The Feature Detection Rule and its application within the Negative Selection Algorithm

by

Mario Poggiolini

Submitted in partial fulfillment of the requirements for the degree Magister Scientiae

in the Faculty of Engineering, Built Environment and Information Technology

University of Pretoria

Pretoria, South Africa

October 2008

The Feature Detection Rule and its application within the Negative Selection Algorithm

by

Mario Poggiolini

Abstract

The negative selection algorithm developed by Forrest *et al*. was inspired by the manner in which T-cell lymphocytes mature within the thymus before being released into the blood system. The resultant T-cell lymphocytes, which are then released into the blood, exhibit an interesting characteristic: they are only activated by non-self cells that invade the human body. The work presented in this thesis examines the current body of research on the negative selection theory and introduces a new affinity threshold function, called the feature-detection rule. The feature-detection rule utilises the inter-relationship between both adjacent and non-adjacent features within a particular problem domain to determine if an artificial lymphocyte is activated by a particular antigen. The performance of the feature-detection rule is contrasted with traditional affinity-matching functions currently employed within negative selection theory, most notably the r-chunks rule (which subsumes the r-contiguous bits rule) and the hamming-distance rule. The performance will be characterised by considering the detection rate, false-alarm rate, degree of generalisation and degree of overfitting. The thesis will show that the feature-detection rule is superior to the r -chunks rule and the hamming-distance rule, in that the feature-detection rule requires a much smaller number of detectors to achieve greater detection rates and less false-alarm rates. The thesis additionally refutes that the way in which permutation masks are currently applied within negative selection theory is incorrect and counterproductive, while placing the feature-detection rule within the spectrum of affinity-matching functions currently employed by artificial immune-system (AIS) researchers.

Keywords: Computational Intelligence, Artificial Immune Systems, Negative Selection Algorithm

Supervisor: Prof. A.P. Engelbrecht

Acknowledgments

- My fiancée Lauren for all of her continued love, motivation and support.
- My father and mother, sister Angela and brother Carlo for their support.
- Prof. Andries Engelbrecht for his guidance and patience.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

“Begin at the beginning,' the King said, very gravely, 'and go on till you come to the end: then stop.”

- Lewis Carrol

In the 18th century, Edward Jenner performed an experiment that would revolutionise the way in which humanity would view disease. An urban legend existed that milk maids who had developed cowpox from contact with cows’ udders were immune to smallpox. Unlike others, Jenner did not simply dismiss the urban legend, but, instead, embraced it and proved it to be scientifically true. Unbeknown to Jenner, cowpox also provided a reliable defence mechanism against other deadly diseases, such as poliomyelitis, measles and neonatal tetanus [24]. The scientific marvel witnessed by the world on 8 May 1980, when, during the 33rd Assembly of the World Health Organization, smallpox was declared to have been eradicated, was due to the astonishing way in which the natural immune system (NIS) functions. The NIS is able to generalise and develop effective defence mechanisms against a harmless agent that is structurally similar to a deadly pathogen (a concept often exploited by vaccines). Furthermore, the NIS continually learns and fine-tunes its response to a particular pathogen throughout the lifetime of an individual.

The resilience and learning capability of the NIS has inspired many researchers to study how the NIS works, in an attempt to emulate certain facets of the NIS within computer systems with a view into solving very complex problems faced on a daily basis. Hence the computational intelligence field, called artificial immune systems (AISs), was born. There are many different models within AIS theory (see Figure 4), each model having advantages and disadvantages associated therewith.

The work presented in this thesis focuses on the negative selection algorithm (NSA), a particular domain of AIS, inspired by the negative selection process occurring within the NIS (refer to section 2.4.2) and improves on the NSA by introducing an affinity-matching function (refer to Figure 7 for a description of affinity-matching functions), called the feature-detection rule. The main premise of the feature-detection rule is to pre-process an antigen by extracting k features from the antigen

before presenting the antigen to an artificial lymphocyte (ALC)/detector. The thesis presents a detailed overview of negative selection theory, in addition to the most popular AIS algorithms to date, by means of an extensive background study. The purpose of the background study is to empower the reader with enough knowledge to critically evaluate NSAs and to show how and why the feature detection rule is an improvement on the set of traditional affinity-matching functions used within NSA implementations.

