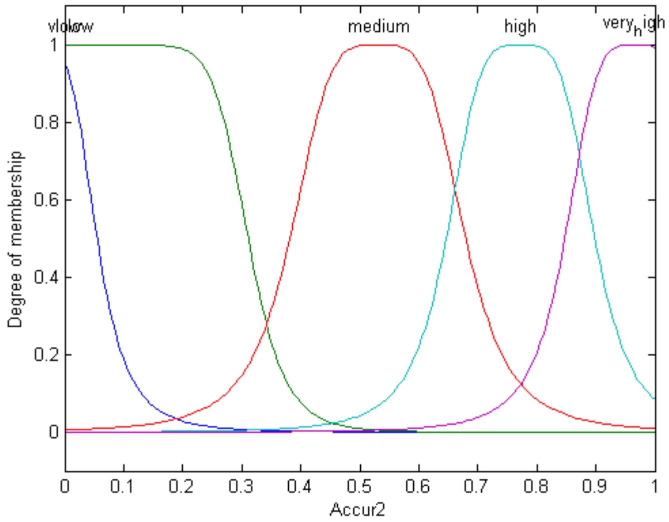
Average testing error: 2.9282e-007

Figure 5-5: ANFIS testing phase

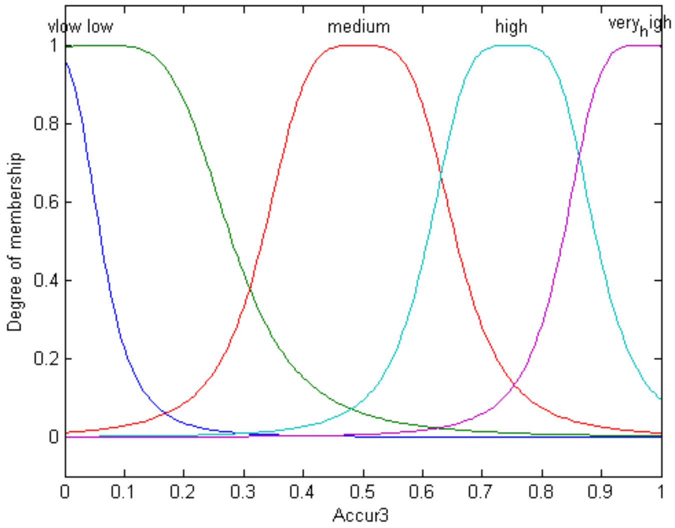
The generalized bell-shaped type membership functions after training the AFNIS for 60 epochs are shown in Figure 5.6 for all the input parameters.



(a)



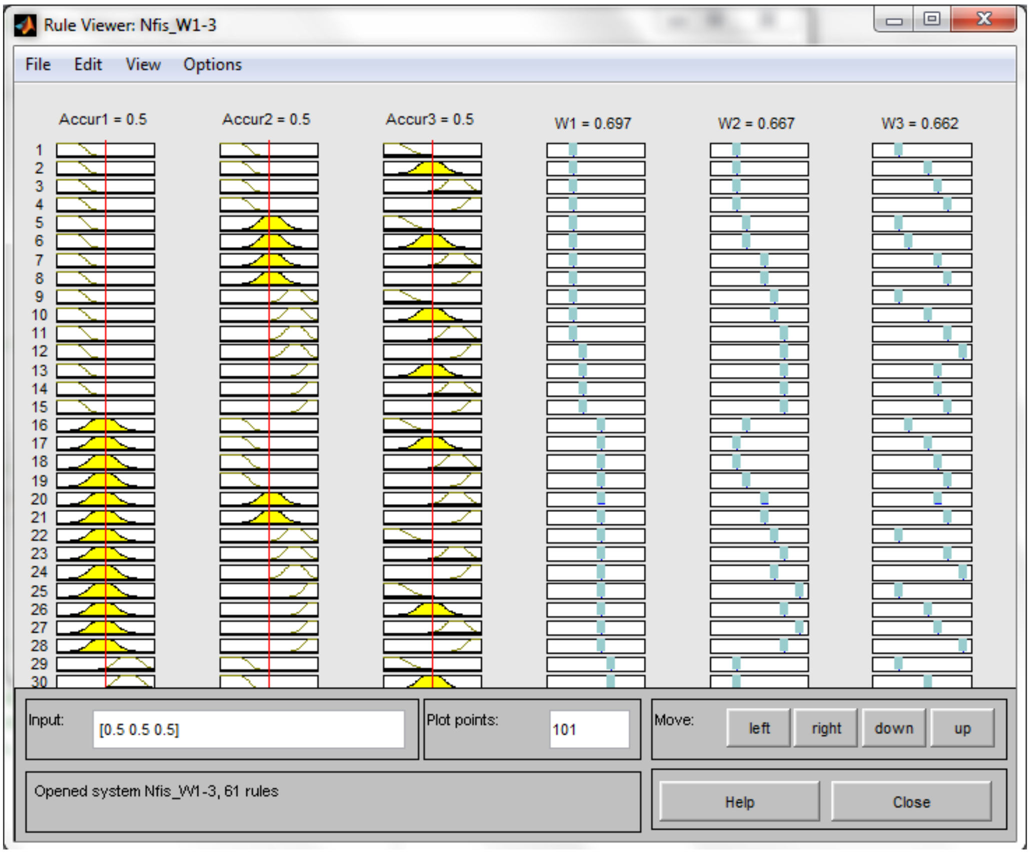
(b)



(c)

Figure 5-6: Gaussian bell-shaped MF's for (a) input 1, (b) input 2, and (c) input 3 parameters

Figure 5.7 illustrates a snapshot of the rules constructed in Matlab. The various surface shape plots between the input parameters are shown in Figure 5.8, which clearly demonstrate that some input parameters are more significant than the others.



Rule Viewer: Nfis_W1-3

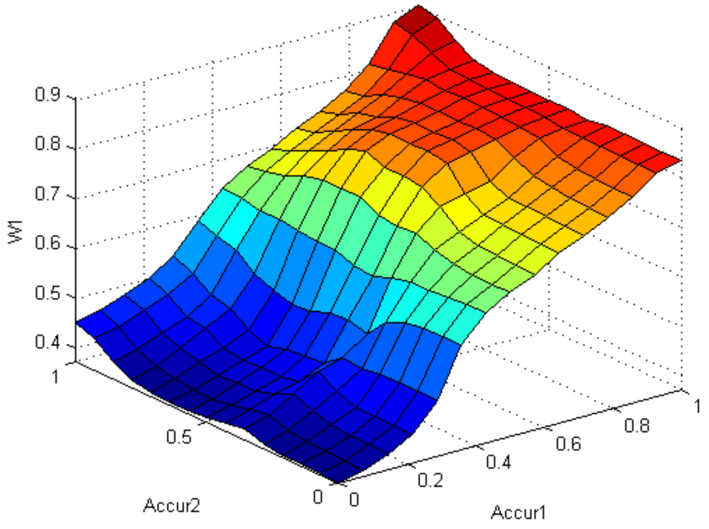
Accur1 = 0.5 Accur2 = 0.5 Accur3 = 0.5 W1 = 0.697 W2 = 0.667 W3 = 0.662

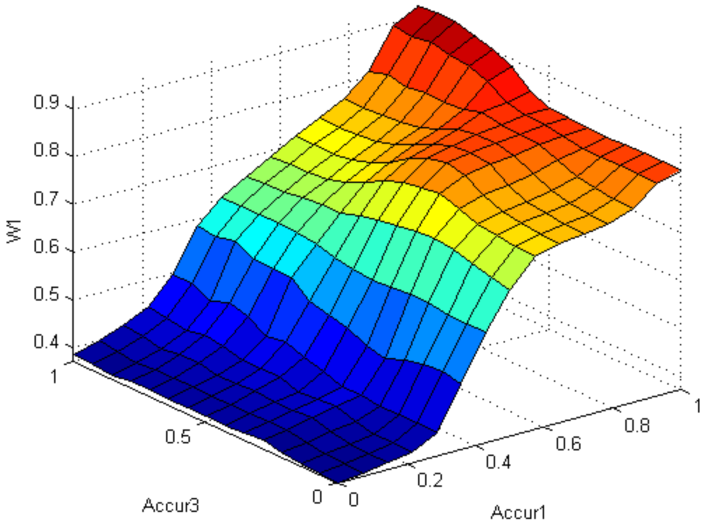
Input: [0.5 0.5 0.5]

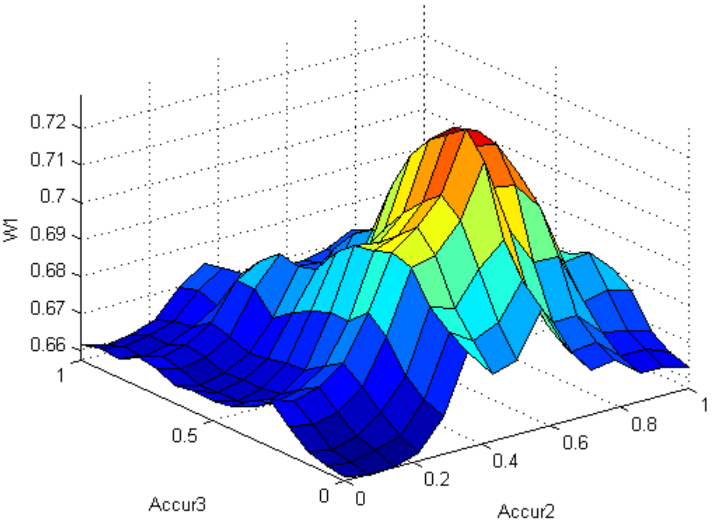
Plot points: 101

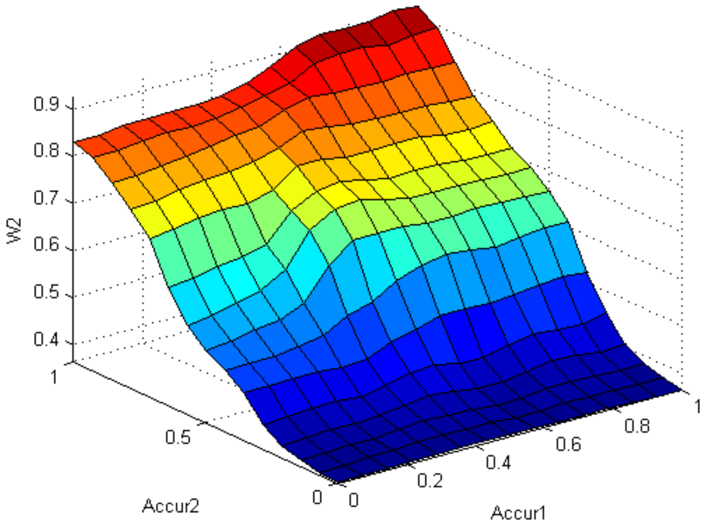
Opened system Nfis_W1-3, 61 rules

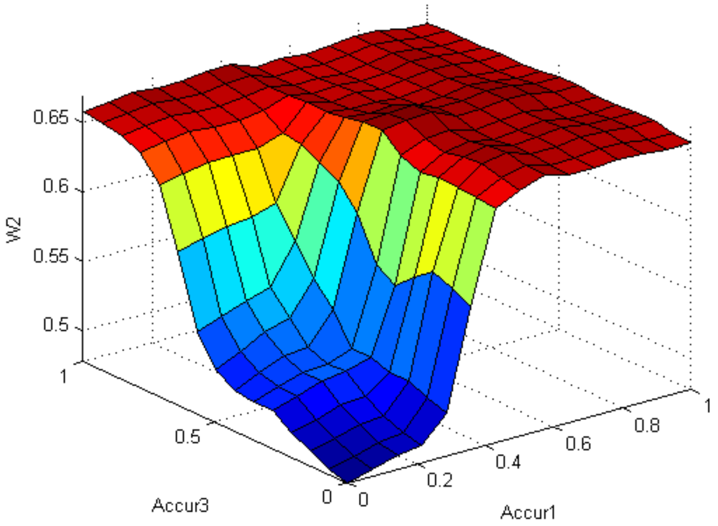
Figure 5-7: A snapshot of the ANFIS rules

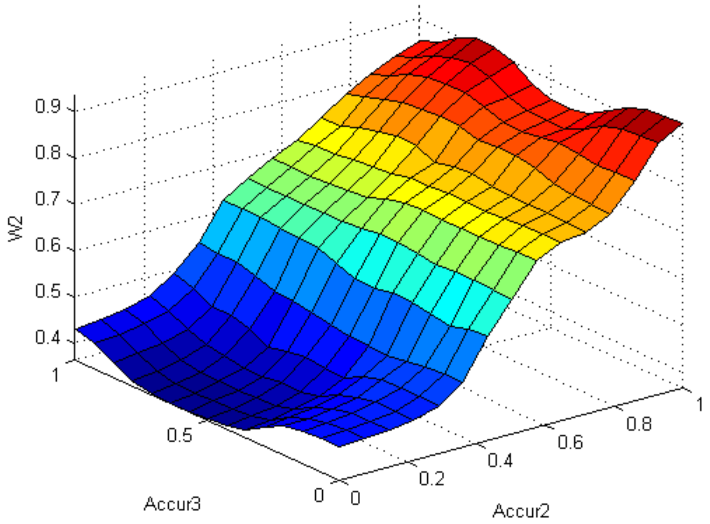


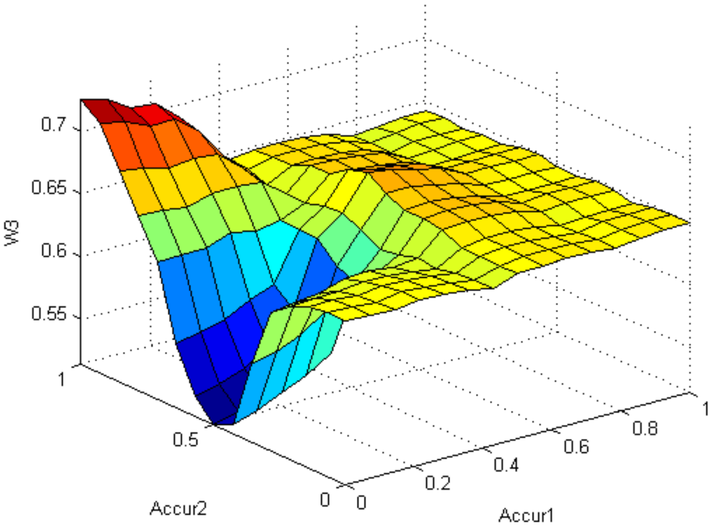

W1
0.9
0.8
0.7
0.6
0.5
0.4
1
0.5
0
Accur3
0
0.2
0.4
0.6
0.8
1
Accur1


W1
0.72
0.71
0.7
0.69
0.68
0.67
0.66
1
0.5
0
Accur3
0
0.2
0.4
0.6
0.8
1
Accur2


W2
0.9
0.8
0.7
0.6
0.5
0.4
1
0.5
0
Accur2
0
0.2
0.4
0.6
0.8
1
Accur1


0.65
0.6
0.55
0.5
W2
1
0.5
0
Accur3
0
0.2
0.4
0.6
0.8
1
Accur1


0.9
0.8
0.7
0.6
0.5
0.4
W2
1
0.5
0
Accur3
0
0.2
0.4
0.6
0.8
1
Accur2


0.7
0.65
0.6
0.55
W3
1
0.5
0
Accur2
0
0.2
0.4
0.6
0.8
1
Accur1

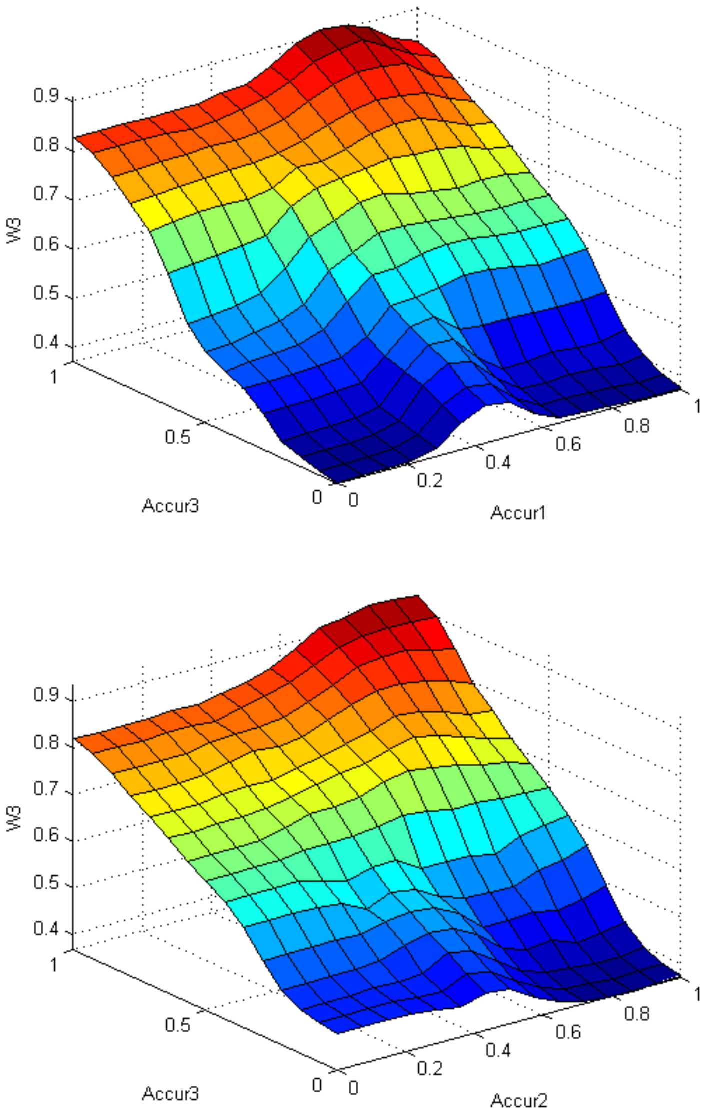


Figure 5-8: Surface Shape Plots for all variables

The surface diagrams in Figure 4.8 clearly demonstrate the relationship between the input parameters controlled by surface roughness and illustrate the interaction between each parameter. It is evident how the continuity of the output surface is assured by the ANFIS model.

5.4. The New Adaptive Learning AIS Ensemble System

The main contribution of this thesis is presented in this section. A new adaptive learning AIS based ensemble architecture is introduced for classification application that integrates different artificial intelligent and optimization techniques to enhance the overall system performance. The new proposed ensemble architecture is shown in Figure 5.9.

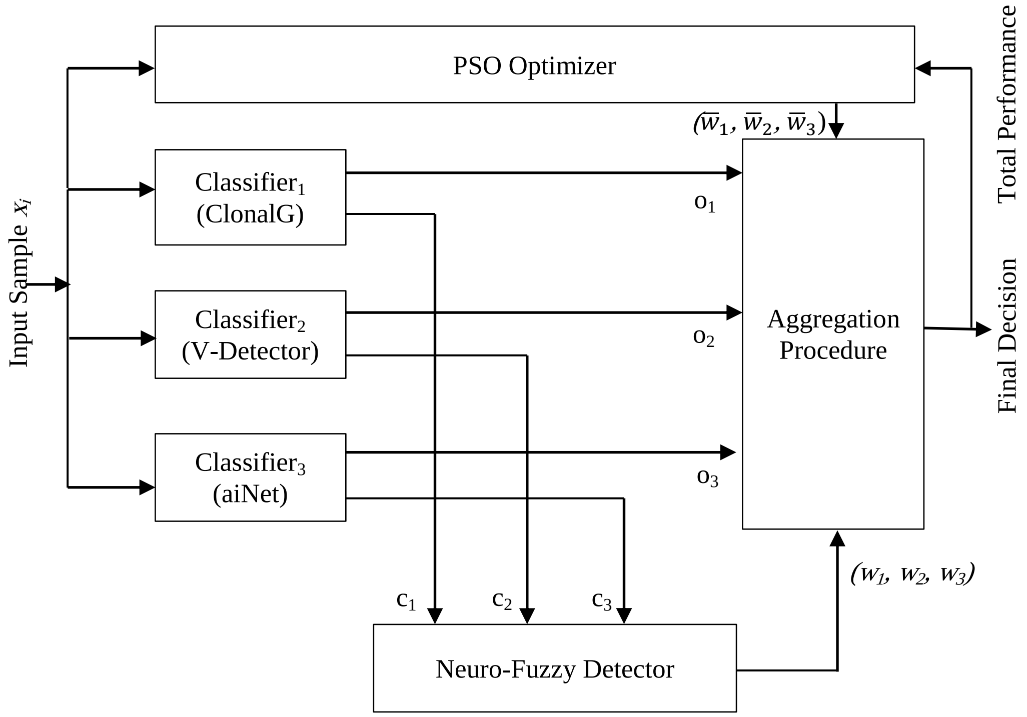


Figure 5-9: Adaptive Learning AIS Based Ensemble Architecture

The proposed adaptive learning ensemble architecture uses the PSO optimizer and the neuro-fuzzy components (explained in sections 4.4 and 5.3 respectively). By integrating these various methods, the proposed architecture utilizes their features including the PSO searching capability for best solutions and the neuro-fuzzy system learning and modelling of imprecise and qualitative knowledge characteristics. The different components of the ensemble model work in synergy with the objective of reaching an optimal classification performance outperform the best base classifier.

The proposed adaptive learning AIS based ensemble system has several main processing steps. They can be described as follows:

1. Initialization: train the three base classifiers (i.e. ClonalG, V-Detector, and aiNet algorithms) against the training data set to create the detectors sets for each algorithm that will be used in the testing phase.

2. Experts Decision: for every sample x_i in the testing dataset, determine the predicted class of x_i by each AIS classifier represented by $\boldsymbol{O} = [O_1, O_2, \ldots, O_i]$, where O_i is the predicted output of classifier number i.
3. Confidence Measure Calculation: obtain the confidence level vector $\boldsymbol{C} = [C_1, C_2, \ldots, C_i]$ on the predicted output for each classifier by:
   3.1. calculate the Euclidean distance between sample x_i and all detectors in detectors set
   3.2. If distance < threshold, then set confidence level C_i equal to 100% and O_i=1. Otherwise, set $O_i = 0$ and divide the area around x_i into bins and then calculate C_i by:

$$C_i = 1 - \frac{\sum_{i=1}^{10} m_i \times t_i}{\sum_{i=1}^{10} m_i}$$

   where m_i represents the number of detectors in each bin and t_i is the bin index.
4. Neuro-fuzzy detector stage: the confidence level values obtained in the previous step will be used by the neuro-fuzzy detector to determine the weight values $\boldsymbol{W} = [w_1, w_2, \ldots, w_i]$ for each base classifier. Considering a first order Sugeno type ANFIS model with the three inputs (C_1=X_1, C_2=X_2, C_3=X_3), and five membership functions for each input, then:
   4.1. The if-then rule base can be expressed as follows

   Rule 1: if X_1 is A_1 and X_2 is B_1 and X_3 is C_1,

   Then $f_{1,1,1} = p_{1,1,1}X_1 + q_{1,1,1}X_2 + r_{1,1,1}X_3 + s_{1,1,1}$

   Rule 2: if X_1 is A_1 and X_2 is B_1 and X_3 is C_2,

   Then $f_{1,1,2} = p_{1,1,2}X_1 + q_{1,1,2}X_2 + r_{1,1,2}X_3 + s_{1,1,2}$

   Rule 3: if X_1 is A_1 and X_2 is B_1 and X_3 is C_3,

   Then $f_{1,1,3} = p_{1,1,3}X_1 + q_{1,1,3}X_2 + r_{1,1,3}X_3 + s_{1,1,3}$

   ⋮

   Rule 125: if X_1 is A_5 and X_2 is B_5 and X_3 is C_5,

   Then $f_{5,5,5} = p_{5,5,5}X_1 + q_{5,5,5}X_2 + r_{5,5,5}X_3 + s_{5,5,5}$

   where A_i, B_j, and C_k (i, j, k = 1, 2, …, 5) are the linguistic labels associated with this node function.

   4.2. *Layer 1*: the node function in this layer represented by:

$$O_{1,i}^{1} = \mu_{A_i}(X_1), \qquad for\ i = 1, 2, \ldots, 5$$

$$O_{2,j}^{1} = \mu_{B_j}(X_2), \qquad for\ j = 1, 2, \ldots, 5$$

$$O_{3,k}^{1} = \mu_{C_k}(X_3), \qquad for\ k = 1, 2, \ldots, 5$$

   Where $O_{1,i}^{1}$, $O_{2,j}^{1}$, and $O_{3,k}^{1}$ are the MFs for A_i, B_j, and C_k respectively.

4.3. *Layer 2*: each node output represents the firing strength of a rule. It multiplies the incoming signals and sends the product out. For instance:

$$O^2_{i,j,k} = w_{i,j,k} = \mu_{A_i}(X_1) \,.\, \mu_{B_j}(X_2) \,.\, \mu_{C_k}(X_3), \quad for\ i,j,k = 1,2,\ldots,5$$

4.4. *Layer 3*: The *ith* node in this layer calculates the ratio of the *ith* rule's firing strength to the sum of all rule's firing strengths:

$$O^3_{i,j,k} = \overline{w}_{i,j,k} = \frac{w_{i,j,k}}{\sum_{i,j,k=1}^{3} w_{i,j,k}}, \quad i,j,k = 1,2,\ldots,5$$

4.5. *Layer 4*: Every node *i* in this layer is a square node with a node function:

$$O^4_{i,j,k} = \overline{w}_{i,j,k} \,.\, f_{i,j,k} = w_{i,j,k} \,.\, \left(p_{i,j,k}X_1 + q_{i,j,k}X_2 + r_{i,j,k}X_3 + s_{i,j,k}\right),$$
$$i,j,k = 1,2,\ldots,5$$

4.6. *Layer 5*: in this layer, the overall output is computed as a summation of all the incoming signals:

$$O^5_i = Overall\ output = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}$$

4.7. The weight for each base classifier is equal to the overall output of the neuro-fuzzy system, i.e. $W = O^5_i$

5. PSO Optimizer stage: obtain the optimized weights vector $\overline{W} = (\overline{W}_1, \overline{W}_2, \overline{W}_3)$ using PSO algorithm by obtaining the *gbest* values for $\overline{W}_i$.

6. Aggregation Process: using the weighted average combining method, perform the aggregation procedure using the predicted output for each classifier ***O*** = [*O*₁, *O*₂,…, *Oⱼ*], the weights obtained by the neuro-fuzzy detector ***W*** = [*W*₁, *W*₂, …, *Wⱼ*], and the optimized weights from the PSO optimizer $\overline{W}$ = [$\overline{W}_1$, $\overline{W}_2$,…, $\overline{W}_j$] by:

$$f(x_i) = \frac{\sum_j O_j \times W_j \times \overline{W}_j}{\sum_j \overline{W}_j}$$

Where $f(x_i)$ represents the final predicted class for sample x_i by the ensemble model, *j* is the number of base classifiers.

The above steps are repeated for all the testing samples. Considering the architecture shown in Figure 5.9, where three AIS classifiers were used, in this case, the adaptive learning AIS based ensemble model works as per the following scenario. Initially, the various members of the ensemble are trained on the whole training dataset in order to make a decision on an input sample x_i from the testing dataset. As explained in Chapter 4, the ClonalG, V-Detector, and aiNet AIS algorithms are used as base classifiers in this architecture. The predicted outputs of the AIS classifiers are represented by vector $\boldsymbol{O} = [O_1, O_2, O_3]$, as shown in Figure 5.6. Accordingly, the confidence values denoted by vector $\boldsymbol{C} = [C_1, C_2, C_3]$ are calculated using the method described in subsection 4.3.1, which are then assigned to each one of the base classifiers to measure their first level accuracies regarding the predicted class. In the next step, the resultant confidence values are fed to the neuro-fuzzy detector, where the new weights represented by vector $\boldsymbol{W} = [w_1, w_2, w_3]$ are extracted, as explained in section 5.3. Using the neuro-fuzzy detector at this stage helps in transforming the crisp values of the confidence levels assigned to the individual classifiers into a more accurate and satisfactory weight measures that will lead to an appropriate decision. In addition, instead of interpreting the knowledge extracted from each base classifier separately, the neuro-fuzzy detector looks at the performance results of the base classifiers altogether and set the weights accordingly.

Once the weights $W = [w_1, w_2, w_3]$ are calculated by the neuro-fuzzy detector, the fusion stage can be performed. However, the new adaptive learning AIS ensemble architecture suggests a method for optimizing the values of the weights using the PSO optimizer technique (as discussed in section 4.4). The aim of performing the PSO optimization technique is to further enhance the overall ensemble performance by searching for optimal weights values. The process of optimization continues till a predefined threshold is reached, hence, the final weights are achieved with optimum values, which are denoted by vector $\overline{W} = (\overline{W}_1, \overline{W}_2, \overline{W}_3)$.

As a final stage, the fusion process is performed using one of the aggregation procedures to conclude the final decision. The weighted average combining procedure (as discussed in section 4.2) is employed in this architecture. The optimized weight values along with the predicted outputs of the base classifier as well as the confidence levels that resulted from the neuro-fuzzy detector are all used to tune the weighted average method in order to reach the final decision. The final decision representing the class of input sample x_i is calculated by:

$$f(x_i) = \frac{o_1 \times w_1 \times \bar{w}_1 + o_2 \times w_2 \times \bar{w}_2 + o_3 \times w_3 \times \bar{w}_3}{\bar{w}_1 + \bar{w}_2 + \bar{w}_3} \tag{5.9}$$

Figure 5.10 shows a flow chart including the main steps of the proposed AIS ensemble with adaptive learning feature. The adaptive learning AIS ensemble architecture allows for the dynamic decision fusion of classifiers. The aggregation procedure in this architecture has the flexibility to adapt to changes in the input and output in order to improve the final decision. In this architecture, the key underlying concept is to understand the changes in the input and its impact on the base classifiers, by means of extracting features using the neuro-fuzzy detector, to direct the way it performs the aggregation. Then, the PSO optimizer helps the fusion process in learning how to combine the different decisions in order to improve the overall classification performance of the system. More discussion on the performance of the proposed adaptive learning AIS based ensemble model is presented in the next chapter.

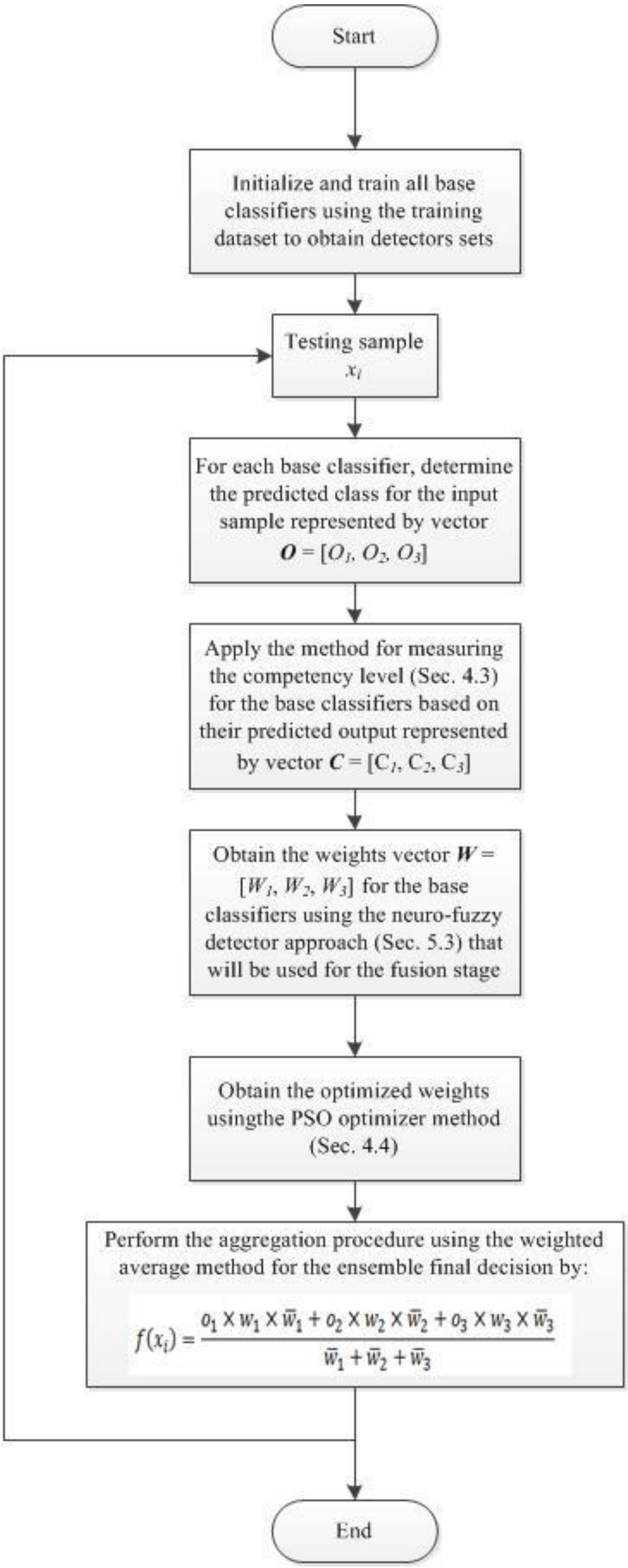

Start
Initialize and train all base classifiers using the training dataset to obtain detectors sets
Testing sample x_i
For each base classifier, determine the predicted class for the input sample represented by vector $\boldsymbol{O} = [O_1, O_2, O_3]$
Apply the method for measuring the competency level (Sec. 4.3) for the base classifiers based on their predicted output represented by vector $\boldsymbol{C} = [C_1, C_2, C_3]$
Obtain the weights vector $\boldsymbol{W} = [W_1, W_2, W_3]$ for the base classifiers using the neuro-fuzzy detector approach (Sec. 5.3) that will be used for the fusion stage
Obtain the optimized weights usingthe PSO optimizer method (Sec. 4.4)
Perform the aggregation procedure using the weighted average method for the ensemble final decision by:
$f(x_i) = \frac{o_1 \text{ X } w_1 \text{ X } \bar{w}_1 + o_2 \text{ X } w_2 \text{ X } \bar{w}_2 + o_3 \text{ X } w_3 \text{ X } \bar{w}_3}{\bar{w}_1 + \bar{w}_2 + \bar{w}_3}$
End

Figure 5-10: Flow chart of the adaptive learning AIS based ensemble system

5.5.Summary

The detector based ensemble architecture as an enhancement to the original ensemble architecture has been introduced in this chapter, and the main components of the architecture are discussed. A neuro-fuzzy detector has been suggested to enhance the performance of the ensemble model using the fuzzy system concepts. Also, an overview of the neuro-fuzzy system principles and the development of the neuro-fuzzy based detector have been presented.

Finally, the chapter introduced a new adaptive learning AIS based ensemble architecture as the main contribution of this work, integrating different artificial intelligence and optimization methods for classification application. The proposed ensemble architecture benefits from the features of particle swarm optimization and neuro-fuzzy methods to achieve an optimal classification performance.

In the next chapter, an empirical study is presented on the classification performance of the new adaptive learning AIS ensemble system in comparison to other models. Various case studies are presented to test the performance of the proposed ensemble architecture against actual datasets and the final results are discussed in detail.

CHAPTER 6: Results and Discussion

6.1.Introduction

This chapter presents an empirical evaluation of the adaptive learning AIS based ensemble as compared to the various conventional classifiers. In this evaluation, several experiments were conducted to demonstrate the effectiveness of the proposed AIS ensemble model with different medical datasets. The main objectives of these experiments were to demonstrate the performance of the proposed adaptive learning AIS ensemble model compared to the conventional AIS classifiers and to compare it with other combining methods, such as majority voting and weighted average.

In the evaluation study, all the comparisons between the different methods were conducted with equal training and testing parts of the data. In addition, for all the experiments, the datasets used were normalized initially to unity before being fed to the algorithms and split into 95% for training set and 5% test set. For each experiment, 100 runs were performed, in which different training and testing samples were chosen randomly in each round. The performance measures used to compare the performance of the different methods are shown below in table 6.1:

Table 6-1: Performance measures used

Performance Measure	Definition	Equation
Accuracy	Measures the proportion of correctly classified instances	*equ. 3.4*
Sensitivity	Measures the fraction of actual positive examples that are correctly classified	*equ. 3.5*
Specificity	Measures the fraction of actual negative examples that are correctly classified	*equ. 3.6*
Total	Measures the overall performance as a summation of the accuracy, sensitivity, and specificity	*equ. 3.7*

The top ten results were selected from all runs to present the overall performance of the individual methods. The experiments conducted in this chapter are a continuation of the case studies performed in Chapters 3 and 4. The Matlab software was used for all the codes, running on a Window 7 (64-bit operating system) machine with Intel core processor i7-2.20 GHz CPU and 8 GB RAM.

6.2.Datasets

In this evaluation study, four different medical datasets were used in various experiments to demonstrate the effectiveness of the proposed adaptive learning AIS based architecture. These data sets are the Wisconsin Breast Cancer (WBC), which was presented in Chapter 3; Bladder Cancer (BC); Haberman's Survival (HS); and the Pima Indians Diabetes (PID) data sets. Beside the fact that three of these datasets are high-dimensional data, they all are unbalanced where the number of positive and negative samples are not equal. Furthermore, the selected datasets have different type of representation for the attributes where some of them are binary and the others are continuous. All these challenges on the datasets are of interest to examine the effectiveness of the proposed ensemble model. A brief description of the data sets used is given below.

A. Bladder Cancer Dataset

The bladder cancer data set consisted of 693 instances with 12 attributes. The predicted output of this dataset represents the recurrence status where the value one as an output indicates the possibility of the patient getting the cancer again in the future. Post-operative tumor recurrence occurred in 219 (31.6%) patients. Table 6.2 shows a snapshot from this dataset. The 12 attributes of the dataset are:

- Gender (0 = F, 1 = M),
- Age (years)
- Pathologic Stage (pT1, pT2, pT3, or pT4)
- Pathologic Grade (Grade 2 or 3)
- Carcinoma In Situ – CIS (0 = Absent, 1= Present)
- Margin status (0 = Clear, 1 = Involved)
- Lymph nodes removed (pN0, pN1-3)
- Lymphovascular invasion (0 = Absent, 1= Present)
- p53
- p21
- pRb
- p27

Table 6-2: Sample of Bladder cancer data set

Gender	Age	Pathologic Stage	Pathologic Grade	CIS	Margin status	Lymph nodes removed	Lymphovascular invasion	p53	p21	pRb	p27	Recurrence Status
0	74	2	2	0	0	25	0	1	0	1	0	0
0	62	2	2	0	0	20	0	1	0	0	1	0
1	70	3	2	0	0	15	1	1	1	0	1	1
1	66	3	2	0	0	11	1	0	0	0	1	1
1	66	2	2	0	0	16	0	0	1	1	1	1
1	76	2	2	0	0	25	0	0	0	1	0	1
1	77	4	2	0	0	13	1	0	0	1	0	0
1	72	3	2	0	0	14	0	0	1	0	1	1
1	68	3	2	1	0	25	0	0	0	1	0	0
1	79	3	2	0	0	26	0	1	0	0	1	0

B. Haberman's Survival Dataset

This dataset includes cases from a study that was conducted at the University of Chicago's Billings Hospital on the survival of patients who had undergone surgery for breast cancer (Frank and Asuncion, 2010). It contains 306 instances and 3 attributes, and the predicted output represents two classes. The first class represents the patient surviving 5 years or longer, and the second class is for patients who died within 5 years. The dataset has three attributes:

- Age of patient at time of operation (years)
- Patient's year of operation (years)
- Number of positive axillary nodes detected (numerical)

C. Pima Indians Diabetes Dataset

The Pima Indians Diabetes Dataset (Frank and Asuncion, 2010) was obtained from the UCI Repository of Machine Learning Databases. All patients in this database are Pima Indian women at least 21 years old. This dataset has two classes with binary values 0 or 1, where 1 means a positive test and 0 means a negative test for diabetes. The number of instances included in the dataset is 768; 268 (34.9%) cases in class 1, and 500 (65.1%) cases in class 0. The dataset has eight attributes:

- Number of pregnancies
- Plasma glucose concentration after 2 hours in an oral glucose tolerance test
- Diastolic blood pressure (mm Hg)
- Triceps skin fold thickness (mm)
- 2-hour serum insulin (mu U/ml)
- Body mass index
- Diabetes pedigree function
- Age (years)

6.3. Experimental Evaluation

Various experiments were conducted to test the effectiveness of the popular AIS algorithms and the ensemble methods proposed in this work against a number of datasets. The detailed performance results for each experiment are presented in the following subsections.

6.3.1. Experiment # 1: AIS Algorithms

In this experiment, the ClonalG, V-Detector and aiNet AIS algorithms were chosen with the purpose of evaluating them and exploring their capabilities for the classification application. The three medical data sets used for this experiment are: the BC, HS, and the PID dataset. All the variables used in the three AIS algorithms have been set in this experiment as described in the case study presented in Chapter 3.

For the BC dataset, the experimental performance results obtained (as shown in Table 6.3 and Table 6.4 respectively) indicate that V-Detector algorithm achieved better average classification accuracy compared to the other two algorithms in six out of ten rounds; ClonalG achieved four out of ten, and aiNet showed the poorest accuracy. For the sensitivity measure, aiNet consistently performed the best in all ten rounds, averaging 100%; V-Detector has consistently ranked second-best, averaging over 68%; and ClonalG consistently performed poorly for this measure. For the specificity measure, the ClonalG algorithm consistently performed the best in all ten rounds, while the V-Detector consistently achieved good performance results, averaging over 77%; however, the testing results obtained from the aiNet algorithm were 0% throughout. Figure 6.1 illustrates the overall performance results for all the algorithms against this dataset. The findings rank V-Detector highest and ClonalG second best in performance for the BC dataset.

Table 6-3: Performance Results for the AIS algorithms against BC dataset

Round #	Accuracy			Sensitivity			Specificity		
	ClonalG	V-Detector	aiNet	ClonalG	V-Detector	aiNet	ClonalG	V-Detector	aiNet
1	0.743	0.771	0.286	0.300	0.900	1.000	0.920	0.720	0.000
2	0.559	0.765	0.294	0.000	0.700	1.000	0.792	0.792	0.000
3	0.686	0.743	0.371	0.154	0.692	1.000	1.000	0.773	0.000
4	0.800	0.743	0.200	0.143	0.714	1.000	0.964	0.750	0.000
5	0.800	0.743	0.200	0.429	0.714	1.000	0.893	0.750	0.000
6	0.657	0.743	0.343	0.250	0.667	1.000	0.870	0.783	0.000
7	0.794	0.706	0.235	0.125	0.750	1.000	1.000	0.692	0.000
8	0.750	0.719	0.188	0.000	0.667	1.000	0.923	0.731	0.000
9	0.676	0.735	0.412	0.357	0.429	1.000	0.900	0.950	0.000
10	0.618	0.706	0.471	0.313	0.625	1.000	0.889	0.778	0.000

Table 6-4: The overall performance of the AIS Algorithms against BC dataset

Round #	Total Performance		
	ClonalG	V-Detector	aiNet
1	1.963	2.391	1.286
2	1.350	2.256	1.294
3	1.840	2.208	1.371
4	1.907	2.207	1.200
5	2.121	2.207	1.200
6	1.777	2.192	1.343
7	1.919	2.148	1.235
8	1.673	2.116	1.188
9	1.934	2.114	1.412
10	1.819	2.109	1.471
Average	**1.830**	**2.195**	**1.300**

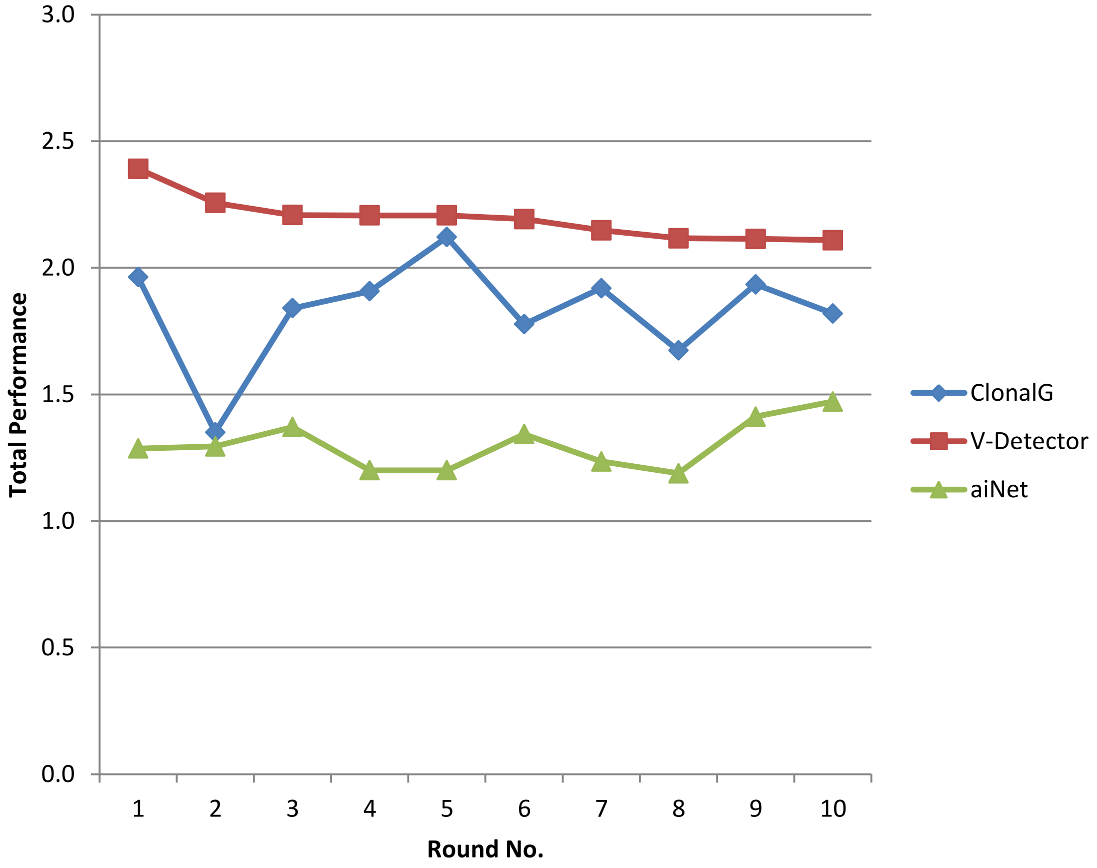


Figure 6-1: The overall performance of AIS Algorithms against BC dataset

Figure 6.1 represents the overall performance of the AIS algorithms against the BC dataset. It is evident that the V-Detector delivers total performance results consistently better than the other two algorithms, scoring above 2.0. ClonalG is second in total performance with results falling in a band just below, but showing wide fluctuation between 1.350 and 2.121. aiNet is consistently poorest in total performance averaging just 1.300.

Similarly, the results presented in Tables 6.5 and 6.6 for the HS dataset has shown that the V-Detector algorithm also achieved better accuracy results. However, the aiNet algorithm also achieved a very good accuracy results compared to the previous dataset. The total classification performance results for the three AIS algorithms against the HS are depicted in Figure 6.2.