1.1 Motivation

The original NSA, developed by Forrest *et al*. [33], is a conceptually simple algorithm and has been adopted widely by the general AIS community. A major advantage of the NSA is that it is conceptually simple and allows a variety of different affinity-matching functions to be employed. The most popular affinity-matching functions currently used by AIS researchers (within the context of the NSA) have limited foresight in that they merely consider relationships between adjacent attributes of an antigen vector and an artificial lymphocyte (ALC)/detector vector to determine whether the ALC is activated by a particular antigen. Furthermore, these affinity-matching functions are also renowned for inducing undetectable strings or holes within a particular problem domain owing to the mechanics drawn upon by these affinity-matching functions.

The work presented in this thesis:

- Discusses a new affinity-matching function, which learns the relationship between both adjacent and non-adjacent attributes of an antigen vector and an ALC/detector to determine whether the ALC/detector is activated by the antigen.

- Shows how the new affinity-matching function does not induce holes and exhibits superior detection rates and false-alarm rates.
- Examines the schism between views on permutation masks currently existing within AIS literature (some researchers state that permutation masks are a vital mechanism to reduce holes induced by detectors, whereas other researchers disagree).

1.2 Objectives

The primary objectives of this thesis can be summarised as follows:

- to provide a detailed overview of AIS research, the key focus being on the NSA of Forrest *et al.*;
- to introduce a new affinity-matching function, which is shown to improve the detection rate offered by the NSA, for data sets exhibiting a relationship between both adjacent and non-adjacent attributes;
- to compare the new affinity-matching function, the feature-detection rule, to a number of other popular affinity-matching functions;
- to suggest an alternative way to implement permutation masks within NSAs, thereby reinstating the importance of permutation masks;
- to develop a mechanism to estimate the overfitting and generalisation capabilities of the NSA under different affinity-matching functions (note that the term affinity-matching function and detection rule are used interchangeably in this thesis); and
- to place the feature-detection rule within a framework comprising of the *r*-contiguous bits rule, the *r*-chunks rule and the feature-detection rule.

1.3 Methodology

The thesis presents an extensive literature study of negative selection theory in order to characterise the methodologies used by AIS researchers to:

- create a set of metrics that can be used to measure the relative success or failure of the application of a particular NSA within a problem domain;
- understand the effect of different affinity-matching functions used within NSAs in order to reason about their effectiveness; and
- mathematically and empirically compare the new affinity-matching function to a number of different affinity-matching functions.

The thesis discusses overfitting and generalisation within artificial immune systems in general and produces two algorithms that can be used to measure the overfitting and generalisation exhibited by a particular instance of an AIS algorithm on a particular data set (refer to sections 3.5.2 and 3.6.3 for a discussion of overfitting and generalisation).

The thesis's new affinity-matching function, the feature-detection rule, is discussed and analysed, both mathematically and empirically, on several data sets. The performance of the feature-detection rule is compared to the *r*-chunks rule and the hamming-distance rule (refer to section 3.4.2 for a general discussion of matching rules).

1.4 Contribution

The work presented in this thesis has the following contributions:

- A new affinity-matching function, the feature-detection rule, which uses the relationship between both adjacent and non-adjacent attributes of an antigen vector in order to determine whether an ALC/detector is activated by the antigen.
- A more effective way to apply permutation masks to NSAs in order to realise the full benefits originally intended by applying permutation masks.
- A means to estimate the current generalisation and overfitting exhibited by a particular detection rule.

1.5 Thesis Outline

The remainder of the thesis is organised as follows:

Chapter 2 presents an overview of the natural immune system. It is vital to have a good understanding of the natural immune system in order to reason about and appreciate artificial immune systems.

Chapter 3 discusses the fundamentals of artificial immune systems (AISs) by providing a taxonomy of different AIS classes, in addition to explaining the key concepts on which all AIS algorithm classes are based.

Chapter 4 provides a detailed overview of negative selection theory, the most popular negative selection algorithms to date and the methodologies often used to reason about negative selection algorithms.

Chapter 5 discusses the foundations of the feature-detection rule from both a conceptual and mathematical perspective.

Chapter 6 empirically compares the feature-detection rule to the hamming-distance rule and the *r*-chunks rule (with and without a permutation mask) under a number of different data sets, with the complexity of the problem domain differing across each data set.

Chapter 7 summarises the work presented by the thesis and outlines additional future directions that can be undertaken to extend the feature-detection rule introduced in this thesis.

Chapter 2
The Natural Immune System

"If you can't explain it simply, you don't understand it well enough."