Table 6-5: Performance Results for the AIS algorithms against HS dataset

Round #	Accuracy			Sensitivity			Specificity		
	ClonalG	V-Detector	aiNet	ClonalG	V-Detector	aiNet	ClonalG	V-Detector	aiNet
1	0.733	0.867	0.933	0.000	0.333	0.667	0.917	1.000	1.000
2	0.733	0.800	0.867	0.000	0.667	0.333	0.917	0.833	1.000
3	0.667	0.800	0.667	0.000	0.400	0.000	1.000	1.000	1.000
4	0.800	0.733	0.733	0.000	0.667	0.000	1.000	0.750	0.917
5	0.800	0.733	0.800	0.000	0.000	0.333	1.000	0.917	0.917
6	0.467	0.667	0.600	0.000	0.375	0.250	1.000	1.000	1.000
7	0.786	0.786	0.786	0.000	0.333	0.000	1.000	0.909	1.000
8	0.600	0.600	0.667	0.143	0.286	0.286	1.000	0.875	1.000
9	0.533	0.667	0.600	0.000	0.286	0.143	1.000	1.000	1.000
10	0.600	0.733	0.667	0.000	0.200	0.000	0.900	1.000	1.000

Table 6-6: The overall performance of the AIS Algorithms against HS dataset

Round #	Total Performance		
	ClonalG	V-Detector	aiNet
1	1.650	2.200	2.600
2	1.650	2.300	2.200
3	1.667	2.200	1.667
4	1.800	2.150	1.650
5	1.800	1.650	2.050
6	1.467	2.042	1.850
7	1.786	2.028	1.786
8	1.743	1.761	1.952
9	1.533	1.952	1.743
10	1.500	1.933	1.667
Average	**1.660**	**2.022**	**1.916**

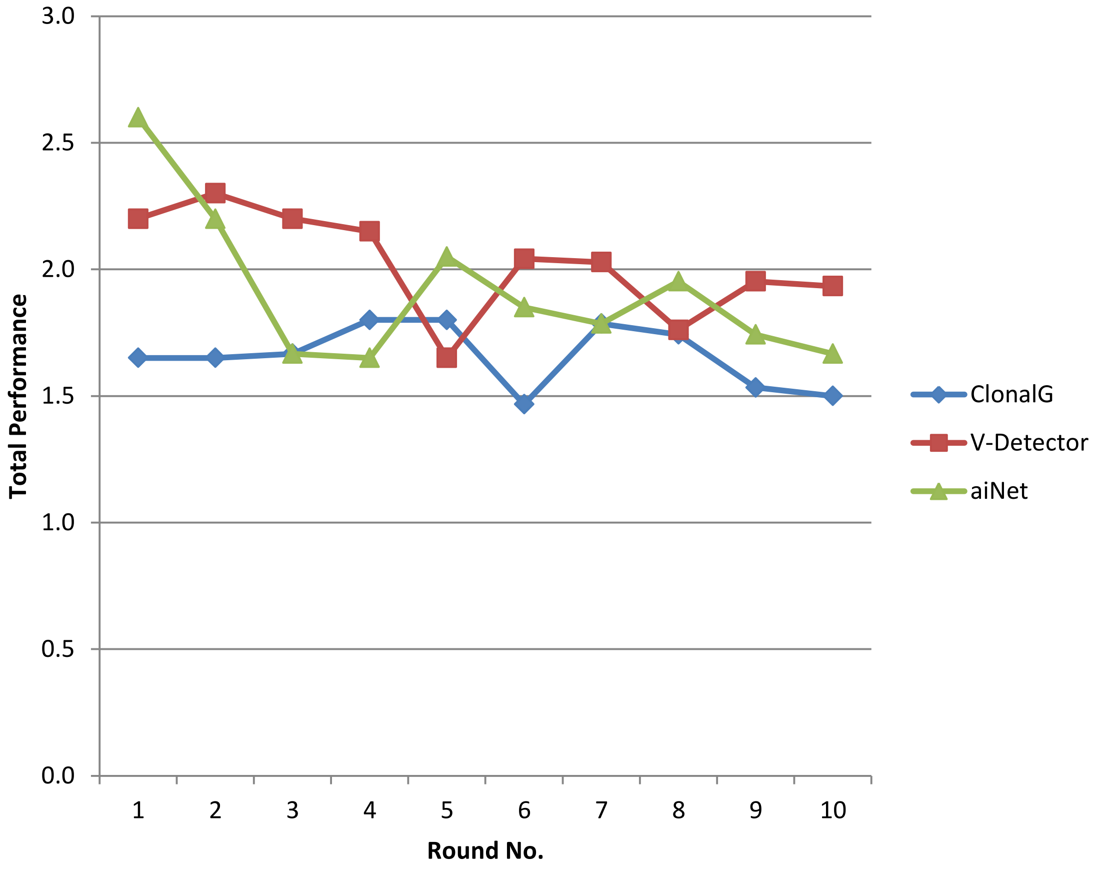


Figure 6-2: The overall performance of AIS Algorithms against HS dataset

Figure 6.2 represents the overall performance of the AIS algorithms against the HS dataset. It is evident that the V-Detector again delivers total performance results generally better than the other two, scoring above 2.000. For this dataset, aiNet performs much better than ClonalG and is closely second in total performance averaging a score of 1.916. ClonalG in third place performed very poorly compared with the other two, with total performance averaging 1.660.

It has also been found that for the PID dataset, the ClonalG algorithm achieved slightly higher average accuracy rates compared to the V-Detector algorithm, as shown in Table 6.7 and Table 6.8. However, the results obtained from the aiNet algorithm still were unacceptably low and need more improvements. Figure 6.3 outlines the overall accuracy results for the three AIS algorithms against this dataset.

Table 6-7: Performance Results for the AIS algorithms against PID dataset

Round #	Accuracy			Sensitivity			Specificity		
	ClonalG	V-Detector	aiNet	ClonalG	V-Detector	aiNet	ClonalG	V-Detector	aiNet
1	0.737	0.711	0.421	0.500	0.375	1.000	0.909	0.955	0.000
2	0.684	0.763	0.395	0.600	0.400	1.000	0.739	1.000	0.000
3	0.722	0.722	0.306	0.545	0.364	1.000	0.800	0.880	0.000
4	0.730	0.730	0.351	0.538	0.385	1.000	0.833	0.917	0.000
5	0.611	0.750	0.333	0.250	0.250	1.000	0.792	1.000	0.000
6	0.757	0.730	0.270	0.400	0.500	1.000	0.889	0.815	0.000
7	0.711	0.684	0.342	0.462	0.231	1.000	0.840	0.920	0.000
8	0.684	0.684	0.342	0.385	0.385	1.000	0.840	0.840	0.000
9	0.632	0.579	0.474	0.389	0.222	1.000	0.850	0.900	0.000
10	0.611	0.722	0.806	0.571	0.286	0.000	0.621	0.828	1.000

Table 6-8: The overall performance of the AIS Algorithms against PID dataset

Round #	Total Performance		
	ClonalG	V-Detector	aiNet
1	2.146	2.040	1.421
2	2.023	2.163	1.395
3	2.068	1.966	1.306
4	2.102	2.031	1.351
5	1.653	2.000	1.333
6	2.046	2.045	1.270
7	2.012	1.835	1.342
8	1.909	1.909	1.342
9	1.870	1.701	1.474
10	1.803	1.836	1.806
Average	**1.963**	**1.953**	**1.404**

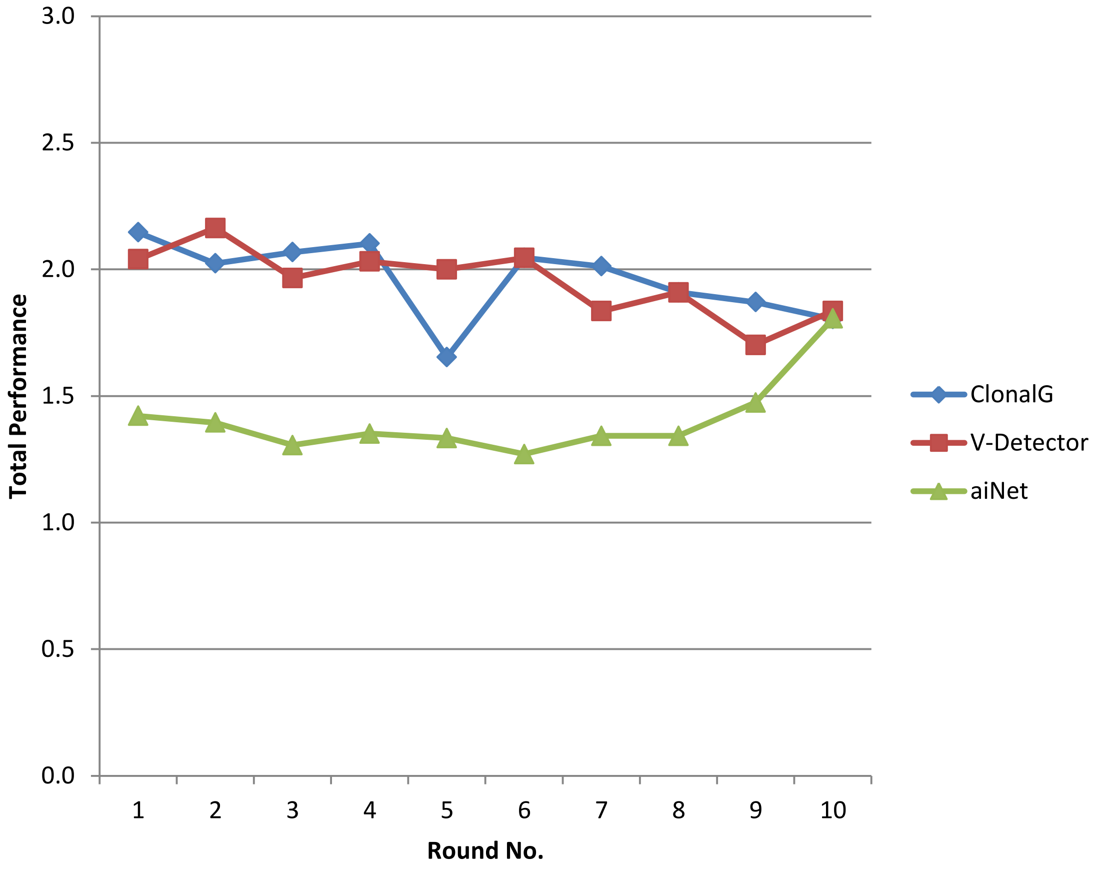


Figure 6-3: The overall performance of AIS Algorithms against PID dataset

Figure 6.3 represents the overall performance of the AIS algorithms against the PID dataset. In this case, ClonalG delivers total performance results averaging 1.963, outperforming V-Detector's average score of 1.953, despite one outlier in round no. 5. V-Detector performs almost as well as ClonalG for this dataset. aiNet is consistently trailing in third place, with the total performance averaging 1.404, indicating the need for significant improvement before it can be considered as a viable option.

The case studies carried out clearly demonstrate how the AIS approaches can be employed in dealing with real-world problems in health and cancer research. The three experiments conducted to test the ClonalG, V-Detector and aiNet algorithms respectively against three medical datasets yielded mixed results. In general, good performance results were obtained in all tests for some of the AIS algorithms, especially for the V-Detector algorithm. However, the overall classification performance results are still unacceptably low, and more improvements are required to yield better outcomes. This outcome leads to the conclusion that some of the AIS techniques are found to be more suitable for medical research than other AIS approaches.

6.3.2. Experiment # 2: Majority Voting AIS Ensemble System

The AIS ensemble system based on the majority voting classifiers combining method was tested in this experiment against three medical datasets. The datasets used here are the BC, HS and PID. Table 6.9 highlights the detailed classification performance results for the majority voting AIS ensemble method. Table 6.10 and Figure 6.4 illustrate the total classification performance of the majority voting AIS ensemble system.

The presented results show that the majority voting AIS ensemble has achieved on average a consistent classification performance outcome for the three datasets. The results obtained from the BC dataset test are slightly better compared to the other two datasets. However, the overall performance results of the majority voting ensemble are considerably below average for all the three datasets.

Figure 6.4 represents the overall performance of the majority voting AIS Ensemble System against the BC, HS, and PID datasets. The closely overlapping curves show that the majority voting AIS ensemble achieved on average almost identical classification performance outcomes for the three datasets. The results obtained from the BC dataset test minimally outperform the other two datasets.

Table 6-9: Performance Results for the MV AIS Ensemble System

Round #	Accuracy			Sensitivity			Specificity		
	BC	HS	PID	BC	HS	PID	BC	HS	PID
1	0.771	0.933	0.816	0.900	0.667	0.625	0.720	1.000	0.955
2	0.765	0.800	0.816	0.700	0.667	0.533	0.792	0.833	1.000
3	0.743	0.800	0.778	0.692	0.400	0.545	0.773	1.000	0.880
4	0.743	0.733	0.757	0.714	0.667	0.538	0.750	0.750	0.875
5	0.743	0.800	0.778	0.714	0.333	0.417	0.750	0.917	0.958
6	0.743	0.667	0.730	0.667	0.375	0.600	0.783	1.000	0.778
7	0.706	0.786	0.737	0.750	0.333	0.462	0.692	0.909	0.880
8	0.719	0.667	0.684	0.667	0.429	0.538	0.731	0.875	0.760
9	0.735	0.667	0.658	0.429	0.286	0.389	0.950	1.000	0.900
10	0.706	0.733	0.806	0.625	0.200	0.143	0.778	1.000	0.966

Table 6-10: The overall performance for the MV AIS Ensemble System

Round #	Total Performance		
	BC	HS	PID
1	2.391	2.600	2.395
2	2.256	2.300	2.349
3	2.208	2.200	2.203
4	2.207	2.150	2.170
5	2.207	2.050	2.153
6	2.192	2.042	2.108
7	2.148	2.028	2.078
8	2.116	1.970	1.983
9	2.114	1.952	1.947
10	2.109	1.933	1.914
Average	**2.195**	**2.123**	**2.130**

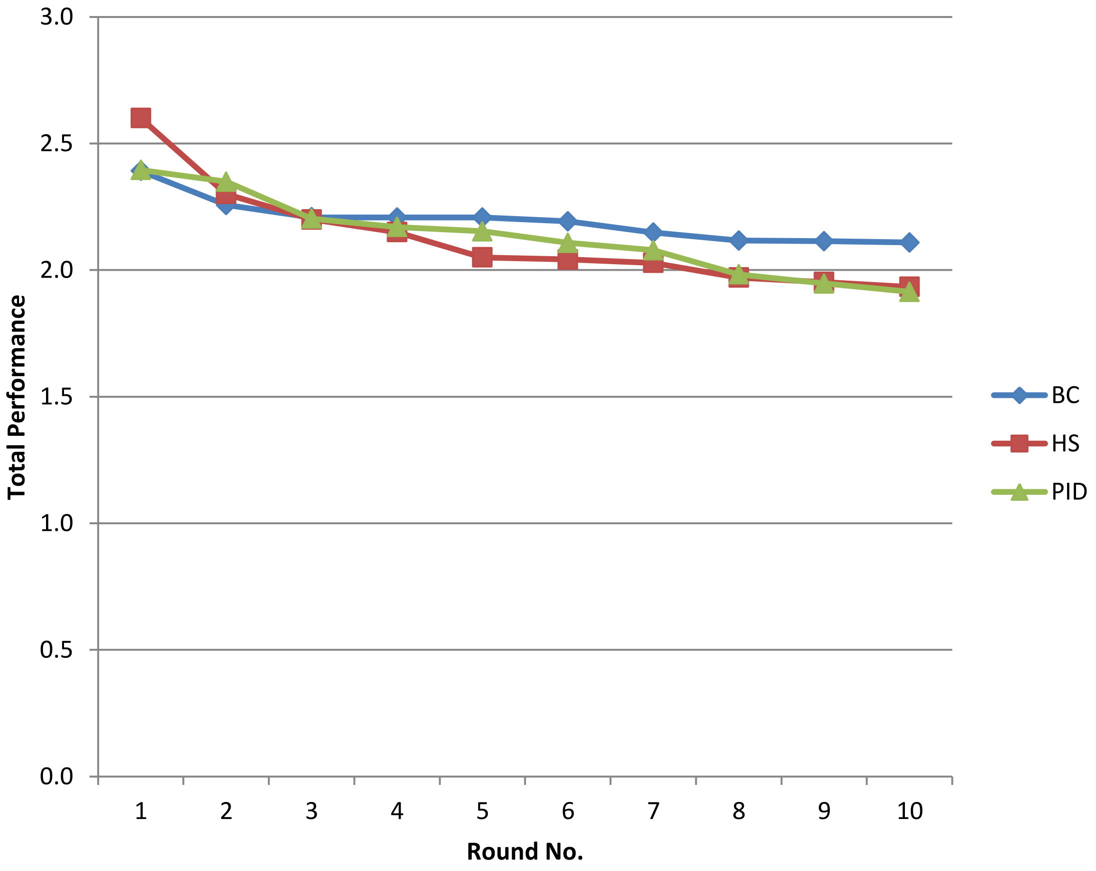


Figure 6-4: The overall performance of MV AIS Ensemble system

6.3.3. Experiment # 3: Weighted Average AIS Ensemble System

Similar to the previous experiment, the AIS ensemble with weighted average used as an aggregation procedure was tested against the same datasets. The detailed classification performance results for the weighted average AIS ensemble are shown in Table 6.11. For all the three datasets, the accuracy results for weighted average AIS ensemble varied between 60% and 81% except for the first sample, where it achieved 93% with the HS dataset. On the other hand, the sensitivity measures are considered low for most of the test runs. For the specificity measure, the AIS ensemble model achieved high results for both HS and PID datasets.

Table 6.12 and Figure 6.5 outline the total performance of the weighted average AIS ensemble with the three datasets used in this experiment. It can be noticed that the AIS ensemble systems achieved (to some extent) better total performance results with the BC dataset.

Figure 6.5 represents the overall performance of the weighted average AIS ensemble system against the BC, HS, and PID datasets. The figure shows some consistency of total performance of the weighted average AIS ensemble system against the BC dataset, and a slightly lower though similar trend against the PID dataset. It is evident that total performance against the HS dataset depicts wider variation with many data outliers.

Table 6-11: Performance Results for the WA AIS Ensemble System

Round #	Accuracy			Sensitivity			Specificity		
	BC	HS	PID	BC	HS	PID	BC	HS	PID
1	0.771	0.933	0.816	0.900	0.667	0.625	0.720	1.000	0.955
2	0.765	0.800	0.816	0.700	1.000	0.533	0.792	0.750	1.000
3	0.743	0.800	0.778	0.692	0.400	0.545	0.773	1.000	0.880
4	0.743	0.667	0.757	0.714	0.667	0.538	0.750	0.667	0.875
5	0.743	0.600	0.778	0.714	0.333	0.417	0.750	0.667	0.958
6	0.743	0.667	0.730	0.667	0.500	0.600	0.783	0.857	0.778
7	0.706	0.786	0.737	0.750	0.333	0.462	0.692	0.909	0.880
8	0.719	0.667	0.684	0.667	0.429	0.538	0.731	0.875	0.760
9	0.735	0.667	0.658	0.429	0.286	0.389	0.950	1.000	0.900
10	0.706	0.800	0.611	0.625	0.400	0.429	0.778	1.000	0.655

Table 6-12: The overall performance for the WA AIS Ensemble System

Round #	Total Performance		
	BC	**HS**	**PID**
1	2.391	2.600	2.395
2	2.256	2.550	2.349
3	2.208	2.200	2.203
4	2.207	2.000	2.170
5	2.207	1.600	2.153
6	2.192	2.024	2.108
7	2.148	2.028	2.078
8	2.116	1.970	1.983
9	2.114	1.952	1.947
10	2.109	2.200	1.695
Average	**2.195**	**2.112**	**2.108**

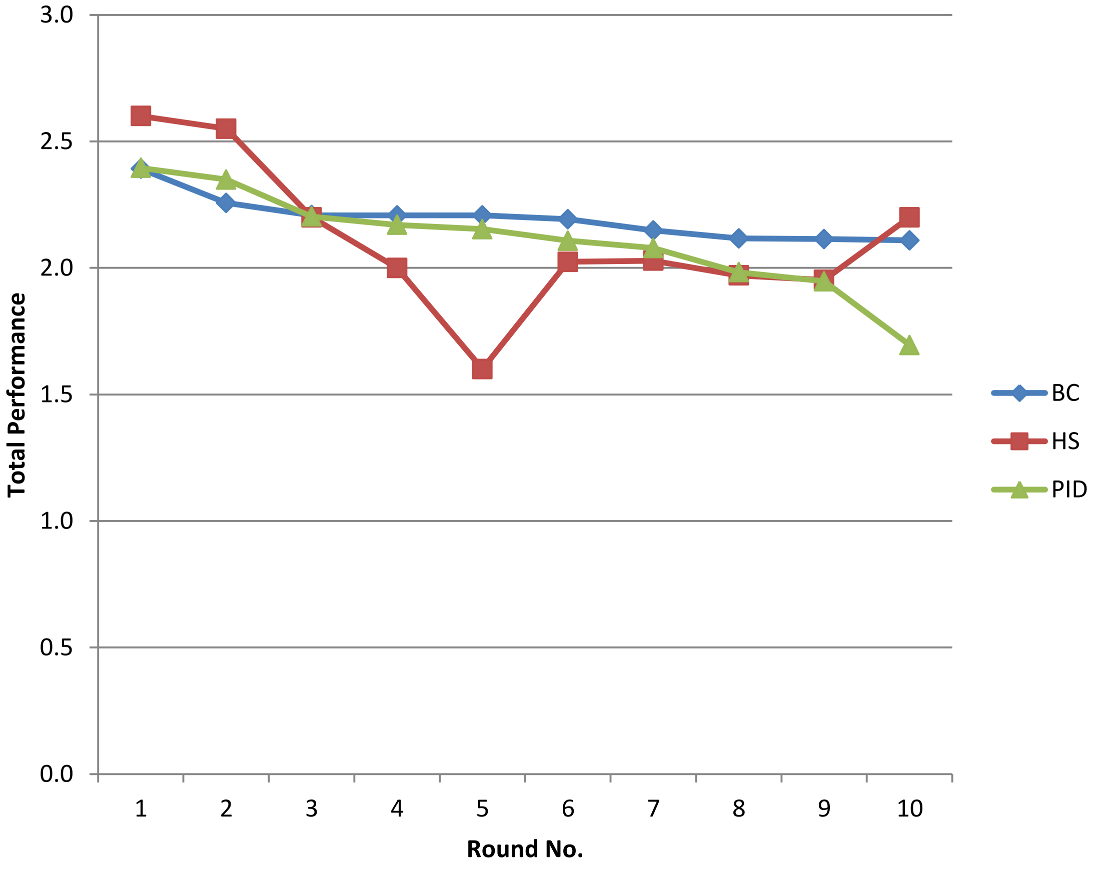


Figure 6-5: The overall performance of WA AIS Ensemble system

6.3.4. Experiment # 4: Weighted Average with PSO AIS Ensemble System

In this experiment, the proposed AIS ensemble with PSO optimizer was tested against the BC, HS and PID datasets. The weighted average aggregation procedure was used to combine the predicted outputs resulted from the base classifiers along with the PSO optimizer. The number of particles used in the optimization algorithm varied between 20 and 100 in each round, to search for an optimal solution within the full space. Tables 6.13, 6.14 and 6.15 highlight the detailed results of the classification performance measures of the weighted average AIS ensemble with PSO optimizer against the BC, HS, and PID datasets respectively.

Generally speaking, improved experimental results were obtained while testing the AIS ensemble with PSO optimizer. More specifically, the optimized AIS ensemble model succeeded in some cases in achieving perfect results by predicting the correct class for all the testing samples, especially with the HS dataset, as shown in Table 6.14. However, the results shown in Table 6.16 and Figure 6.6 highlight that the AIS ensemble system with PSO optimizer achieved a slight improvement on the average total performance compared to the best base classifier, which can be further enhanced.

Table 6-13: Performance Results for the WA_PSO AIS Ensemble System against BC dataset

Sample #	**# of Particles**	**WA_PSO AIS Ensemble**						
		Accuracy	**Sensitivity**	**Specificity**	**Total**	***Gbest***		
1	**20**	0.771	0.900	0.720	2.391	0.190	1.274	1.148
	30	0.950	0.500	1.000	2.450	0.857	0.745	1.096
	40	0.771	0.900	0.720	2.391	0.273	1.232	0.706
	50	0.771	0.900	0.720	2.391	0.274	1.427	1.359
	70	0.771	0.900	0.720	2.391	0.971	1.105	1.108
	100	0.950	0.500	1.000	2.450	0.116	0.092	0.549
2	**20**	0.807	0.667	0.864	2.337	0.131	0.217	0.065
	30	0.794	0.600	0.875	2.269	1.194	0.826	1.225
	40	0.794	0.600	0.875	2.269	0.815	0.636	0.591
	50	0.794	0.600	0.875	2.269	1.137	0.833	1.306
	70	0.807	0.667	0.864	2.337	0.856	1.423	0.899

Sample #	# of Particles	WA_PSO AIS Ensemble						
		Accuracy	Sensitivity	Specificity	Total	*Gbest*		
	100	0.794	0.600	0.875	2.269	1.326	1.126	0.768
3	20	0.743	0.692	0.773	2.208	0.739	1.073	0.880
	30	0.743	0.692	0.773	2.208	0.032	0.610	0.413
	40	0.743	0.692	0.773	2.208	0.756	1.004	1.413
	50	0.743	0.692	0.773	2.208	0.269	1.256	0.259
	70	0.743	0.692	0.773	2.208	1.213	1.353	1.285
	100	0.743	0.692	0.773	2.208	1.117	1.478	0.084
4	20	0.800	0.714	0.821	2.336	0.883	0.589	2.530
	30	0.771	0.714	0.786	2.271	1.450	1.145	0.816
	40	0.771	0.714	0.786	2.271	1.371	1.325	0.649
	50	0.771	0.714	0.786	2.271	1.294	1.179	0.641
	70	0.771	0.714	0.786	2.271	1.328	0.979	1.176
	100	0.794	0.714	0.815	2.323	0.101	0.217	1.275
5	20	0.743	0.714	0.750	2.207	0.934	1.052	1.063
	30	0.743	0.714	0.750	2.207	0.182	1.342	1.492
	40	0.743	0.714	0.750	2.207	0.214	0.366	0.640
	50	0.743	0.714	0.750	2.207	0.541	1.476	0.559
	70	0.743	0.714	0.750	2.207	0.840	0.967	0.904
	100	0.743	0.714	0.750	2.207	1.187	1.387	1.062
6	20	0.743	0.667	0.783	2.192	1.155	1.200	0.280
	30	0.743	0.667	0.783	2.192	0.549	1.391	1.367
	40	0.743	0.667	0.783	2.192	0.402	0.464	0.582
	50	0.743	0.667	0.783	2.192	0.516	0.621	0.252
	70	0.743	0.667	0.783	2.192	0.592	0.796	0.768
	100	0.743	0.667	0.783	2.192	0.520	0.658	1.046
7	20	0.706	0.750	0.692	2.148	1.139	1.214	1.128
	30	0.706	0.750	0.692	2.148	0.413	1.418	1.174
	40	0.706	0.750	0.692	2.148	0.739	0.880	0.429

Sample #	# of Particles	WA_PSO AIS Ensemble						
		Accuracy	Sensitivity	Specificity	Total	*Gbest*		
	50	0.706	0.750	0.692	2.148	0.413	1.495	0.164
	70	0.706	0.750	0.692	2.148	0.262	1.144	0.930
	100	0.706	0.750	0.692	2.148	0.719	0.936	1.394
8	20	0.781	0.500	0.846	2.128	1.224	0.921	0.807
	30	0.781	0.500	0.846	2.128	1.227	1.027	0.427
	40	0.786	0.600	0.826	2.212	1.018	1.688	0.278
	50	0.781	0.500	0.846	2.128	1.321	1.064	0.283
	70	0.781	0.500	0.846	2.128	0.860	0.541	1.379
	100	0.781	0.500	0.846	2.128	0.999	0.629	0.154
9	20	0.735	0.429	0.950	2.114	1.147	1.371	1.004
	30	0.735	0.429	0.950	2.114	0.192	0.847	0.308
	40	0.735	0.429	0.950	2.114	1.136	1.147	1.110
	50	0.735	0.429	0.950	2.114	0.308	0.772	0.950
	70	0.735	0.429	0.950	2.114	0.721	1.022	0.806
	100	0.735	0.429	0.950	2.114	1.207	1.421	0.679
10	20	0.706	0.625	0.778	2.109	0.187	1.125	1.019
	30	0.706	0.625	0.778	2.109	1.108	1.393	1.025
	40	0.706	0.625	0.778	2.109	0.240	1.099	1.344
	50	0.706	0.625	0.778	2.109	0.430	0.921	1.081
	70	0.706	0.625	0.778	2.109	0.206	0.946	0.896
	100	0.706	0.625	0.778	2.109	0.475	1.487	0.013

It is clear that in sample 1, for example, the WA_PSO AIS ensemble achieved 95% accuracy when the number of particles is 30; whilst the sensitivity and specificity are 0.5 and 1 (respectively). The same is also true when the number of particles is 100. However, the average accuracy achieved across all the samples is only 76%, the average sensitivity is 0.65 and average specificity 0.8.

Table 6-14: Performance Results for the WA_PSO AIS Ensemble System against HS dataset

Sample #	**# of Particles**	**WA_PSO AIS Ensemble**						
		Accuracy	**Sensitivity**	**Specificity**	**Total**	***Gbest***		
1	**20**	0.933	0.667	1.000	2.600	0.610	0.676	0.862
	30	0.933	0.667	1.000	2.600	0.610	0.208	0.913
	40	0.933	0.667	1.000	2.600	1.090	0.930	1.257
	50	0.933	0.667	1.000	2.600	0.077	0.014	1.326
	70	0.933	0.667	1.000	2.600	0.614	1.293	1.352
	100	0.933	0.667	1.000	2.600	1.371	1.131	1.320
2	**20**	0.933	1.000	0.917	2.850	0.457	0.052	0.934
	30	0.933	0.667	1.000	2.600	0.248	0.271	0.991
	40	0.933	1.000	0.917	2.850	0.218	0.027	0.438
	50	0.933	0.667	1.000	2.600	0.288	0.312	1.159
	70	0.933	1.000	0.917	2.850	1.138	0.105	2.206
	100	1.000	1.000	1.000	3.000	0.754	0.606	0.747
3	**20**	0.800	0.400	1.000	2.200	0.079	1.323	0.926
	30	0.800	0.400	1.000	2.200	0.382	0.404	0.873
	40	0.800	0.400	1.000	2.200	0.190	1.148	0.878
	50	0.800	0.400	1.000	2.200	0.585	0.452	1.328
	70	0.800	0.400	1.000	2.200	1.222	1.261	1.280
	100	0.800	0.400	1.000	2.200	0.820	1.052	0.002
4	**20**	0.733	0.667	0.750	2.150	1.146	1.446	0.297
	30	0.733	0.667	0.750	2.150	0.692	1.285	1.343
	40	0.733	0.667	0.750	2.150	0.934	1.399	1.445
	50	0.733	0.667	0.750	2.150	0.945	1.225	0.878
	70	0.733	0.667	0.750	2.150	1.194	1.141	1.157
	100	0.733	0.667	0.750	2.150	0.210	0.851	0.701
5	**20**	0.800	0.333	0.917	2.050	1.222	0.984	0.658
	30	0.800	0.333	0.917	2.050	0.243	0.647	1.170
	40	0.800	0.333	0.917	2.050	0.904	0.741	1.459

Sample #	**# of Particles**	**WA_PSO AIS Ensemble**						
		Accuracy	**Sensitivity**	**Specificity**	**Total**	***Gbest***		
	50	0.800	0.333	0.917	2.050	1.359	1.020	1.191
	70	0.800	0.333	0.917	2.050	1.164	1.328	0.647
	100	0.800	0.333	0.917	2.050	0.382	0.522	0.828
6	**20**	0.818	0.800	0.833	2.452	0.148	0.855	1.728
	30	0.733	0.625	0.857	2.215	0.933	0.603	0.780
	40	0.733	0.625	0.857	2.215	1.132	0.631	1.109
	50	0.733	0.625	0.857	2.215	1.093	0.671	0.878
	70	0.733	0.625	0.857	2.215	0.900	0.461	0.957
	100	0.733	0.625	0.857	2.215	0.692	0.142	1.166
7	**20**	0.857	0.333	1.000	2.190	0.029	0.146	0.175
	30	0.857	0.333	1.000	2.190	0.405	0.957	0.807
	40	0.857	0.333	1.000	2.190	0.283	0.458	1.108
	50	0.786	0.667	0.818	2.271	0.580	0.874	0.397
	70	0.857	0.333	1.000	2.190	0.207	0.561	1.148
	100	0.857	0.333	1.000	2.190	0.403	0.186	0.849
8	**20**	0.733	0.714	0.750	2.198	0.956	0.467	0.730
	30	0.818	0.800	0.833	2.452	0.381	1.410	2.416
	40	0.733	0.714	0.750	2.198	0.841	0.246	0.868
	50	0.733	0.714	0.750	2.198	0.749	0.073	1.021
	70	0.733	0.714	0.750	2.198	0.947	0.070	1.324
	100	0.733	0.714	0.750	2.198	1.092	0.256	1.202
9	**20**	0.667	0.857	0.500	2.024	1.055	1.820	1.288
	30	0.667	0.571	0.750	1.988	1.105	0.653	0.766
	40	0.667	0.571	0.750	1.988	1.539	0.864	1.166
	50	0.733	0.571	0.875	2.180	1.086	1.128	0.036
	70	0.667	0.286	1.000	1.952	0.202	1.251	0.637
	100	0.667	0.857	0.500	2.024	1.230	1.506	0.438
10	**20**	0.867	0.800	0.900	2.567	1.642	1.424	0.421

Sample #	# of Particles	WA_PSO AIS Ensemble						
		Accuracy	Sensitivity	Specificity	Total	*Gbest*		
	30	0.867	0.800	0.900	2.567	0.938	0.248	1.244
	40	0.867	0.800	0.900	2.567	1.293	0.682	1.026
	50	0.867	0.800	0.900	2.567	1.072	0.543	0.953
	70	0.867	0.800	0.900	2.567	0.263	0.127	0.208
	100	0.800	1.000	0.778	2.578	0.112	0.502	0.901

In Table 6.14 (showing the Performance Results for the AIS Ensemble with PSO optimizer system against the HS dataset), it can be seen that the optimized AIS ensemble model succeeded in some cases in achieving perfect results by predicting the correct class for all the test samples. This is evident in sample 2; when number of particles is 100, encouraging performance results were obtained in samples 2, 8, and 10. Additionally, the figures for the three measures of accuracy, sensitivity, and specificity in samples 1, 3, 4 and 5, are constant within the sample, regardless of the number of particles. For example, in sample 1, the optimized AIS ensemble achieved 93.3% accuracy, 0.667 sensitivity, and a perfect 1.000 specificity for the number of particles spanning the range 30 to 100.

Table 6-15: Performance Results for the WA_PSO AIS Ensemble System against PID dataset

Sample #	# of Particles	WA_PSO AIS Ensemble						
		Accuracy	Sensitivity	Specificity	Total	*Gbest*		
1	**20**	0.737	0.438	0.955	2.129	0.710	0.686	0.212
	30	0.737	0.438	0.955	2.129	0.180	0.176	0.700
	40	0.737	0.438	0.955	2.129	1.454	1.392	0.811
	50	0.737	0.438	0.955	2.129	1.064	0.986	1.100
	70	0.737	0.438	0.955	2.129	0.530	0.508	0.484
	100	0.737	0.438	0.955	2.129	1.463	1.327	1.451
2	**20**	0.790	0.467	1.000	2.256	0.806	0.567	0.532
	30	0.790	0.467	1.000	2.256	1.137	0.824	0.932
	40	0.790	0.467	1.000	2.256	0.391	0.310	0.827
	50	0.790	0.467	1.000	2.256	1.435	1.202	0.763

Sample #	# of Particles	WA_PSO AIS Ensemble						
		Accuracy	Sensitivity	Specificity	Total	*Gbest*		
	70	0.790	0.467	1.000	2.256	1.316	1.031	1.381
	100	0.790	0.467	1.000	2.256	0.355	0.334	0.113
3	**20**	0.778	0.364	0.960	2.101	1.359	0.054	0.572
	30	0.778	0.364	0.960	2.101	0.467	0.273	0.363
	40	0.778	0.364	0.960	2.101	0.892	0.160	0.989
	50	0.778	0.364	0.960	2.101	1.359	0.382	0.688
	70	0.778	0.364	0.960	2.101	0.772	0.220	0.267
	100	0.778	0.364	0.960	2.101	0.727	0.187	0.941
4	**20**	0.757	0.462	0.917	2.135	0.407	0.441	0.058
	30	0.757	0.462	0.917	2.135	0.835	0.923	0.824
	40	0.757	0.462	0.917	2.135	1.329	1.330	0.925
	50	0.757	0.462	0.917	2.135	0.418	0.462	0.711
	70	0.757	0.462	0.917	2.135	0.781	0.861	0.154
	100	0.757	0.462	0.917	2.135	1.327	1.431	1.172
5	**20**	0.750	0.250	1.000	2.000	0.976	1.464	1.326
	30	0.750	0.250	1.000	2.000	0.502	0.989	0.703
	40	0.750	0.250	1.000	2.000	0.450	0.897	1.144
	50	0.750	0.250	1.000	2.000	0.736	1.185	0.992
	70	0.750	0.250	1.000	2.000	0.190	1.221	0.526
	100	0.750	0.250	1.000	2.000	0.820	1.052	0.002
6	**20**	0.811	0.600	0.889	2.300	1.832	1.750	1.403
	30	0.811	0.600	0.889	2.300	1.436	1.425	0.224
	40	0.811	0.600	0.889	2.300	1.170	1.158	0.089
	50	0.811	0.600	0.889	2.300	1.045	0.953	0.669
	70	0.811	0.600	0.889	2.300	0.822	0.797	0.354
	100	0.811	0.600	0.889	2.300	0.418	0.394	1.314
7	**20**	0.711	0.308	0.920	1.938	0.986	0.830	0.919
	30	0.711	0.308	0.920	1.938	0.839	0.610	0.208

Sample #	# of Particles	WA_PSO AIS Ensemble						
		Accuracy	Sensitivity	Specificity	Total	*Gbest*		
	40	0.711	0.308	0.920	1.938	1.144	0.290	0.864
	50	0.711	0.308	0.920	1.938	1.213	0.297	0.362
	70	0.711	0.308	0.920	1.938	1.154	0.251	0.668
	100	0.711	0.308	0.920	1.938	1.316	0.171	0.052
8	20	0.737	0.462	0.880	2.078	0.920	0.911	0.856
	30	0.737	0.462	0.880	2.078	0.931	0.917	0.571
	40	0.737	0.462	0.880	2.078	1.435	1.428	0.955
	50	0.737	0.462	0.880	2.078	0.676	0.613	0.297
	70	0.737	0.462	0.880	2.078	0.879	0.816	0.302
	100	0.737	0.462	0.880	2.078	0.747	0.697	0.550
9	20	0.658	0.333	0.950	1.941	1.460	1.326	1.365
	30	0.658	0.333	0.950	1.941	0.311	0.286	1.241
	40	0.658	0.333	0.950	1.941	0.901	0.863	0.066
	50	0.658	0.333	0.950	1.941	1.409	1.345	0.980
	70	0.658	0.333	0.950	1.941	0.530	0.477	1.029
	100	0.658	0.333	0.950	1.941	1.162	1.152	0.144
10	20	0.722	0.286	0.828	1.836	0.387	1.080	0.208
	30	0.722	0.286	0.828	1.836	0.309	1.222	0.830
	40	0.722	0.286	0.828	1.836	0.029	1.300	0.103
	50	0.800	0.200	0.950	1.950	0.527	0.863	1.800
	70	0.800	0.200	0.950	1.950	0.704	0.701	0.063
	100	0.800	0.200	0.950	1.950	0.569	0.677	1.105

In Table 6.15 (showing the performance results for the AIS ensemble system with PSO optimizer against the PID dataset), the performance accuracy did not exceed 81.1%, which was achieved only in sample 6. However, except for sample no. 10, the optimized AIS ensemble model succeeded in all other cases in achieving persistent performance results, as evidenced in samples 1 through 9, where the figures for the three measures of accuracy, sensitivity, and specificity are constant within the sample, regardless of the number of particles. For example, in sample 2, the AIS ensemble system achieved 79% accuracy, 0.467 sensitivity, and a perfect 1.000 specificity for the number of particles spanning the range 30 to 100. However, the average accuracy achieved across all the samples is only 74.9%, the average sensitivity is 0.390 and average specificity a good 0.936.

Table 6-16: The overall performance for the WA_PSO AIS Ensemble System

Round #	Total Performance		
	BC	HS	PID
1	2.391	2.600	2.129
2	2.269	2.850	2.256
3	2.208	2.200	2.101
4	2.271	2.150	2.135
5	2.207	2.050	2.000
6	2.192	2.215	2.300
7	2.148	2.190	1.938
8	2.128	2.198	2.078
9	2.114	1.988	1.941
10	2.109	2.567	1.836
Average	**2.204**	**2.301**	**2.071**

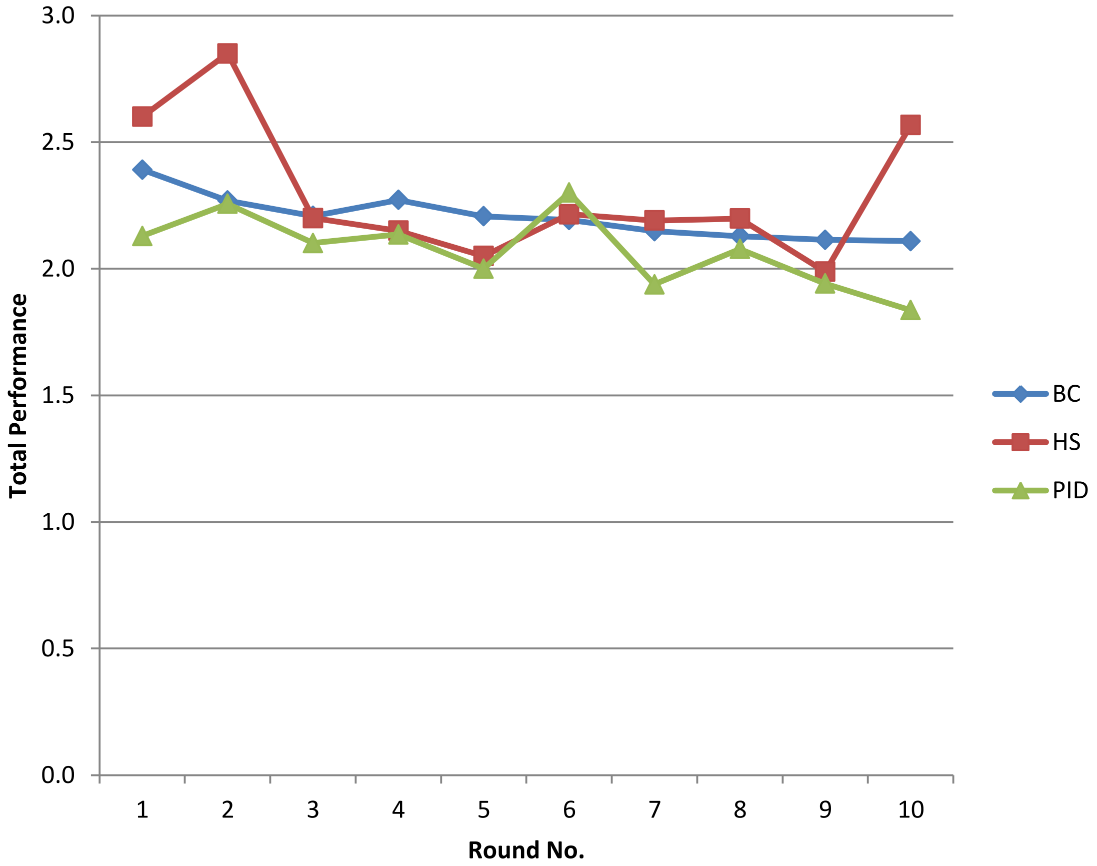


Figure 6-6: The overall performance of WA_PSO AIS Ensemble system

Figure 6.6 represents the overall performance of the weighted average with PSO AIS ensemble system against the BC, HS, and PID datasets. The figure shows some consistency of total performance of the optimized AIS ensemble system against the BC dataset, and a slightly lower though similar trend against the PID dataset. It is evident that total performance against the HS dataset is the highest, although it depicts wider variation because of some data outliers.

6.3.5. Experiment # 5: Weighted Average with NFS AIS Ensemble System

The AIS based ensemble with neuro-fuzzy detector feature was proposed in Chapter 5. In this experiment, four medical datasets were used to test the effectiveness of this ensemble model. In addition to the three datasets introduced in section 6.2 of this chapter, the BC dataset presented in Chapter 3 was also used to test the detector based AIS ensemble.

Table 6.17 illustrates the detailed classification performance results of the AIS ensemble model with neuro-fuzzy detector for the four datasets, while Table 6.18 and Figure 6.7 highlight the total performance results. It can be noticed from the two tables that the proposed ensemble systems has achieved very high accuracy, sensitivity and specificity rates for the breast cancer compared to the other datasets. Also, the results obtained for testing the ensemble model with the BC, HS, and PID datasets are relatively similar, as shown in Figure 6.7. Furthermore, the experimental results demonstrate clearly that AIS ensemble with neuro-fuzzy detector works effectively with high dimensional datasets such as BC and and PID datasets, and reasonable classification performance was achieved. However, this can be further improved by integrating the proposed architecture with other optimization techniques.