- Albert Einstein

The natural immune system (NIS) has evolved over millions of years and comprises various facets, which act in tandem to protect the body. One of the most remarkable aspects of the NIS is that it has both a genetic (germ-line) and an adaptive (somatic) component. The innate immune system is genetically based: that is, it does not require a previous encounter with an antigen in order to be able to recognise it, and it does not develop a memory. The adaptive immune system, however, is capable of adapting and fine-tuning its response to an encountered antigen and does develop a memory [8, 1, 62]. Immunologists generally subscribe to different schools of thought about both the functional and the organisational behaviour of lymphocytes within the immune system. Immunologists have produced several theories governing the behaviour of immune systems, the most notable being the classical view, the danger-theory view and the network-theory view [26]:

- The classical view of the NIS, defined by Burnet [9], postulates that the main function of the NIS is to successfully discriminate between self (cells occurring naturally within the body) and non-self (foreign cells/antigens). It should be noted that immunologists do not fully comprehend how this self/non-self discrimination is accomplished; they believe that the body learns to distinguish the difference between self and non-self very early in life [3].

- The immune network theory, introduced by Jerne [51, 52] and explored further by Perelson [71], postulates that the immune system is an intricate network of cells that recognise one another even in the absence of antigens. Immune network theory hypothesises a viewpoint on actions performed by lymphocytes, pre-immune repertoire selection, tolerance and self/non-self discrimination and immunological memory.

- The danger-theory, introduced by Matzinger [3, 60, 61], concludes that the immune system actually discriminates “some self from some non-self”. For example, why does the immune system not react to ingested food residing in the gut? Although each view differs on how the NIS successfully responds to antigen, each view is firmly grounded on concepts established within the classical view of the immune system.

This chapter provides a brief overview of the constituents comprising the NIS, from the classical viewpoint of the NIS, and the mechanisms that it employs to eliminate unwanted pathogens from the human body. The key objective of this chapter is to provide the reader with a broad overview of what is currently known about the NIS by following a bottom-up approach starting with fluid systems in section 2.1.

2.1 Fluid Systems

Constituents of the immune system are housed primarily in the body’s two primary entwined fluid systems, namely, the blood system and the lymph system [8, 1]. This section provides a brief overview of the blood and lymph system.

2.1.1 Blood System

An average human being of 70 kg has roughly 5 litres of blood, which is 7 percent of the body’s total weight. Blood is composed of 52 to 62 percent liquid plasma, which is 91.5 percent water and 38 to 48 percent cells. Blood is manufactured, by means of a process called haematopoiesis, by stem cells situated predominantly in the bone marrow. These stem cells produce haemocytoblasts, which differentiate into three types of blood cell: erythrocytes (red blood cells), leukocytes (white blood cells) and thrombocytes (platelets) [8, 1].

- Leukocytes play a predominant role within the NIS and are categorised as being either granulocytes or agranulocytes (containing no granules).
- Granulocytes are composed of neutrophils (55 to 70 percent), eosinophils (1 to 3 percent), and basophils (0.5 to 1.0 percent).
- Eosinophils are weak phagocytes, believed to remove toxic substances from the blood [1].
- Basophils are very similar to large mast cells and are possibly transported to tissues where they become mast cells and release heparin, a substance that prevents blood coagulation [1].
- Neutrophils are short-lived cells responsible for the bulk of the immune response by ingesting antigens. Neutrophils, being short lived, have a half-life of four to ten hours when not activated and are subject to immediate death upon ingesting an antigen.

Agranulocytes are lymphocytes – comprising T-cells and B-cells – and monocytes [8, 1]:

- Lymphocytes circulate throughout the blood and lymph systems and reside in the lymphoid glands [8, 1]. Lymphocytes and their functions are discussed further in section 2.3.
- Monocytes develop from myelomonocytic stem cells in the bone marrow. They are released into the blood, where they circulate for several days before migrating into tissues, where they mature into macrophages.

The blood system is illustrated in Figure 1.

2.1.2 Lymph System

Lymph is a clear, transparent and colourless fluid, containing no red blood cells which envelopes and protects organs. Lymph flows through lymphatic vessels from the interstitial fluid up to either the thoracic duct or the right lymph duct. Ducts terminate in the subclavian veins, where the lymph is mixed into blood. The right lymph duct drains the right side of the thorax, neck, and head, whereas the thoracic duct drains the rest of the body. Lymph transports lipids and lipid-soluble vitamins, which are absorbed via the gastrointestinal tract.