Table 6-17: Performance Results for the WA_NFS AIS Ensemble System

Round #	Accuracy				Sensitivity				Specificity			
	WBC	BC	HS	PID	WBC	BC	HS	PID	WBC	BC	HS	PID
1	0.971	0.771	0.933	0.816	0.944	0.900	0.667	0.625	1.000	0.720	1.000	0.955
2	0.971	0.765	0.800	0.816	0.929	0.700	0.667	0.533	1.000	0.792	0.833	1.000
3	0.971	0.743	0.800	0.778	0.923	0.692	0.400	0.545	1.000	0.773	1.000	0.880
4	0.971	0.743	0.733	0.757	0.917	0.714	0.667	0.538	1.000	0.750	0.750	0.875
5	0.971	0.743	0.800	0.778	0.917	0.714	0.333	0.417	1.000	0.750	0.917	0.958
6	0.971	0.743	0.667	0.730	0.917	0.667	0.375	0.600	1.000	0.783	1.000	0.778
7	0.971	0.706	0.786	0.737	0.909	0.750	0.333	0.462	1.000	0.692	0.909	0.880
8	0.971	0.719	0.667	0.684	0.900	0.667	0.429	0.538	1.000	0.731	0.875	0.760
9	0.971	0.735	0.667	0.658	0.900	0.429	0.286	0.389	1.000	0.950	1.000	0.900
10	0.971	0.706	0.733	0.806	0.889	0.625	0.200	0.143	1.000	0.778	1.000	0.966

Table 6-18: The overall performance for the WA_NFS AIS Ensemble System

Round #	Total Performance			
	WBC	BC	HS	PID
1	2.915	2.391	2.600	2.395
2	2.899	2.256	2.300	2.349
3	2.895	2.208	2.200	2.203
4	2.888	2.207	2.150	2.170
5	2.887	2.207	2.050	2.153
6	2.887	2.192	2.042	2.108
7	2.881	2.148	2.028	2.078
8	2.871	2.116	1.970	1.983
9	2.871	2.114	1.952	1.947
10	2.860	2.109	1.933	1.914
Average	**2.885**	**2.195**	**2.123**	**2.130**

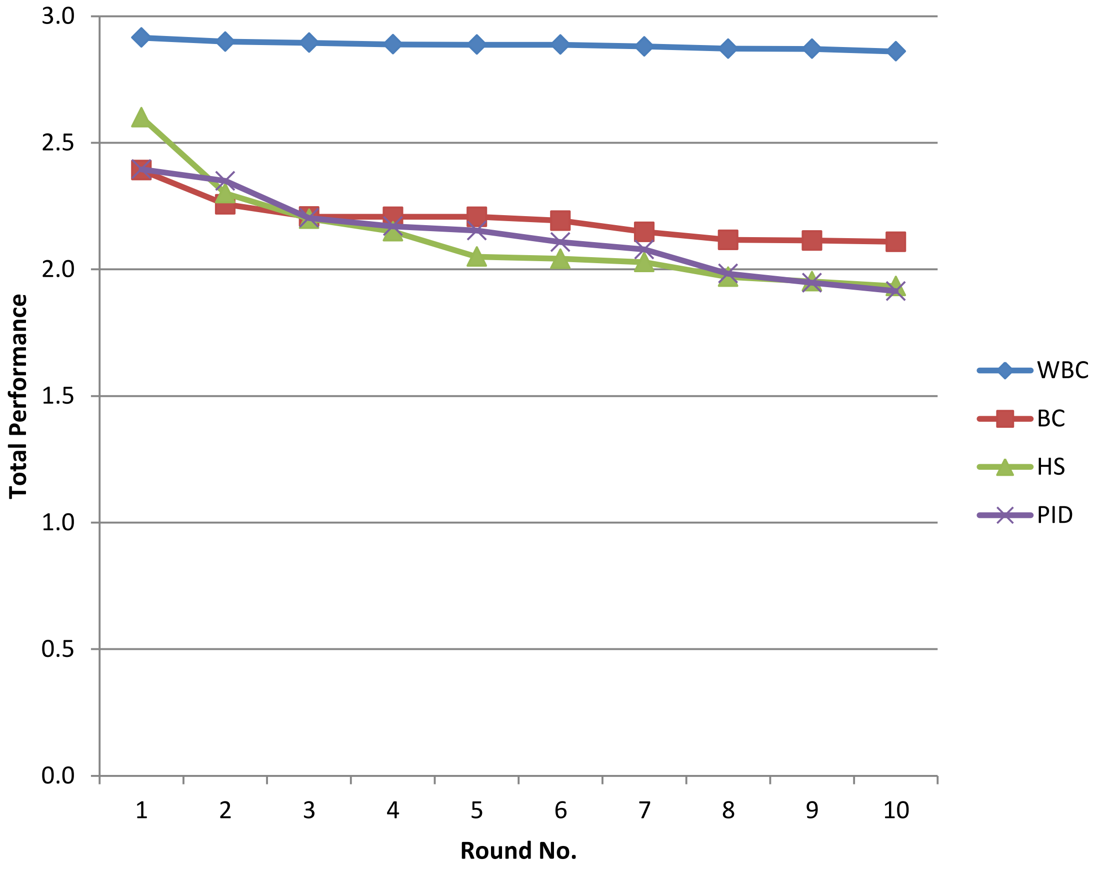


Figure 6-7: The overall performance of WA_NFS AIS Ensemble system

6.3.6. Experiment # 6: The Adaptive Learning AIS Based Ensemble System

The proposed AIS ensemble has a unique architecture that is based on adaptive learning neuro-fuzzy detector to enhance the classification performance. It has been further enhanced based on the PSO technique to further improve the overall performance of the architecture. These innovative solutions are combined together in an effective, computationally efficient architecture. In this experiment, various high-dimensional datasets are used to evaluate the performance of the proposed solution, where the results demonstrate that the performance of the new proposed system outperforms the conventional AIS based algorithms. Tables 6.19, 6.20, 6.21, and 6.22 represent the detailed experimental results of the adaptive learning AIS ensemble with WBC, BC, HS, and PID datasets respectively.

The weighted average combining method has been used as an aggregation procedure in this architecture. The number of particles used in the optimization algorithm varied between 20 and 100 in each round, to test PSO optimizer capability for achieving an optimal solution.

Table 6-19: Performance Results for the WA_PSO_NFS AIS Ensemble System against WBC dataset

Sample #	# of Particles	WA_PSO_NFS AIS Ensemble						
		Accuracy	Sensitivity	Specificity	Total	*Gbest*		
1	20	0.971	0.944	1.000	2.915	0.272	0.353	0.622
	30	0.971	0.944	1.000	2.915	1.428	1.475	1.031
	40	0.971	0.944	1.000	2.915	1.342	0.906	1.464
	50	0.971	0.944	1.000	2.915	0.529	0.594	0.709
	70	0.971	0.944	1.000	2.915	0.763	0.541	1.035
	100	0.971	0.944	1.000	2.915	0.612	0.839	0.867
2	20	1.000	1.000	1.000	3.000	0.402	0.239	0.329
	30	0.971	0.929	1.000	2.899	0.528	0.606	0.589
	40	0.971	0.929	1.000	2.899	1.039	1.017	0.614
	50	0.971	0.929	1.000	2.899	1.433	0.966	1.222
	70	1.000	1.000	1.000	3.000	0.490	0.221	0.464
	100	1.000	1.000	1.000	3.000	0.416	0.216	1.474
3	20	0.971	0.923	1.000	2.895	0.805	0.734	1.053
	30	0.971	0.923	1.000	2.895	0.767	0.394	1.439
	40	1.000	1.000	1.000	3.000	0.223	0.120	0.850
	50	0.971	0.923	1.000	2.895	1.010	1.351	1.357
	70	0.971	0.923	1.000	2.895	1.414	1.327	0.221
	100	0.971	0.923	1.000	2.895	1.209	1.236	0.096
4	20	0.971	0.917	1.000	2.888	0.263	0.716	1.226
	30	0.971	0.917	1.000	2.888	0.940	0.434	1.277
	40	0.971	0.917	1.000	2.888	1.027	1.147	1.400
	50	0.971	0.917	1.000	2.888	0.684	0.349	1.041
	70	0.971	0.917	1.000	2.888	0.594	0.753	0.828
	100	0.971	0.917	1.000	2.888	0.158	0.497	0.957
5	20	0.971	0.917	1.000	2.887	0.974	1.492	1.414
	30	0.971	0.917	1.000	2.887	0.846	0.721	1.094
	40	0.971	0.917	1.000	2.887	0.785	0.803	0.503

Sample #	# of Particles	WA_PSO_NFS AIS Ensemble						
		Accuracy	Sensitivity	Specificity	Total	*Gbest*		
	50	0.971	0.917	1.000	2.887	0.899	1.022	0.999
	70	0.971	0.917	1.000	2.887	1.023	1.445	1.490
	100	0.971	0.917	1.000	2.887	0.933	1.046	1.419
6	20	0.971	0.917	1.000	2.887	1.079	0.884	1.016
	30	0.971	0.917	1.000	2.887	0.812	0.908	0.482
	40	0.971	0.917	1.000	2.887	0.859	0.751	1.459
	50	0.971	0.917	1.000	2.887	0.545	0.515	0.881
	70	0.971	0.917	1.000	2.887	0.752	0.721	0.105
	100	0.971	0.917	1.000	2.887	0.648	0.469	1.465
7	20	0.971	0.909	1.000	2.881	0.508	0.515	1.459
	30	0.971	0.909	1.000	2.881	0.599	0.970	1.274
	40	0.971	0.909	1.000	2.881	0.981	0.556	1.409
	50	0.971	0.909	1.000	2.881	0.092	0.151	0.294
	70	0.971	0.909	1.000	2.881	1.044	1.276	1.422
	100	0.971	0.909	1.000	2.881	0.670	0.735	0.892
8	20	0.971	0.900	1.000	2.871	0.393	0.106	0.897
	30	0.971	0.900	1.000	2.871	0.448	0.761	1.366
	40	0.971	0.900	1.000	2.871	0.107	0.785	0.463
	50	0.971	0.900	1.000	2.871	1.359	1.020	1.191
	70	0.971	0.900	1.000	2.871	1.230	1.298	1.153
	100	0.971	0.900	1.000	2.871	0.685	1.041	0.038
9	20	0.971	0.900	1.000	2.871	0.163	0.343	0.900
	30	0.971	0.900	1.000	2.871	0.944	1.013	1.346
	40	1.000	1.000	1.000	3.000	0.480	0.034	1.116
	50	0.971	0.900	1.000	2.871	1.146	0.923	0.692
	70	0.971	0.900	1.000	2.871	0.365	0.283	0.963
	100	0.971	0.900	1.000	2.871	1.431	1.201	0.947
10	20	0.971	0.889	1.000	2.860	0.913	0.976	1.056

Sample #	# of Particles	WA_PSO_NFS AIS Ensemble						
		Accuracy	Sensitivity	Specificity	Total	*Gbest*		
	30	0.971	0.889	1.000	2.860	0.434	0.387	1.321
	40	0.971	0.889	1.000	2.860	0.964	1.368	1.345
	50	0.971	0.889	1.000	2.860	0.188	0.375	0.796
	70	0.971	0.889	1.000	2.860	1.251	1.384	1.059
	100	0.971	0.889	1.000	2.860	0.777	0.819	1.080

In Table 6.19 showing the performance results for the adaptive learning AIS ensemble system against the WBC dataset, the performance accuracy consistently equals 97.1% and achieves 100% in 8% of the instances. Performance sensitivity ranges between 0.889 and 1.000 averaging 0.921. Performance specificity is 1.000 in all cases. These results clearly demonstrate that this approach outperforms conventional AIS algorithms for the WBC dataset.

Table 6-20: Performance Results for the WA_PSO_NFS AIS Ensemble System against BC dataset

Sample #	# of Particles	WA_PSO_NFS AIS Ensemble						
		Accuracy	Sensitivity	Specificity	Total	*Gbest*		
1	**20**	0.771	0.900	0.720	2.391	1.015	1.457	0.404
	30	0.771	0.900	0.720	2.391	0.739	0.756	1.395
	40	0.771	0.900	0.720	2.391	0.610	0.862	1.495
	50	0.771	0.900	0.720	2.391	0.215	0.365	0.799
	70	0.771	0.900	0.720	2.391	0.963	1.362	1.069
	100	0.771	0.900	0.720	2.391	0.290	1.117	1.478
2	**20**	0.794	0.600	0.875	2.269	1.436	0.983	1.064
	30	0.794	0.600	0.875	2.269	0.467	0.273	0.363
	40	0.781	0.667	0.826	2.274	0.465	1.168	0.181
	50	0.794	0.600	0.875	2.269	1.044	0.691	0.477
	70	0.794	0.600	0.875	2.269	0.801	0.186	1.368
	100	0.794	0.600	0.875	2.269	0.727	0.187	0.941
3	**20**	1.000	1.000	1.000	3.000	0.649	0.885	0.306
	30	0.743	0.692	0.773	2.208	0.693	1.271	1.006

Sample #	# of Particles	WA_PSO_NFS AIS Ensemble						
		Accuracy	Sensitivity	Specificity	Total	*Gbest*		
	40	0.743	0.692	0.773	2.208	0.366	0.980	1.046
	50	1.000	1.000	1.000	3.000	2.109	1.860	0.344
	70	0.743	0.692	0.773	2.208	1.397	1.496	0.896
	100	0.743	0.692	0.773	2.208	0.261	0.430	0.447
4	**20**	0.771	0.714	0.786	2.271	1.144	0.575	1.202
	30	0.800	0.714	0.821	2.336	1.617	1.363	0.354
	40	0.800	0.714	0.821	2.336	1.327	0.892	0.197
	50	0.800	0.714	0.821	2.336	1.571	1.071	0.209
	70	0.771	0.714	0.786	2.271	1.304	0.613	1.264
	100	0.955	1.000	0.952	2.907	0.105	0.103	0.004
5	**20**	0.743	0.714	0.750	2.207	0.472	0.942	0.946
	30	0.743	0.714	0.750	2.207	0.522	1.419	0.227
	40	0.944	1.000	0.941	2.886	1.307	1.537	0.134
	50	0.643	1.000	0.615	2.258	0.241	0.386	0.225
	70	0.743	0.714	0.750	2.207	1.335	1.207	0.924
	100	0.743	0.714	0.750	2.207	0.940	0.438	1.339
6	**20**	0.771	0.917	0.696	2.384	0.676	1.534	1.407
	30	0.743	0.667	0.783	2.192	0.225	0.832	1.321
	40	0.771	0.833	0.739	2.344	0.403	0.966	0.862
	50	0.771	0.917	0.696	2.384	0.263	0.797	0.907
	70	0.771	0.917	0.696	2.384	0.490	1.266	1.308
	100	0.743	0.667	0.783	2.192	0.841	1.281	0.198
7	**20**	0.706	0.750	0.692	2.148	0.604	1.335	0.488
	30	0.706	0.750	0.692	2.148	0.086	0.419	0.498
	40	0.706	0.750	0.692	2.148	0.169	0.909	1.030
	50	0.706	0.750	0.692	2.148	1.284	1.333	0.075
	70	0.706	0.750	0.692	2.148	0.308	1.428	0.261
	100	0.706	0.750	0.692	2.148	0.851	0.701	0.570

Sample #	# of Particles	WA_PSO_NFS AIS Ensemble						
		Accuracy	Sensitivity	Specificity	Total	*Gbest*		
8	**20**	0.781	0.500	0.846	2.128	0.664	0.165	1.077
	30	0.781	0.500	0.846	2.128	0.962	0.594	0.375
	40	0.781	0.500	0.846	2.128	1.464	1.074	0.343
	50	0.759	0.600	0.792	2.150	0.770	1.586	0.658
	70	0.781	0.500	0.846	2.128	0.656	0.112	1.120
	100	0.781	0.500	0.846	2.128	0.852	0.697	0.023
9	**20**	0.735	0.429	0.950	2.114	0.525	0.902	0.622
	30	0.735	0.429	0.950	2.114	0.328	0.818	0.313
	40	0.735	0.429	0.950	2.114	0.287	0.443	0.193
	50	0.735	0.429	0.950	2.114	0.709	1.118	0.192
	70	0.735	0.429	0.950	2.114	0.238	1.149	0.223
	100	0.735	0.429	0.950	2.114	0.364	0.405	0.472
10	**20**	0.706	0.625	0.778	2.109	0.844	0.883	0.075
	30	0.706	0.625	0.778	2.109	0.512	0.869	1.479
	40	0.706	0.625	0.778	2.109	0.214	0.410	0.832
	50	0.706	0.625	0.778	2.109	0.283	0.322	0.439
	70	0.706	0.625	0.778	2.109	0.341	0.356	1.118
	100	0.706	0.625	0.778	2.109	1.416	1.240	0.445

Table 6.20 shows the performance results for the adaptive learning AIS ensemble system against the BC dataset. For this dataset, the performance accuracy ranges from 64.3% to 100% averaging 76.5%. Performance sensitivity ranges between 0.429 and 1.000 averaging 0.702. Performance specificity varies from 0.615 to 1.000 averaging 0.802. It is interesting to note that optimal results were obtained for sample 3, when number of particles is 20 and 40, by predicting the correct class for most test samples. Also, the ensemble model achieved encouraging performance results for samples 4, 5 and 6, when the number of particles is 100, 40, and 50 respectively.

Table 6-21: Performance Results for the WA_PSO_NFS AIS Ensemble System against HS dataset

Sample #	# of Particles	WA_PSO_NFS AIS Ensemble						
		Accuracy	Sensitivity	Specificity	Total	*Gbest*		
1	20	0.933	0.667	1.000	2.600	1.370	1.401	1.193
	30	0.933	0.667	1.000	2.600	0.243	0.647	1.170
	40	0.933	0.667	1.000	2.600	1.004	0.706	1.267
	50	0.933	0.667	1.000	2.600	0.607	1.186	1.108
	70	0.933	0.667	1.000	2.600	0.810	0.835	1.401
	100	0.929	1.000	0.917	2.845	0.045	0.431	0.743
2	20	1.000	1.000	1.000	3.000	2.769	3.113	1.439
	30	0.867	1.000	0.833	2.700	1.239	1.250	0.569
	40	0.867	1.000	0.833	2.700	1.455	1.473	0.943
	50	0.867	1.000	0.833	2.700	0.930	0.956	0.535
	70	1.000	1.000	1.000	3.000	0.043	0.186	1.607
	100	1.000	1.000	1.000	3.000	0.070	0.270	2.343
3	20	0.800	0.400	1.000	2.200	0.244	1.228	0.098
	30	0.800	0.400	1.000	2.200	0.716	1.248	0.507
	40	0.800	0.500	1.000	2.300	0.220	0.021	1.140
	50	0.800	0.400	1.000	2.200	0.029	1.119	0.790
	70	0.800	0.400	1.000	2.200	0.383	0.944	1.461
	100	1.000	1.000	1.000	3.000	0.093	0.480	1.313
4	20	0.733	0.667	0.750	2.150	0.801	0.919	1.154
	30	0.733	0.667	0.750	2.150	0.201	0.267	0.684
	40	0.846	1.000	0.818	2.664	0.010	0.471	0.777
	50	0.733	0.667	0.750	2.150	0.096	1.366	1.126
	70	0.733	0.667	0.750	2.150	0.492	1.080	1.177
	100	0.733	0.667	0.750	2.150	0.580	0.898	0.872
5	20	0.800	0.333	0.917	2.050	0.631	0.742	1.293
	30	0.800	0.333	0.917	2.050	1.123	1.344	0.722
	40	0.800	0.333	0.917	2.050	0.180	0.430	1.307

Sample #	# of Particles	WA_PSO_NFS AIS Ensemble						
		Accuracy	Sensitivity	Specificity	Total	*Gbest*		
	50	0.800	0.333	0.917	2.050	0.447	0.709	0.968
	70	0.800	0.333	0.917	2.050	0.998	1.363	1.391
	100	0.800	0.333	0.917	2.050	0.272	0.541	1.037
6	20	0.733	0.625	0.857	2.215	1.112	0.981	0.828
	30	0.769	0.714	0.833	2.317	0.201	0.186	1.238
	40	0.733	0.625	0.857	2.215	1.101	1.026	0.682
	50	0.733	0.625	0.857	2.215	0.831	0.762	0.644
	70	0.733	0.625	0.857	2.215	1.229	1.144	0.828
	100	0.733	0.625	0.857	2.215	1.088	1.018	0.590
7	20	0.857	0.333	1.000	2.190	0.080	0.611	1.453
	30	0.786	0.333	0.909	2.028	0.091	1.472	0.312
	40	0.857	0.333	1.000	2.190	0.155	0.567	1.460
	50	0.846	0.500	0.909	2.255	0.108	0.479	0.214
	70	0.786	0.333	0.909	2.028	1.067	1.294	0.664
	100	1.000	1.000	1.000	3.000	0.106	0.122	0.032
8	20	0.667	0.714	0.625	2.006	0.478	0.297	0.420
	30	0.733	0.714	0.750	2.198	0.369	0.078	1.102
	40	0.733	0.714	0.750	2.198	1.255	0.339	2.197
	50	0.733	0.714	0.750	2.198	0.443	0.149	0.736
	70	0.733	0.714	0.750	2.198	0.579	0.064	1.200
	100	0.769	0.667	0.857	2.293	0.104	0.344	1.059
9	20	0.667	0.571	0.750	1.988	1.414	1.028	1.324
	30	0.733	0.429	1.000	2.162	0.962	0.937	0.376
	40	0.667	0.857	0.500	2.024	1.192	1.387	0.511
	50	0.733	0.429	1.000	2.162	0.708	0.705	0.243
	70	0.667	0.571	0.750	1.988	1.095	0.681	1.474
	100	0.667	0.571	0.750	1.988	1.459	1.095	1.358
10	20	0.867	0.800	0.900	2.567	0.550	0.180	1.422

Sample #	# of Particles	WA_PSO_NFS AIS Ensemble						
		Accuracy	Sensitivity	Specificity	Total	*Gbest*		
	30	0.867	0.800	0.900	2.567	1.201	0.668	1.352
	40	0.867	0.800	0.900	2.567	0.956	0.662	0.792
	50	0.800	1.000	0.778	2.578	0.278	0.579	1.738
	70	0.867	0.800	0.900	2.567	0.747	0.448	1.239
	100	0.867	0.800	0.900	2.567	1.002	0.741	0.922

In Table 6.21 showing the performance results for the AIS ensemble system with adaptive learning feature against the HS dataset, the performance accuracy averaged 81.2% and ranged from 66.7% to 100%. The performance for sensitivity spanned a much wider range from 0.333 to 1.000 and averaged 0.652. The performance for specificity averaged 0.881 while varying from 0.500 to 1.000. In addition, the adaptive learning AIS ensemble system succeeded on achieving perfect results by predicting the correct class for all the test samples for samples 2 (when no. of particles is 20, 70 and 100), 3 (no. of particles is 100), and 7 (no. of particles is 100). For sample no. 5, the performance measures of accuracy, sensitivity, and specificity were constant within the sample, regardless of the number of particles, at 0.800, 0.333 and 0.917 respectively for the number of particles spanning the range 30 to 100. Samples 1 and 3 also showed similar consistency.

Table 6-22: Performance Results for the WA_PSO_NFS AIS Ensemble System against PID dataset

Sample #	# of Particles	WA_PSO_NFS AIS Ensemble						
		Accuracy	Sensitivity	Specificity	Total	*Gbest*		
1	20	0.816	0.625	0.955	2.395	1.150	1.088	1.489
	30	0.868	0.813	0.909	2.590	0.063	0.599	1.609
	40	0.816	0.625	0.955	2.395	0.853	1.154	1.180
	50	0.868	0.813	0.909	2.590	0.331	0.230	1.345
	70	0.868	0.813	0.909	2.590	0.560	0.117	1.607
	100	0.868	0.813	0.909	2.590	0.360	0.089	1.063
2	20	0.816	0.533	1.000	2.349	0.624	0.746	0.353
	30	0.816	0.533	1.000	2.349	1.473	1.070	1.166
	40	0.816	0.533	1.000	2.349	1.445	1.221	0.685

Sample #	# of Particles	WA_PSO_NFS AIS Ensemble						
		Accuracy	Sensitivity	Specificity	Total	*Gbest*		
	50	0.816	0.533	1.000	2.349	0.770	0.498	0.939
	70	0.816	0.533	1.000	2.349	0.843	0.834	1.469
	100	0.816	0.533	1.000	2.349	0.465	0.597	0.919
3	**20**	0.778	0.546	0.880	2.203	0.419	0.423	0.510
	30	0.778	0.546	0.880	2.203	0.988	0.666	1.185
	40	0.778	0.636	0.840	2.254	0.376	0.402	1.847
	50	0.778	0.546	0.880	2.203	1.146	0.923	0.692
	70	0.750	0.546	0.840	2.136	0.135	0.341	1.142
	100	0.778	0.546	0.880	2.203	1.431	1.201	0.947
4	**20**	0.757	0.539	0.875	2.170	0.913	0.976	1.056
	30	0.757	0.539	0.830	2.125	0.599	1.029	1.326
	40	0.730	0.692	0.750	2.172	0.257	0.322	1.406
	50	0.757	0.539	0.875	2.170	1.294	1.179	0.641
	70	0.757	0.539	0.875	2.170	1.328	0.979	1.176
	100	0.730	0.692	0.750	2.172	0.241	0.229	1.133
5	**20**	0.806	0.500	0.958	2.264	0.538	0.576	2.521
	30	0.806	0.667	0.875	2.347	0.286	0.117	0.978
	40	0.806	0.667	0.875	2.347	0.228	0.012	0.572
	50	0.806	0.667	0.875	2.347	0.415	0.109	1.253
	70	0.806	0.500	0.958	2.264	0.128	0.521	1.467
	100	0.806	0.667	0.875	2.347	0.477	0.077	1.337
6	**20**	0.811	0.600	0.889	2.300	1.129	0.656	1.147
	30	0.811	0.600	0.889	2.300	1.023	0.647	0.842
	40	0.811	0.600	0.889	2.300	0.400	0.229	0.397
	50	0.811	0.600	0.889	2.300	1.030	0.840	0.384
	70	0.811	0.600	0.889	2.300	1.084	0.635	1.116
	100	0.811	0.600	0.889	2.300	1.207	0.728	1.083
7	**20**	0.737	0.462	0.880	2.078	1.128	0.734	1.137

Sample #	# of Particles	WA_PSO_NFS AIS Ensemble						
		Accuracy	Sensitivity	Specificity	Total	*Gbest*		
	30	0.737	0.462	0.880	2.078	1.453	1.350	1.348
	40	0.737	0.462	0.880	2.078	0.584	0.828	1.024
	50	0.737	0.462	0.880	2.078	0.137	0.308	0.562
	70	0.737	0.462	0.880	2.078	0.261	0.447	1.307
	100	0.737	0.462	0.880	2.078	0.709	0.976	1.479
8	**20**	0.737	0.462	0.880	2.078	1.092	0.674	0.922
	30	0.737	0.462	0.880	2.078	1.479	0.994	1.156
	40	0.763	0.692	0.800	2.256	0.769	0.018	1.771
	50	0.737	0.462	0.880	2.078	1.075	0.830	0.585
	70	0.778	0.800	0.750	2.328	0.080	0.582	1.656
	100	0.711	0.769	0.680	2.160	0.032	0.156	0.438
9	**20**	0.711	0.722	0.700	2.133	0.498	0.143	1.552
	30	0.658	0.389	0.900	1.947	1.222	1.059	1.127
	40	0.684	0.556	0.800	2.040	0.320	0.116	0.493
	50	0.658	0.389	0.900	1.947	1.002	1.155	0.879
	70	0.711	0.722	0.700	2.133	0.199	0.191	0.943
	100	0.711	0.722	0.700	2.133	0.159	0.342	1.217
10	**20**	0.806	0.143	0.966	1.914	0.520	0.127	1.342
	30	0.806	0.286	0.931	2.022	0.131	0.700	1.437
	40	0.806	0.143	0.966	1.914	1.289	1.072	0.814
	50	0.800	0.200	0.950	1.950	2.036	2.693	0.081
	70	0.806	0.286	0.931	2.022	0.034	0.456	1.063
	100	0.806	0.143	0.966	1.914	1.226	1.231	0.682

Table 6.22 shows the performance results for the proposed AIS ensemble system against the PID dataset. The performance accuracy averaged 77.8%, ranging from 65.8% to 86.8%. The performance for sensitivity spanned a much lower range from 0.143 to 0.813, averaging 0.549. The performance for specificity averaged 0.883, varying from 0.680 to 1.000. This shows a decline in performance for accuracy and sensitivity in comparison with the HS dataset,

although it matches it in specificity. However, good overall performance results were obtained in most samples, taking into consideration the high dimensional aspect of this dataset. This was demonstrated clearly in the results for samples 1, 2, 5 and 8. The performance measures of accuracy, sensitivity, and specificity for samples 2, 6, and 7 were constant within the sample, regardless of the number of particles that spanned the range 30 to 100.

Table 6.23 and Figure 6.8 depict the overall performance of the adaptive learning AIS ensemble with all datasets used in this experiment. The presented results show that the proposed ensemble model has achieved an outstanding classification performance against the WBC dataset. Improved results have been obtained also with the other datasets, including the high dimensional datasets such as the BC and the PID datasets. It can be noticed from Table 6.23 that the AIS ensemble system with adaptive learning feature has succeeded in achieving optimum performance on some samples by predicting the correct class for most of the testing samples.

Table 6-23: The overall performance for the WA_PSO_NFS AIS Ensemble System

Round #	Total Performance			
	WBC	BC	HS	PID
1	2.915	2.391	2.600	2.590
2	2.899	2.269	2.700	2.349
3	2.895	2.208	2.200	2.254
4	2.888	2.336	2.150	2.172
5	2.887	2.207	2.050	2.347
6	2.887	2.192	2.215	2.300
7	2.881	2.148	2.190	2.078
8	2.871	2.128	2.198	2.256
9	2.871	2.114	2.162	2.133
10	2.860	2.109	2.567	2.022
Average	**2.885**	**2.210**	**2.303**	**2.250**

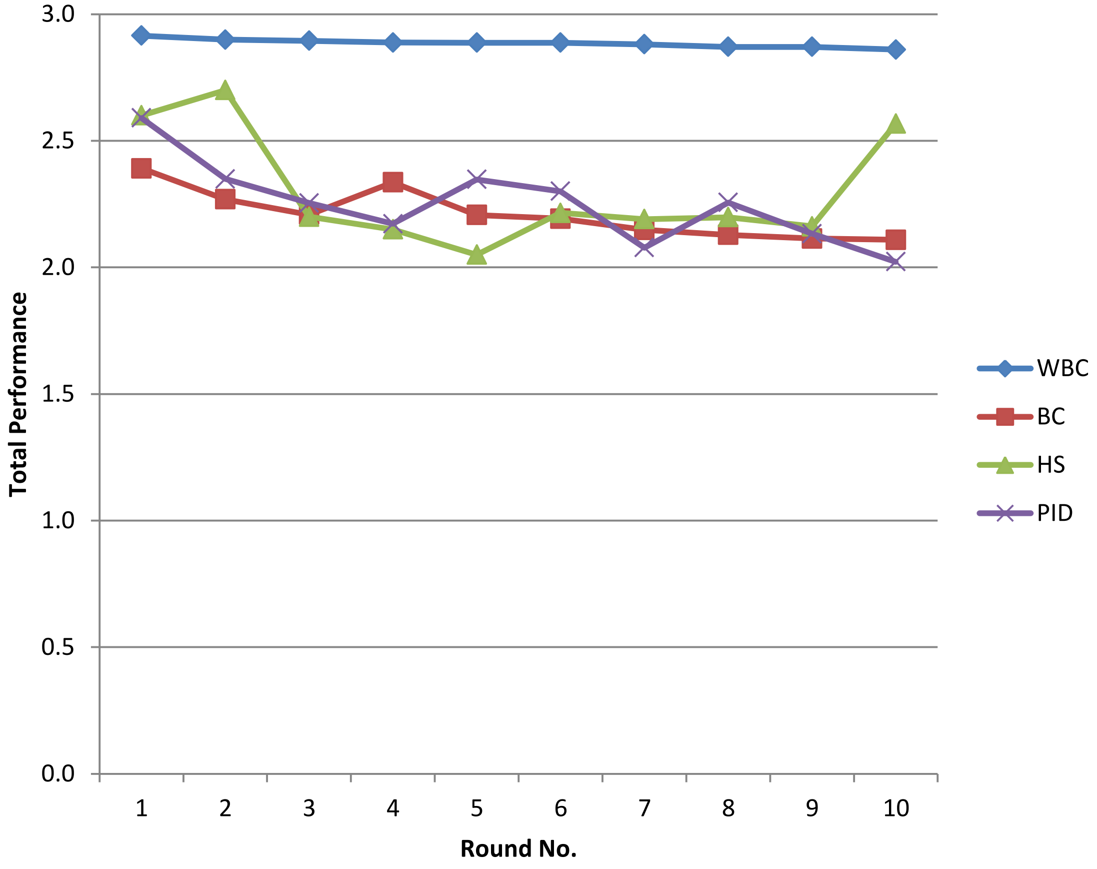


Figure 6-8: The overall performance of WA_PSO_NFS AIS Ensemble

In conclusion, this experiment demonstrates the effectiveness of the adaptive learning AIS ensemble when used for the classification problem. The proposed AIS ensemble system has achieved better results with all the datasets used compared to the other experiments.

6.4. Results comparison and discussion

So far, the experiments carried out in the previous section have shown the effectiveness of the various methods against the different datasets used separately. In this section, a comparative review between the AIS base classifiers and the several proposed ensemble methods is presented.

In addition to the three AIS base classifiers, Table 6.24 summarizes all the results achieved by the various ensemble methods presented in this work against the WBC dataset. These methods include the majority voting (MV_E), weighted average (WA_E), weighted average with PSO optimizer (WA_PSO), weighted average with neuro-fuzzy detector (WA_NFS), and the adaptive learning (WA_PSO_NFS) AIS based ensemble systems.

For this dataset, the results shown in Table 6.24 highlight that all the ensemble methods improve the total classification performance above of that of the individual base classifiers. Among all the ensemble methods, the presented results confirm that the overall classification accuracy achieved by using the majority voting AIS based ensemble, the AIS ensemble with neuro-fuzzy detector, and the adaptive learning AIS based ensemble approaches has shown the best overall performance. Figure 6.9 illustrates the overall performance of all the methods.

Table 6-24: The overall performance for the AIS algorithms and all Ensemble Systems against WBC dataset

Round #	**Total Performance (WBC dataset)**							
	ClonalG	**V-Detector**	**aiNet**	**MV_E**	**WA_E**	**WA_PSO**	**WA_NFS**	**WA_PSO_NFS**
1	2.313	2.830	1.444	2.915	2.915	2.830	2.915	2.915
2	2.719	2.798	1.729	2.899	2.899	2.899	2.899	2.899
3	2.367	2.789	1.446	2.895	2.895	2.895	2.895	2.895
4	2.145	2.552	1.343	2.888	2.888	2.664	2.888	2.888
5	2.549	2.775	1.353	2.887	2.887	2.887	2.887	2.887
6	2.436	2.775	1.240	2.887	2.887	2.887	2.887	2.887
7	2.403	2.642	1.314	2.881	2.881	2.642	2.881	2.881
8	2.477	2.871	1.286	2.871	2.871	2.871	2.871	2.871
9	2.612	2.541	1.436	2.871	2.800	2.541	2.871	2.871
10	2.441	2.860	1.257	2.860	2.860	2.860	2.860	2.860
Average	**2.446**	**2.743**	**1.385**	**2.885**	**2.878**	**2.798**	**2.885**	**2.885**

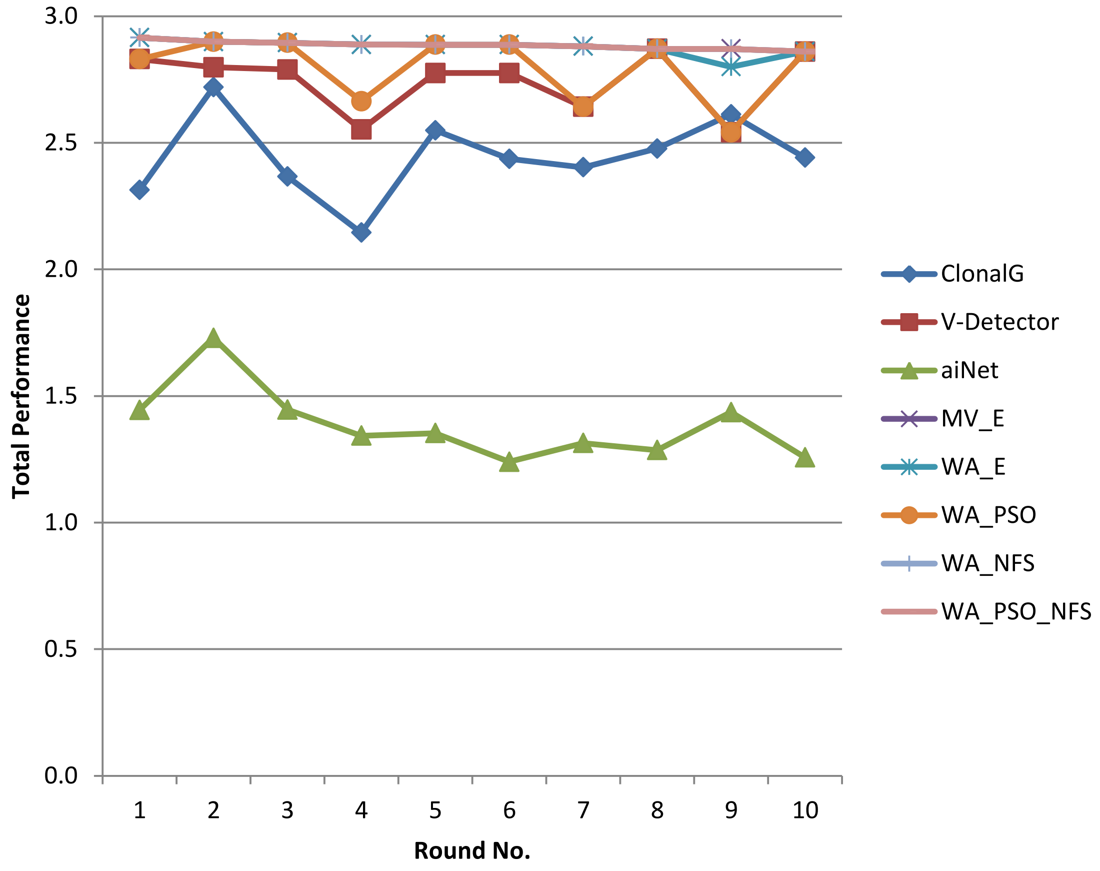


Figure 6-9: The overall performance of all methods against WBC

Similarly, Table 6.25 and Figure 6.10 clearly depict the slight improvement in classification performance obtained when using the adaptive learning AIS ensemble architecture with the BC dataset. With this high dimensional dataset, the results showed that the proposed adaptive learning ensemble systems outperformed not only the base classifiers, but all the other ensemble methods.

The experimental outcomes for testing all the methods with the HS and PID datasets are illustrated in Table 6.26, Figure 6.11 and Table 6.27, Figure 6.12 respectively. It can be noticed from the results shown that the adaptive learning AIS based ensemble has achieved the best results compared to the other methods. In Figure 6.13, the ROC plots were drawn for the various ensemble methods used in the PID testing experiment to visualize their classification performance.

Figure 6.11 represents the overall performance of all methods (AIS algorithms and all ensemble systems) against the HS dataset. The figure shows that with the best average performance of 2.303, the adaptive learning AIS based ensemble has achieved the best results compared to the other methods. The AIS ensemble with PSO optimizer method ranks second best with an average performance of 2.301. Both of these range between 2.000 and 2.850. The remaining methods score lower against this HS dataset.

In conclusion, the experiments conducted in this work have shown that the combination of multiple classifiers is an alternative strategy of improving the robustness and performance of the overall system. The experimental results have demonstrated clearly the effectiveness of the new adaptive learning AIS based ensemble architecture for the classification application especially when used with high dimensional datasets. The different particle swarm optimization and neuro-fuzzy intelligent methods used in building the proposed ensemble system have generously improved the overall classification performance of the ensemble system on every dataset. Furthermore, the slight increase in the computational requirements introduced by the proposed ensemble system can be justified by the possible gains while training with high dimensional data sets.

Table 6-25: The overall performance for the AIS algorithms and all Ensemble Systems against BC dataset

Round #	**Total Performance (BC dataset)**							
	ClonalG	**V-Detector**	**aiNet**	**MV_E**	**WA_E**	**WA_PSO**	**WA_NFS**	**WA_PSO_NFS**
1	1.963	2.391	1.286	2.391	2.391	2.391	2.391	2.391
2	1.350	2.256	1.294	2.256	2.256	2.269	2.256	2.269
3	1.840	2.208	1.371	2.208	2.208	2.208	2.208	2.208
4	1.907	2.207	1.200	2.207	2.207	2.271	2.207	2.336
5	2.121	2.207	1.200	2.207	2.207	2.207	2.207	2.207
6	1.777	2.192	1.343	2.192	2.192	2.192	2.192	2.192
7	1.919	2.148	1.235	2.148	2.148	2.148	2.148	2.148
8	1.673	2.116	1.188	2.116	2.116	2.128	2.116	2.128
9	1.934	2.114	1.412	2.114	2.114	2.114	2.114	2.114
10	1.819	2.109	1.471	2.109	2.109	2.109	2.109	2.109
Average	**1.830**	**2.195**	**1.300**	**2.195**	**2.195**	**2.204**	**2.195**	**2.210**

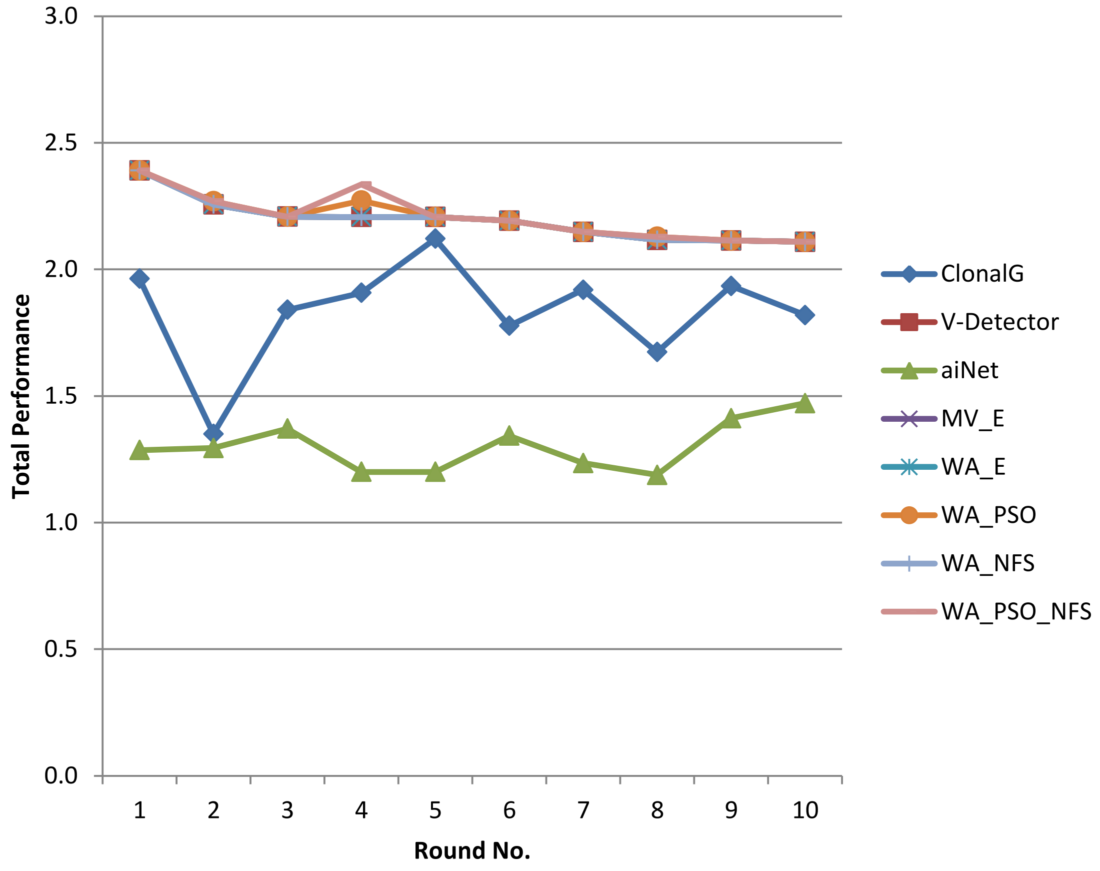


Figure 6-10: The overall performance of all methods against BC

Table 6-26: The overall performance for the AIS algorithms and all Ensemble Systems against HS dataset

Round #	**Total Performance (HS dataset)**							
	ClonalG	**V-Detector**	**aiNet**	**MV_E**	**WA_E**	**WA_PSO**	**WA_NFS**	**WA_PSO_NFS**
1	1.650	2.200	2.600	2.600	2.600	2.600	2.600	2.600
2	1.650	2.300	2.200	2.300	2.550	2.850	2.300	2.700
3	1.667	2.200	1.667	2.200	2.200	2.200	2.200	2.200
4	1.800	2.150	1.650	2.150	2.000	2.150	2.150	2.150
5	1.800	1.650	2.050	2.050	1.600	2.050	2.050	2.050
6	1.467	2.042	1.850	2.042	2.024	2.215	2.042	2.215
7	1.786	2.028	1.786	2.028	2.028	2.190	2.028	2.190
8	1.743	1.761	1.952	1.970	1.970	2.198	1.970	2.198
9	1.533	1.952	1.743	1.952	1.952	1.988	1.952	2.162
10	1.500	1.933	1.667	1.933	2.200	2.567	1.933	2.567
Average	**1.660**	**2.022**	**1.916**	**2.123**	**2.112**	**2.301**	**2.123**	**2.303**

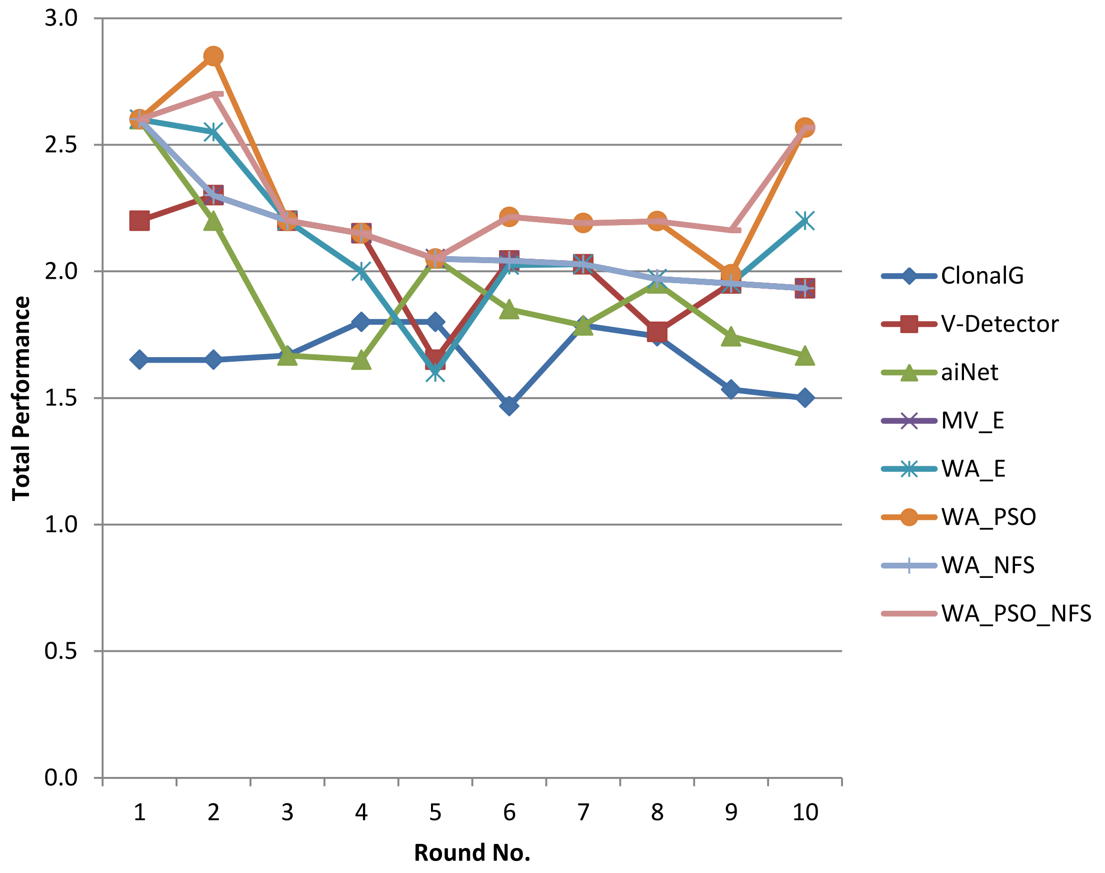


Figure 6-11: The overall performance of all methods against HS dataset

Table 6-27: The overall performance for the AIS algorithms and all Ensemble Systems against PID dataset

Round #	**Total Performance (PID dataset)**							
	ClonalG	**V-Detector**	**aiNet**	**MV_E**	**WA_E**	**WA_PSO**	**WA_NFS**	**WA_PSO_NFS**
1	2.146	2.040	1.421	2.395	2.395	2.129	2.395	2.590
2	2.023	2.163	1.395	2.349	2.349	2.256	2.349	2.349
3	2.068	1.966	1.306	2.203	2.203	2.101	2.203	2.254
4	2.102	2.031	1.351	2.170	2.170	2.135	2.170	2.172
5	1.653	2.000	1.333	2.153	2.153	2.000	2.153	2.347
6	2.046	2.045	1.270	2.108	2.108	2.300	2.108	2.300
7	2.012	1.835	1.342	2.078	2.078	1.938	2.078	2.078
8	1.909	1.909	1.342	1.983	1.983	2.078	1.983	2.256
9	1.870	1.701	1.474	1.947	1.947	1.941	1.947	2.133
10	1.803	1.836	1.806	1.914	1.695	1.836	1.914	2.022
Average	**1.963**	**1.953**	**1.404**	**2.130**	**2.108**	**2.071**	**2.130**	**2.250**

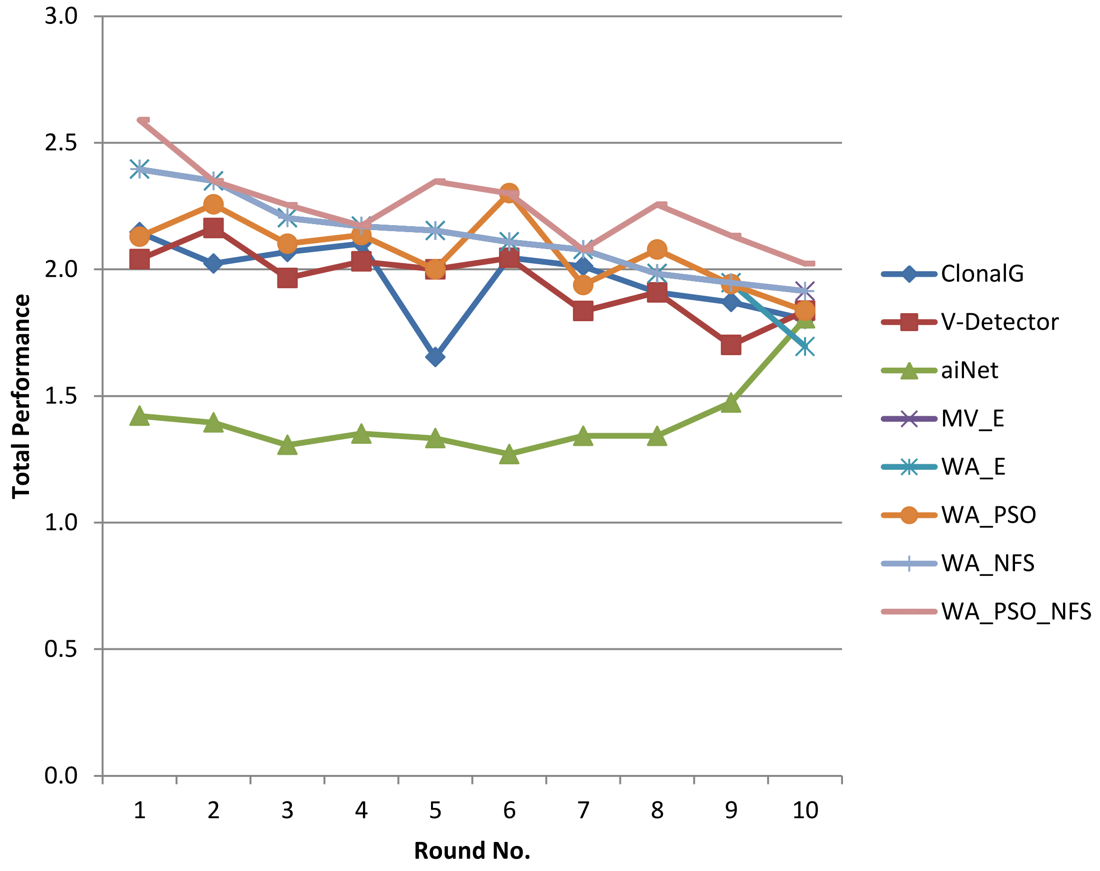


Figure 6-12: The overall performance of all methods against PID dataset

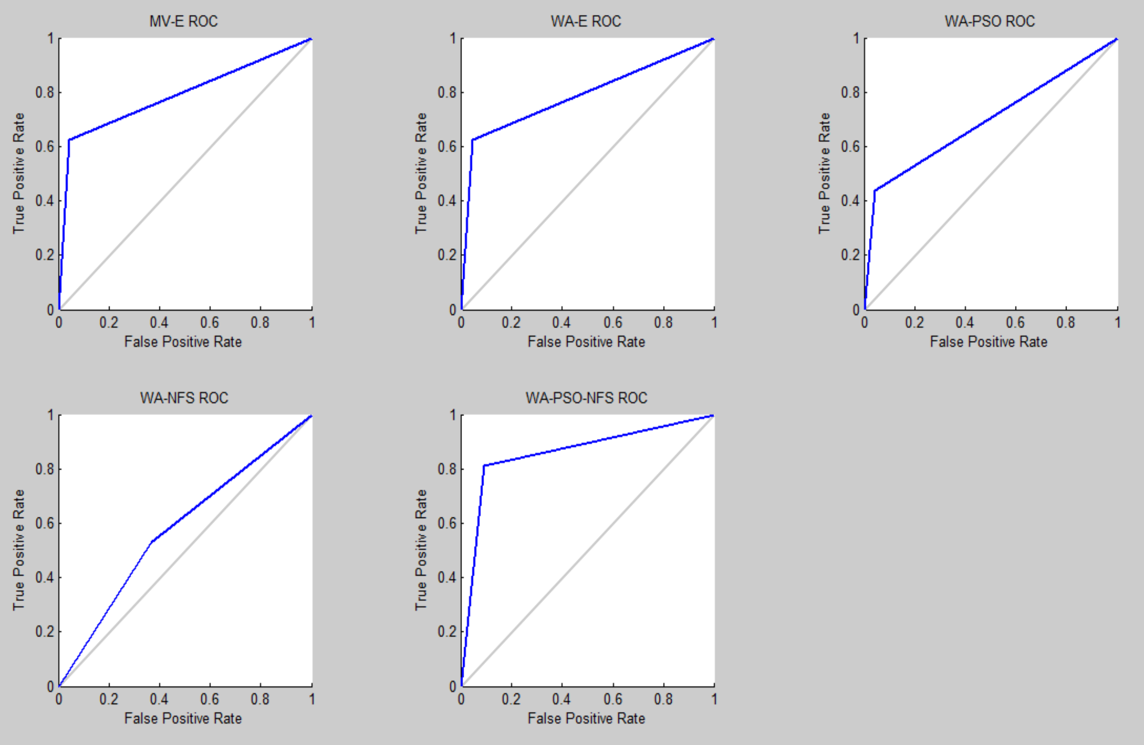


Figure 6-13: The ROC plots for all the ensemble methods for one of the samples in the PID dataset

6.5. Summary

An empirical evaluation of the various proposed methods for combining AIS classifiers has been presented in this chapter using four datasets. Several experiments have been conducted to evaluate the proposed AIS ensemble methods in comparison to the individual base classifiers and the other classifiers combining methods. The results obtained from all experiments have shown that the method of combining classifiers in an ensemble has enhanced the overall classification performance.

Furthermore, the experimental results have confirmed that the new adaptive learning AIS based ensemble system proposed in this work has achieved the best classification performance results compared to the other methods. The consistency in the performance of the proposed adaptive learning AIS ensemble model and the reliability in handling large datasets makes it the preferred choice in the designing of multiple classifier systems.

The success rates achieved with the proposed AIS ensemble models indicates that the developments of the artificial immune systems would benefit not only from the inspiration of biological immune principles and mechanisms, but also from integration with other soft computing methods, such as neuro-fuzzy systems and particle swarm optimization algorithms.

The next chapter highlights the conclusion of this dissertation and suggests avenues for further research.

CHAPTER 7: Conclusion and Future Work

7.1. Conclusion

This chapter highlights directions for future work and forms the conclusion of this research. It summarizes the main technical contributions of this thesis that were discussed in the previous chapters and directions for future research arising from this work, as presented in Section 7.2.

This research presented the essential needs of a new AIS based ensemble for data classification and the motivation behind this work. The thesis also studied several aspects related to AIS and a new ensemble based on artificial immune algorithms introduced to solve the data classification problem.

A study and survey on the AIS field (see Chapter 2), including theoretical background on the main ideas and concepts of AIS and the recent advances in the literature, has been presented in this thesis. This study has provided a motivation to continue exploring the AIS field and contribute to the development of the new AIS models and techniques. Researchers have explored the main features of the AIS mechanisms and exploited them in many application areas. Based on their aspects, some AIS techniques have been found to be more suitable for certain application areas compared to other AIS approaches. This study found that negative selection models and algorithms were widely used in fault detection and computer security applications utilizing the self/non-self recognition aspect. Alternatively, the artificial immune network approaches were used in clustering, classification, data analysis and data mining applications. The clonal selection models were used mostly for optimization problems.

In this research, a comparative study has been presented between three well known AIS models and algorithms as applied to cancer research by validation against actual cancer dataset. The three AIS algorithms inspired by the immunological principles are considered for the case study are ClonalG, V-Detector, and aiNet algorithms. The case study clearly demonstrates how AIS approaches can be employed in dealing with real-world problems in health and cancer research. The experimental results have shown that a better performance result was achieved in the experiment, especially with the V-Detector algorithm, by detecting successfully the number and the clusters for the tested dataset. This outcome leads to the conclusion that some of the AIS techniques are found to be more suitable for cancer research than other AIS approaches.

Furthermore, various methods and techniques for combining multiple classifiers have been presented. In addition, the thesis highlighted the key challenges in the development of multiple classifier systems. Different strategies and architectures have been introduced to address these challenges and the relevant work in the literature has been presented. Furthermore, a new AIS based ensemble model combining the three AIS algorithms was proposed using the majority voting and weighted average combining techniques as a first step toward achieving the objectives of this research. A new technique to measure the confidence level for the base classifiers of the ensemble system was suggested in this thesis. The method is focused on assigning the weights for the base classifier on the basis of its classification competence in order to achieve the maximum performance for the ensemble system. The proposed technique was applied successfully and the effect of using it on the performance of the AIS ensemble model is shown in the results.

Additionally, a PSO optimizer technique was utilised in this research to further enhance the performance of the proposed AIS based ensemble. The PSO optimizer is developed to find the optimal weight values for the base classifiers to further enhance the AIS ensemble overall performance. A case study was conducted to evaluate the performance of the different AIS ensemble models against a real cancer dataset. The experimental result of this study has shown that the AIS ensemble systems achieved the best classification performance results in all test runs. More specifically, the classification performance of the AIS based ensembles using the majority voting, weighted average and PSO AIS ensemble models outperformed individual AIS classifiers. However, in addition to the fact that the AIS ensemble with PSO optimizer model achieved better results than the base classifiers, the results have shown that its average performance was slightly lower than the other two classical combining techniques. Hence, further enhancement is required to improve the classification performance of the AIS ensemble system.

In this research, a new ensemble based on artificial immune algorithms for the classification problem is proposed. The proposed AIS ensemble has a unique architecture that is based on adaptive learning neuro-fuzzy detector to enhance the classification performance. The proposed ensemble has further enhanced the overall performance with the aid of the particle swarm optimization technique. These innovative solutions are combined together in an effective, computationally efficient architecture. The neuro-fuzzy detector is used to help in transforming the crisp values of the confidence levels assigned to the individual classifiers into a more accurate and satisfactory weight measures, while the PSO optimizer is utilized to find the optimal weight values, hence leading to an appropriate decision and providing desired classification accuracy. Both methods are applied successfully in the proposed architecture and the effect of using them on the classification accuracy of the ensemble system is demonstrated in the final results.

After the adaptive learning AIS ensemble system was developed, the system was tested using different samples of high-dimensional data by conducting several experiments to evaluate the performance of the proposed solution; the experimental results clearly demonstrate that the performance of the new AIS based ensemble system outperforms the conventional AIS based algorithms and the other classifiers combining methods.

An evaluation of the various proposed methods for combining AIS classifiers using four medical datasets is presented. Several experiments have been conducted to evaluate the proposed AIS ensemble methods in comparison to the individual base classifiers and the other classifiers combining methods. The results obtained from all experiments have shown that the method of combining classifiers in an ensemble has enhanced the overall classification performance. Moreover, the results have confirmed that the proposed adaptive learning AIS based ensemble system has achieved the best classification performance results compared to the other methods. The consistency in the performance of the proposed adaptive learning AIS ensemble model and the reliability in handling high-dimensional datasets make it the preferred choice in the designing of multiple classifier systems. The success rates achieved with the proposed AIS ensemble model demonstrate the added value of integrating AIS models with various artificial intelligent and optimization methods when applied to challenging application domains. Accordingly, the outcomes of the evaluation study have confirmed the effectiveness and capability of the proposed ensemble model for the classification application, especially when used with high dimensional datasets.

Although AIS models have achieved great successes in various application domains, there are still some theoretical issues that need to be further explored, such as the development of unified frameworks, convergence and scalability. The developments of the artificial immune systems would benefit not only from the inspiration of biological immune principles and mechanisms, but also hybridization with other soft computing paradigms, such as neuro-fuzzy systems and PSO algorithms. They could also be further studied and applied to more challenging application areas such as clustering of high-dimension data and to solve complex real-world problems.

The research aim formulated in the introduction of the thesis served as the main guideline and motivation. The main goal to develop an AIS based classifier with high classification accuracy was achieved based on the design and development of an adaptive learning AIS based ensemble system. Secondly, the applicability of the proposed AIS ensemble model in the classification problem of high-dimensional datasets was also successfully demonstrated. In addition, the combination of various adaptation and optimization techniques provided significant improvement to the overall classification performance of the AIS based classifier. This research provides a comprehensive foundation for such investigations. The research objectives set in this thesis can be concretely compared with the results of the dissertation, in which it evident that those goals have been successfully accomplished as highlighted in the previous conclusion paragraphs.

A number of issues remain open to future investigation, and this research can be extended to allow further improvements to the proposed adaptive learning AIS ensemble system. These issues suggest a variety of research directions that need to be pursued, some of which are briefly described in the following subsection.

7.2.Future Research

Although this thesis presented several contributions related to the classification problem and AIS based ensemble, there are still some general directions to extend the work of this thesis. The directions highlighted for future exploration are as follows:

- The success rates achieved with the adaptive learning AIS based ensemble model for some of the datasets indicate that there is further need for refinement and modifications of the approaches considered to gain greater accuracy and reliability to derive optimum benefit in cancer research.

- The focus of this study was on combining homogenous classifiers in an ensemble. Heterogeneous classifiers may help on improving the diversity of the proposed ensemble architecture and accordingly improve the overall performance of the architecture.

- The neuro-fuzzy detector and the PSO optimizer were the two main components suggested in the proposed adaptive learning ensemble architecture to tune the aggregation procedure. Future work could extend the adaptive learning feature to adjust also the base classifiers in order to achieve higher success rates, hence improving the overall ensemble classification performance.

- Testing the scalability of the proposed adaptive learning AIS ensemble architecture is an interesting subject. The proposed ensemble architecture was tested on three AIS base classifiers; the effects of the inclusion of more base classifiers can be further explored.

- The medical datasets used in testing the proposed AIS ensemble systems with adaptive learning feature have two classes. The proposed ensemble model can be further tested against other datasets with multiple classes to evaluate its performance.

- Future research work may include further investigation of the adaptive learning convergence features of the AIS based ensemble architecture and employing it to deal with more real-world engineering problems.

REFERENCES

Aggarwal C. C., Han J., Wang J., and Yu P. S., 2003. A framework for clustering evolving data streams. In Proc. of VLDB.

Aggarwal C. C., Han J., Wang J., and Yu P. S., 2004. A framework for projected clustering of high dimensional data streams. In Proc. of VLDB.

Al-Enezi J. R., Abbod M.F., and Al-Sharhan S.,2009. Advancement in Artificial Immune Systems: A perspective of Models, Algorithms and Applications. The 5th IEEE-GCC Conference, March, Kuwait.

Alonso O., Nino F., and Velez M., 2004. A Robust Immune Based Approach to the Iterated Prisoner's Dilemma. Lecture Notes in Computer Science, Issue 3239, pp. 290-301.

Alpaydin E., 1998. Technique for Combining Multiple Learners. Engineering of Intelligent System '98 Conference, Vol. 2, pp. 6-12.

Al-Sharhan S., Karray F., and Gueaieb W., 2003. Learning-based resource optimization in asynchronous transfer mode (ATM) networks. Systems, Man and Cybernetics, Part B, IEEE Transactions on, Vol. 33, Issue 1, pp. 122 – 132.

Ayara M., Timmis J., Castro L. de, and Duncan R., 2002. Negative Selection: How to Generate Detectors. 1st International Conference on Artificial Immune Systems, pp. 89-98, September.

Battiti R., and Colla A., 1994. Democracy in neural nets: Voting schemes for classification. Neural Networks, Vol. 7, No. 4, pp. 691–707.

Bentley P. and Timmis J., 2004. A Fractal Immune Network. G. Nicosia, V. Cutello, P. J. Bentley, and J. Timmis, editors, Proceedings of the Third Conference ICARIS, September, pp. 133-145, Edinburgh, UK.

Bian X. and Qiu J., 2006. Adaptive Clonal Algorithm and its Application for Optimal PMU Placement. Proceedings of 2006 International Conference on Communications, Circuits and Systems, 25-28 June, Vol. 3, pp. 2102-2106.

Breiman L., 1996. Bagging predictors. Machine Learning, Vol. 26, No. 2, pp. 123–140.

Burnet F.M., 1959. The Clonal Selection Theory of Acquired Immunity. Cambridge University Press.

Campelo F., Guimaraes F., Igarashi H. and Ramirez J., 2005. A Clonal Selection Algorithm for Optimization in Electromagnetics. IEEE Transactions on Magnetics, Vol. 41, No. 5.

Chang F.-J., Chang Y.-T., 2006. Adaptive neuro-fuzzy inference system for prediction of water level in reservoir. Advances in Water Resources, Vol. 29, pp. 1–10

Chen Z., 2007. Data Mining Based on Clonal Selection Wavelet Network. Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 30 July - 1 August, pp. 665-669.

Cho S. and Kim J., 1995. Combining multiple neural networks by fuzzy integral for robust classification. IEEE Transactions on Systems, Man, and Cybernetics, Vol. 25, No. 2, pp. 380–384.

Cutello V., Narzisi G., Nicosia G., and Pavone M., 2005. An Immunological Algorithm for Global Numerical Optimization. Artificial Evolution: 7th International Conference, Evolution Artificielle, EA 2005, 26-28 October, Lille, France, Springer, LNCS 3871, pp. 284-295.

Cutello V., Nicosia G. and Pavone M., 2006. Real Coded Clonal Selection Algorithm for Unconstrained Global Optimization using a Hybrid Inversely Proportional Hyper Mutation Operator. 21st Annual ACM Symposium on Applied Computing, SAC 2006, 23-27 April, Dijon, France. ACM Press, Vol. 2, pp. 950-954.

Dabrowski J. and Kubale M., 2008. Computer Experiments with a Parallel Clonal Selection Algorithm for the Graph Coloring Problem. IEEE International Symposium on Parallel and Distributed Processing (IPDPS), 14-18 April, Miami, FL, USA, pp. 1-6.

Danzhen G., Qian A., and Chen C., 2008. The Application of Artificial Immune Network in Load Classification. Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies, 6-9 April, Nanjing, China, pp. 1394-1398.

Dasgupta D., Ji Z., and Gonzalez F., 2003. Artificial Immune System (AIS) Research in the Last Five Years. The 2003 Congress on Evolutionary Computation (CEC '03), Vol. 1, pp. 123-130.

Dasgupta D. and Nino L., 2009. Immunological Computation: Theory and Applications. CRC Press.

Dasgupta D., 1997. Artificial Neural Networks and Artificial Immune Systems: Similarities and differences. IEEE International Conference on Systems, Man and Cybernetics, Orlando, FL, pp. 873-878.

de Castro L. and Timmis J., 2002. An Artificial Immune Network for Multimodal Function Optimization. Proceedings of IEEE Congress on Evolutionary Computation (CEC'02), Vol. 1, pp. 699-674, May, Hawaii.

de Castro L. and Timmis J., 2002. Hierarchy and Convergence of Immune Networks: Basic ideas and preliminary results. Proceedings of 1st International Conference on Artificial Immune Systems (ICARIS), University of Kent, Canterbury, UK, 9-11 September.

de Castro L. and Timmis J., 2003. Artificial Immune Systems as a Novel Soft Computing Paradigm. Soft Computing Journal, Vol. 7, Issue 7.

de Castro L. and Zuben F. von, 2001. aiNET: An artificial immune network for data analysis. H. Abbass A., Saker, R. A. and Newton C. S. (eds.). Data Mining: A Heuristic Approach, Idea Group Publishing, Chapter XII, pp. 231-259.

de Castro L. N. and von Zuben F. J., 1999. Artificial Immune Systems: Part I – Basic theory and applications. TR – DCA 01/99.

de Castro L. N. and von Zuben F. J., 2000. An Evolutionary Immune Network for Data Clustering. Proceedings of the IEEE Brazilian Symposium on Neural Networks, pp. 84–89.

de Castro L. N. and von Zuben F. J., 2001. aiNet: An artificial immune network for data analysis. H. Abbass A., Saker, R. A. and Newton C. S. (eds.). Data Mining: A Heuristic Approach. Idea Group Publishing, Chapter XII, pp. 231-259.

de Castro L. N. and von Zuben F. J., 2002. Learning and Optimization Using the Clonal Selection Principle. IEEE Transactions on Evolutionary Computation, Vol. 3, No. 6, pp. 239-251.

Delahunty A. and Callaghan D., 2004. Artificial Immune Systems for the Prediction of Corporate Failure and Classification of Corporate Bond Ratings. Master of Management Science, University College Dublin, August.

Denouex T., 1995. A K-nearest neighbor classification rule based on Dempter-Shafer theory. IEEE Transactions on Systems, Man, and Cybernetics, Vol. 25, No. 5, pp. 804–813.

Duin R. P. W., 2002. The Combining Classifier: To Train or Not to Train?. Intl. Conf. on Pattern Recognition, Vol. 2, pp. 765-770.

Eberhart R. C., and Kennedy J., 1995. A new optimizer using particle swarm theory. Proc. Sixth Intl. Symposium on Micro Machine and Human Science (Nagoya, Japan), IEEE Service Center, Piscataway, NJ, pp. 39-43.

Fern X.Z., and Brodley C.E., 2006. Cluster ensembles for high dimensional clustering: an empirical study. Technical Report, CS06-30-02, Oregon State University.

Forrest S., Perelson A. S., Allen L., and Cherukuri R., 1994. Self-Nonself Discrimination in a Computer. Proceedings of IEEE Symposium on Research in Security and Privacy, pp. 202-212, Okland, 16-18 May.

Franca F., Zuben F., and Castro L. de, 2005. An Artificial Immune Network for Multimodal Function Optimization on Dynamic Environments. Proceedings of the 2005 conference on Genetic and evolutionary computation (GECCO'05), 25-29 June, pp. 289 – 296, USA.

Frank A., and Asuncion A., 2010. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Freund Y. and Schapire R., 1996. Experiments with a new boosting algorithm. In Proceedings of the Thirteenth International Conference on Machine Learning, pp. 149–156. Morgan Kaufmann.

Fu J., Li Z., and Tan H., 2007. A Hybrid Artificial Immune Network with Swarm Learning. 2007 International Conference on Communications, Circuits and Systems (ICCCAS'07), 11-13 July, Kokura, pp. 910-914.

Gader P., Mohamed M., and Keller J., 1996. Fusion of handwritten word classifiers. Pattern Recognition Letters, Vol. 17, No. 6, pp. 577–584.

Gambino B., 2006. Reflections on Accuracy. Journal of Gambling Studies, Vol. 22, pp. 393–404.

Gan Z., Li G., Yang Z., and Jiang M., 2007. Automatic Modeling of Complex Functions with Clonal Selection-based Gene Expression Programming. Third International Conference on Natural Computation (ICNC 2007), Haikou , 24-27 August, Vol. 4, pp. 228-232.

Garrett S., 2004. Parameter-Free Adaptive Clonal Selection. Congress on Evolutionary Computation (CEC2004), 19-23 June, Vol. 1, pp. 1052- 1058.

Gong M., Jiao L., Zhang L., and Ma W., 2007. Improved Real-Valued Clonal Selection Algorithm Based On A Novel Mutation Method. Proceedings of 2007 International Symposium on Intelligent Signal Processing and Communication Systems, 28 November - 1 December, Xiamen, China.

Gong M., Zhang L., Jiao L. and Ma W., 2007. Differential Immune Clonal Selection Algorithm. Proceedings of 2007 International Symposium on Intelligent Signal Processing and Communication Systems Nov. 28 - Dec. 1, Xiamen, China.

Gonzalez L. and Cannady J., 2004. A Self-Adaptive Negative Selection Approach for Anomaly Detection. Congress on Evolutionary Computation (CEC2004), Vol. 2, 19-23 June, pp. 1561 - 1568.

Graaff A. and Engelbrecht A., 2007. A Local Network Neighborhood Artificial Immune System for Data Clustering. 2007 IEEE Congress on Evolutionary Computation, Singapore, 25-28 September, pp. 260-267.

Grabisch M., 1994. Classification by fuzzy integral: Performance and tests. Fuzzy Sets and Systems, Vol. 65, pp. 255–271.

Grabisch M., 1995. Fuzzy integral in multi-critieria deision making. Fuzzy Sets and Systems, Vol. 69, pp. 279–298.

Hajela P., Yoo J., and Lee J., 1997. GA based simulation of immune networks – applications in structural optimization. Journal of Engineering Optimization, No. 29, pp. 131–149.

Halavati R., Shouraki S., Heravi M. and Jashmi B., 2007. An Artificial Immune System with Partially Specified Antibodies. Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation (GECCO'07), 7-11 July, London, UK, pp. 57 - 62.

Hao X. and Cai-xin S., 2007. Artificial Immune Network Classification Algorithm for Fault Diagnosis of Power Transformer. IEEE Transactions on Power Delivery, Vol. 22, Issue 2, April, pp. 930-935.

Hart E. and Timmis J., 2005. Application areas of AIS: The past, the present and the future. C. Jacob, M. Pilat, P. Bentley, and J. Timmis, editors, Proceedings of the 4th International Conference on Artificial Immune Systems (ICARIS 2005), LNCS 3627, Vol. 3627 of LNCS, pp. 126–138.

Hashem S., 1997. Algorithms for optimal linear combinations of neural networks. In International Conference on Neural Networks, Vol. 1, pp. 242–247, Houston, 1997.

He Y. and Jian C., 2007. Clonal Selection Algorithm with Adaptive Mutation and Roulette Wheel Selection. International Conference on Computational Intelligence and Security Workshops, 15-19 Dec., pp. 93-96.

Heo J. S., Lee K. Y., and Garduno-Ramirez R., 2006. Multiobjective Control of Power Plants Using Particle Swarm Optimization Techniques. IEEE Transactions on Energy Conversion, Vol. 21, No. 2, June.

Heskes T., 1997. Balancing between bagging and bumping. In M. Mozer, M. Jordan, and T. Petsche, Eds., Neural Information Processing Systems, Vol. 9, pp. 466–472. MIT Press, Cambridge.

Ho T. K., Hull J. J., and Srihari S. N., 1994. Decision combination in multiple classifier systems. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 16, No. 1, pp. 66-75.

Hofmeyr S. A., 2000. An Interpretative Introduction to the Immune System. Cohen I., and Segel, L. A. (eds.). Design Principles for the Immune System and other Distributed Autonomous Systems. Oxford University Press: New York.

Hu J., Guo C., Li T. and Bu R., 2007. A Mutation-Classified, Parameter-Dynamic Immunological Algorithm for Global Optimization. Proceedings of the 2007 American Control Conference, NY, USA, July 11-13, pp. 546-551.

Huang W. and Jiao L., 2008. Artificial Immune Kernel Clustering Network for Unsupervised Image Segmentation. Progress in Natural Science, Vol. 18, No. 4, pp. 455–461.

Hunt J. and Cooke D., 1996. Learning Using an Artificial Immune System. Journal of Network and Computer Applications (1996) 19, pp. 189-212.

Igawa K. and Ohashi H., 2008. A Negative Selection Algorithm for Classification and Reduction of the Noise Effect. Appl. Soft Comput. J. (2008), doi:10.1016/j.asoc.2008.05.003.

Ishiguro A., Watanabe Y., and Uchikawa Y., 1994. Fault Diagnosis of Plant System Using Immune Networks. Proceedings of the 1994 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MF1'94) Las Vegas NV, 2-5 Oct., pp. 34 – 42.

Jacobs R., 1995. Methods of combining experts' probability assessments. Neural Computation, Vol. 7, pp. 867–888.

Jang, J.-S.R., 1993. ANFIS: Adaptive Network Based Fuzzy Inference System. IEEE Transactions on Systems, Man and Cybernetics Vol. 3, No. 23, pp. 665–683.

Jassbi J.J., Serra P.J.A., Ribeiro R.A., and Donati A., 2006. A comparison of mandani and sugeno inference systems for a space fault detection application. World Automation Congress (WAC '06), pp. 1-8, July.

Jerne N. K., 1974. Towards a Network Theory of the Immune System. Annals of Immunology, Vol. 125C, pp. 373–389.

Ji C. and Ma S., 1997. Combinations of weak classifiers. IEEE Transactions on Neural Networks, Vol. 8, No. 1, pp. 32–42.

Ji Z. and Dasgupta D., 2004a. Real-Valued Negative Selection Algorithm with Variable-Sized Detectors. LNCS 3102, Proceedings of GECCO, pp. 287–298.

Ji Z. and Dasgupta D., 2004b. Augmented Negative Selection Algorithm with Variable Coverage Detectors. Congress on Evolutionary Computation, Portland, Oregon, June.

Ji Z. and Dasgupta D., 2005. Estimating the Detector Coverage in a Negative Selection Algorithm. Proceedings of the 2005 conference on Genetic and evolutionary computation (GECCO'05), Washington, DC, USA, 25–29 June.

Ji Z., 2004. Influence of Training Data's Interpretation Strategy on Real-Valued Negative Selection Algorithm. Technical report, The University of Memphis, June.

Ji, Z. and Dasgupta D., 2009. V-detector: An efficient negative selection algorithm with "probably adequate" detector coverage. Information Sciences, Vol. 179, No. 10, pp. 1390–1406.

Jiao L. and Li Y., 2005. Quantum-Inspired Immune Clonal Optimization. The 2005 IEEE International Conference on Neural Networks and Brain, 13-15 Oct., pp. 461-466.

Jin Z., Liu G. and Wen W., 2006. Clonal Selection Algorithm with Hyper Mutation and Spatial Clone Extension. 2006 International Conference on Computational Intelligence and Security, Vol. 1, November, pp. 402 - 405.

Kang H., Kim K., and Kim J., 1997. Optimal approximation of discrete probability distribution with kth-order dependency and its application to combining multiple classifiers. Pattern Recognition Letters, Vol. 18, pp. 515–523.

Karakasis V. and Stafylopatis A., 2006. Data Mining based on Gene Expression Programming and Clonal Selection. 2006 IEEE Congress on Evolutionary Computation, Canada, 16-21 July, pp. 514-521.

Karray F. O., and de Silva C., 2004. Soft Computing Intelligent Systems Design: Theory, Tools and Application. England, UK. Addison Wesley.

Karray F., Gueaieb W., and Al-Sharhan S., 2002. The Hierarchical Expert Tuning of PID Controllers Using Tools of Soft Computing. IEEE Transactions on Systems, Man and Cybernetics, Vol. 32, pp.77-90.

Kennedy J., and Eberhart R. C., 1995. Particle swarm optimization. Proc. IEEE Intl. Con$ on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, Vol. 4, pp. 1942-1948.

Kennedy J., and Eberhart R. C., 1997. A discrete binary version of the particle swarm algorithm. Proceedings of the World Multi conference on Systemic Cybernetics and Informatics. Piscataway, NJ: IEEE Service Center, pp. 4104-4109.

Kittler J., and Roli F., 2000, Eds. Multiple Classifier Systems. First International Workshop, MCS2000, Cagliari, Italy June 2000, Proceedings, Vol. 1857 of Lecture Notes in Computer Science. Springer-Verlag Publishers, Berlin.

Kittler J., Hatef M., Duin R. P. W., and Matas J., 1998. On combining classifiers. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 3, No. 20, pp. 226-239.

Kittler J., Hojjatoleslami A., and Windeatt T., 1997a. Strategies for combining classifiers employing shared and distinct pattern representations. Pattern Recognition Letters, Vol. 18, pp. 1373– 1377.

Kittler J., Matas J., Jonsson J., and Ramos Sanchez M., 1997b. Combining evidence in personal identity verification systems. Pattern Recognition Letters, Vol. 18, pp. 845–852.

Knight T. and Timmis J., 2002. A Multi-Layered Immune Inspired Approach to Data Mining. *Proceedings of the 4th International Conference on Recent Advances in Soft Computing (*RASC’02), pp. 266-271, Nottingham, UK, December.

Kuncheva L. I., 2004. Combining Pattern Classifiers: Methods and Algorithms. John Wiley and Sons, Inc.

Kuncheva L., 1997. An application of owa operators to the aggregation of multiple classification decisions. In R. Yager and J. Kacprzyk, Eds., The Ordered Weighted Averaging Operators. Theory and Applications, pp. 330–343. Kluwer Academic Publishers, Dordrecht.

Kuncheva L., 2002. A theoritical study on six classifier fusion strategies. IEEE Transactions on Systems, Man, and Cybernetics, Vol. 24, No. 2, pp. 281–286.

Kuncheva L., Bezdek J., and Sutton M., 1998 On combining multiple classifiers by fuzzy templates. In Proceedings of the 1998 Annual Meeting of the North American Fuzzy Information Processing Society, NAFIPS’98, pp. 193–197, Pensacola, FL.

Lam L., and Suen C., 1995. Optimal combination of pattern classifiers. Pattern Recognition Letters, Vol. 16, pp. 945–954.

Le Hegarat-Mascle S., Bloch I., and Vidal-Madjar D., 1998. Introduction of neighborhood information in evidence theory and application to data fusion of radar and optical images with partial cloud cover. Pattern Recognition, Vol. 31, No. 11, pp. 1811–1823.

Li H., Hao X. and Zhang L., 2008. A Clonal Selection Algorithm Based Optimal Iterative Learning Control Algorithm. Proceedings of the 7th World Congress on Intelligent Control and Automation, 25-27 June, Chongqing, China.

Li J., Jiao L. and He W., 2005. Lamarkian Clonal Selection Algorithm for CDMA Multiuser Detection over Multi-Path Channels. International Conference on Neural Networks and Brain (ICNN&B '05), 13-15 October, Vol. 1, pp. 601- 606.

Li L., Shao-dan Z., Hong-wen L., Fen X., and Wen-bo X., 2008. Parameter Optimization of Pharmacokinetics Based on Artificial Immune Network. Applied Mathematics and Mechanics (English edition), Vol. 29(4), pp. 549–558, DOI 10.1007/s10483-008-0414-7.

Liu R., Jiao L. and Du H., 2004. Adaptive Immune Clonal Strategy Algorithm. 7th International Conference on Signal Processing (ICSP’04), 31 August - 4 September, Vol. 2, pp. 1554- 1557.

Lu H. and Zhichun M., 2008. A Clonal Chaos Adjustment Algorithm for Multi-modal Function Optimization. Proceedings of the 27th Chinese Control Conference, 16-18 July, Kunming, Yunnan, China.

Lu, Y., 1996. Knowledge Integration in a Multiple Classifier System. Applied Intelligence Vol. 6, No. 2, pp. 75–86.

Luh G. and Liu W., 2004. Reactive Immune Network Based Mobile Robot Navigation. G. Nicosia, V. Cutello, P. J. Bentley, and J. Timmis, editors, Proceeding of the Third Conference ICARIS, pp. 119-132, Springer.

LV J., 2007. Study on Chaos Immune Network Algorithm for Multimodal Function Optimization. Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007), Haikou, 24-27 August, Vol. 3, pp. 684-689.

Mamdani E.H, 1974. Application of fuzzy algorithms for control of a simple dynamic plant. Proceedings IEEE, Vol. 1, No. 12, pp. 1585–1588.

McCallum A. K., Nigam K., and Ungar L. H., 2000. Efficient Clustering of High-Dimensional Data Sets with Application to Reference Matching. In SIGKDD.

Merz C., and Pazzani M., 1997. Combining neural network regression estimates with regularized linear weights. In M. Mozer, M. Jordan, and T. Petsche, Eds., Neural Information Processing Systems, Vol. 9, pp. 564–570. MIT Press, Cambridge, Cambridge.

Mirhosseini A., Yan H., Lam K., and Pham T., 1998. Human face image recognition: An evidence aggregation approach. Computer Vision and Image Understanding, Vol. 71, No. 2, pp. 213– 230, August.

Mitra S. and Hayashi Y., 2000. Neuro–Fuzzy Rule Generation: Survey in Soft Computing Framework. IEEE Transactions on Neural Networks, Vol. 11, No. 3, MAY.

Munro P. and Parmanto B., 1997. Competition among networks improves committee performance. In Advances in Neural Information Processing Systems 9, pp. 592–598. MIT Press, Cambridge.

Nasaroui O., Gonzalez F., and Dasgupta D., 2002. The Fuzzy Artificial Immune System: Motivations, basic concepts, and application to clustering and web profiling. Proceedings of the 2002 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'02), Honolulu, HI, USA, 12-17 May, Vol. 1, pp. 711-716.

Nasraoui O., Uribe C., Coronel C., and Gonzalez F., 2003. TECHNO-STREAMS: Tracking evolving clusters in noisy data streams with a scalable immune system learning model. Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03), 19-22 Nov., pp. 235- 242.

Nauck D., Klawoon F., and Kruse R., 1997. Foundations of Neuro-Fuzzy Systems. Wiley, New York.

Neal M., 2002. An Artificial Immune System for Continuous Analysis of Time-Varying Data. Proceedings of the 1st International Conference on Artificial Immune Systems (ICARIS), Vol. 1, pp.76-85.

Neal M., 2003. Met-Stable Memory in an Artificial Immune Network. Proceedings of the 2nd International Conference on Artificial Immune Systems (ICARIS), Springer, pp. 168-180, LNCS 2787/2003.

Parkin J. and Cohen B., 2001. An Overview of the Immune System. The Lancet, Vol. 357, June 2.

Poli R., 2007. On the moments of the sampling distribution of particle swarm optimisers. In Proceedings of the workshop on particle swarm optimization: the second decade. Genetic and Evolutionary Computation Conference (GECCO), pages 2907–2914, New York, NY, USA. ACM Press.

Qiao P., Wang T. and Su J., 2008. An Improved Clone Selection Immune Algorithm. Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security, edited by Belur V. Dasarathy, Proc. of SPIE Vol. 6973, DOI: 10.1117/12.772464.

Qiao Y. and Jianping Y., 2006. AINIDS: An Immune-Based Network Intrusion Detection System. Proc. of SPIE Vol. 6241, April 18.

Rogova G., 1994. Combining the results of several neural network classifiers. Neural Networks, Vol. 7, No. 5, pp. 777–781.

Ruochen L., Haifeng D. and Licheng J., 2003. Immunity Clonal Strategies. Proceedings of Fifth International Conference on Computational Intelligence and Multimedia applications (ICCIMA'03), 27-30 September, pp. 290- 295.

Secker A., Freitas A., and Timmis J., 2003. AISEC: an Artificial Immune System for E-mail Classification. *Proceedings of the Congress on Evolutionary Computation*, pp. 131-139, Canberra. Australia, December, IEEE.

Sokolova M., Japkowicz N., and Szpakowicz S., 2006. Beyond Accuracy, F-score and ROC: A family of discriminant measures for performance evaluation. Australian Conference on Artificial Intelligence, Vol. 4304. LNCS, Germany, pp. 1015–1021.

Somayaji A., Hofmeyr S., and Forrest S., 1997. Principles of a Computer Immune System. In: Proceedings of the Second New Security Paradigms Workshop, pp. 75–82.

Tahani H. and Keller J., 1990. Information fusion in computer vision using fuzzy integral. IEEE Transactions on Systems, Man, and Cybernetics, Vol. 20, pp. 733–741.

Takagi T. and Sugeno M., 1985. Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on Systems, Man and Cybernetics, Vol. 15, pp. 116-132.

Taniguchi M. and Tresp V., 1997. Averaging regularized estimators. Neural Computation, Vol. 9, pp. 1163–1178.

Tax D., Van Breukelen M., Duin R., and Kittler J., 2000. Combining multiple classifiers by averaging or by multiplying? Pattern Recognition, Vol. 33, pp. 1475–1485.

Tian X., Yang H., and Deng F., 2006. A Novel Artificial Immune Network Algorithm. Proceedings of the Fifth International Conference on Machine Learning and Cybernetics, Dalian, 13-16 August, pp. 2159-2165.

Timmis J. and Neal M., 2001. A Resource Limited Artificial immune System for Data Analysis. Knowledge Based Systems, Vol. 3-4, No. 14, pp. 121-130, June.

Timmis J., Hone A., Stibor T. and Clark E., 2008. Theoretical Advances in Artificial Immune Systems. Theoretical Computer Science Vol. 1, No. 403, pp.11-32.

Timmis J., Knight T., de Castro L. and Hart E., 2004. An Overview of Artificial Immune Systems. Paton R., Bolouri H., Holcombe M., Parish J. H., and Tateson R. (eds.). Computation in Cells and Tissues: Perspectives and Tools for Thought. Natural Computation Series, pp. 51-86.

Timmis J., Neal M., and Hunt J., 2000. An Artificial Immune System or Data Analysis. Bios Stems, Vol. 55, pp. 143-150.

Vergas P., Castro L. de, Michelan R., and Zuben F., 2003. An Immune Learning Classifier Network for Autonomous Navigation. International Conference on Artificial Immune Systems, Edimburgo. Lecture Notes in Computer Science. Berlin Heidelberg: Springer-Verlag, Vol. 2787, pp. 69 – 80.

Vieira J., Morgado Dias F., and Mota A., 2004. Neuro-Fuzzy Systems: A Survey. Proceedings of WSEAS conference on Neural Networks and Applications (NNA 2004), Udine, Italy, March 25-27.

Wanas N., and Kamel M., 2002. Learning aggregation for combining classifier ensembles. 9th International Conference on Neural Information Processing (ICONIP’02), November 18–22, Singapore.

Waterhouse S. and Cook G., 1997. Ensemble methods for phoneme classification. In M. Mozer, M. Jordan, and T. Petsche, Eds., Neural Information Processing Systems, Vol. 9, pp. 800– 806. MIT Press, Cambridge.

Woods K., Kegelmeyer W., Philip Jr., and Bowyer K., 1997. Combination of multiple classifiers using local accuracy estimates. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 4, pp. 405–410.

Wu Y., Zhang B., Lu J., and Du K., 2011. Fuzzy Logic and Neuro-fuzzy Systems: A Systematic Introduction. International Journal of Artificial Intelligence and Expert Systems (IJAE), Vol. 2, Issue 2.

Xia F., Zhu Y., and Gao Y., 2007. Shape-Space Based Negative Selection Algorithm And Its Application On Power Transformer Fault Diagnosis. Proceedings of the 2007 IEEE International Conference on Robotics and Biomimetics, 15-18 December, Sanya, China, pp. 2149 - 2154.

Xian J., Lang F., and Tang X., 2005. A Novel Intrusion Detection Method Based on Clonal Selection Clustering Algorithm. Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21 August, Vol. 6, pp. 3905- 3910.

Xiu-Li P. and Yu-Qiang F., 2006. Solving Competitive Facilities Location Problem with the Clonal Selection Algorithm", International Conference on Management Science and Engineering (ICMSE), 5-7 October, pp. 413-417.

Xu L., Krzyzak A., and Suen C.Y., 1992. Methods of combining multiple classifiers and their applications to handwriting recognition. IEEE Trans. System, Man, and Cybernetics, Vol. 3, No. 22, pp. 418-435.

Yang J., Sun L., Lee H., Qian Y. and Liang Y., 2008. Clonal Selection Based Memetic Algorithm for Job Shop Scheduling Problems. Journal of Bionic Engineering, Vol. 5, pp. 111-119.

Yu Y. and Hou C., 2004. A Clonal Selection Algorithm By Using Learning Operator. Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, 26-29 August.

Zeng J., Li T., Liu X., Liu C., Peng L. and Sun F., 2007. A Feedback Negative Selection Algorithm to Anomaly Detection. Third International Conference on Natural Computation (ICNC 07), 24-27 August, Haikou, Vol. 3, pp. 604-608.

Zhang C. and Yi Z., 2007. An Artificial Immune Network Model Applied to Data Clustering and Classification. Proceedings of 4th International Symposium on Neural Networks, ISNN 2007, Nanjing, China, 3-7 June, Part II, LNCS 4492, pp. 526-533.

Zhang L., Zhong Y., Huang B., Gong J. and Li P., 2007. Dimensionality Reduction Based on Clonal Selection for Hyper spectral Imagery. IEEE Transactions on Geoscience and Remote Sensing, Vol. 45, Issue No. 12, December, pp. 4172 - 4186.

Zhang T., R. Ramakrishnan, and Livny M., 1996. BIRCH: an efficient data clustering method for very large databases. ACM SIGMOD Record, Vol. 2, No. 25, pp. 103-114.

Zhengbing H., Ji Z., and Ping M., 2008. A Novel Anomaly Detection Algorithm Based on Real-Valued Negative Selection System. 2008 Workshop on Knowledge Discovery and Data Mining, 23-24 January, Adelaide, SA, pp. 499-502.

Zhou Ji and Dasgupta D. Real-valued negative selection algorithm with variable-sized detectors. In LNCS 3102, Proceedings of GECCO 2004, Seattle, Washington, June 2004.

Zuo X. and Li S., 2003. The Chaos Artificial Immune Algorithm and Its Application to RBF Neuro-Fuzzy Controller Design. IEEE International Conference on Systems, Man and Cybernetics, Vol. 3, 5-8 October, pp. 2809 - 2814.