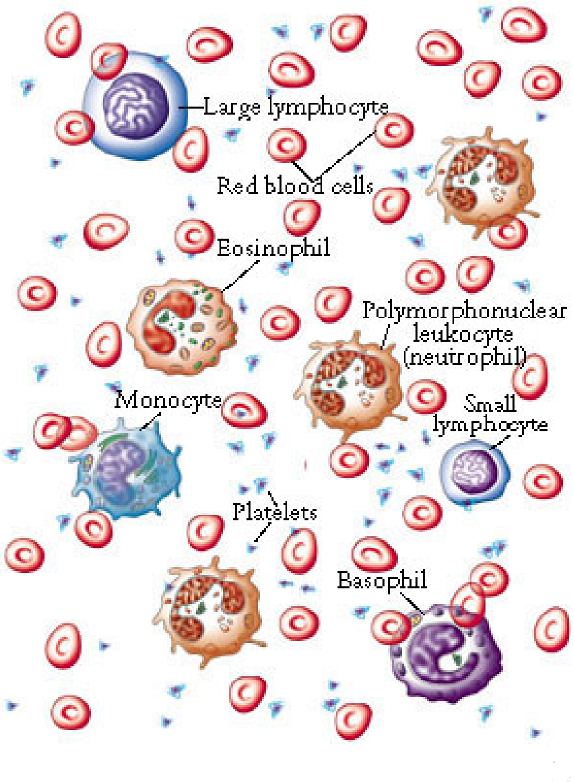


Figure 1. *Blood system (this image was taken from [8])*

Since there is no active pump in the lymph system, backpressure is not produced. Lymphatic vessels have one-way valves, that prevent backflow and are similar to veins. These valves contain additional small bean-shaped lymph nodes, which filter the lymphatic fluid. Antigens are usually presented to the immune system at the lymph nodes [8, 1].

The human lymphoid system comprises primary and secondary organs*:*

- Primary organs include bone marrow (in the hollow centre of the bones) and the thymus gland (located behind the breastbone above the heart).
- Secondary organs are located near possible pathogen portals, for example, adenoids, tonsils, spleen (located at the upper left of the abdomen), lymph nodes (along the lymphatic vessels, with large concentrations in the neck,

armpits and abdomen), Peyer's patches (within the intestines) and the appendix [8].

The structure of the lymph system is illustrated in its entirety in Figure 2.

Immunity is achieved as a result of the mechanisms employed by the immune system to protect the body against re-infection from a previously encountered antigen.

The NIS provides two different levels of immunity, namely, innate immunity and adaptive immunity. Adaptive and innate immunity are described in more detail in the following subsections.

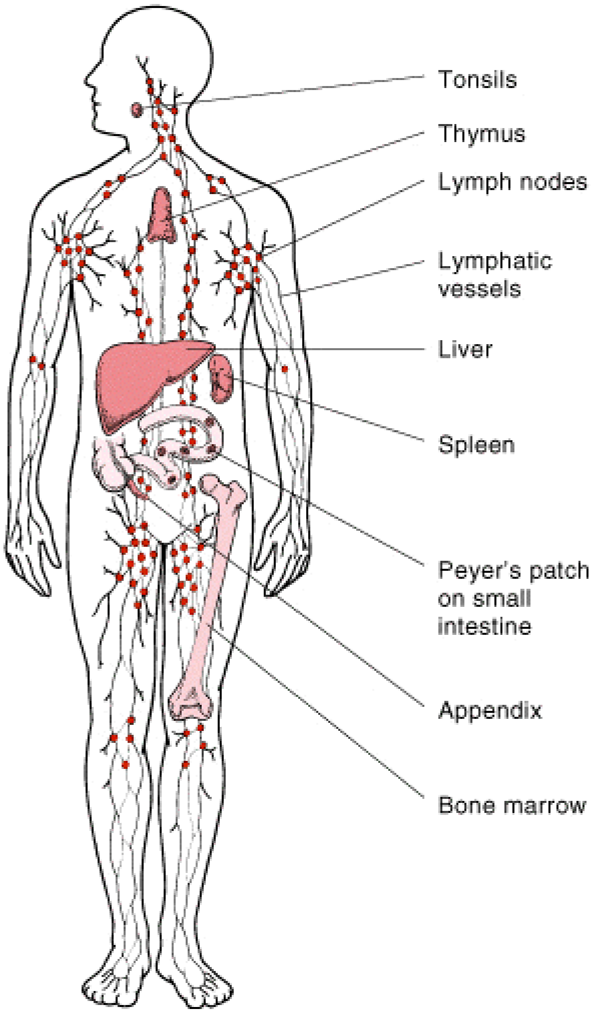


Figure 2. *Overview of the human lymphoid system (this image was taken from [62])*

2.2 Innate Immunity

The innate immune system is genetically based and, as such, is referred to as non-specific immunity. It requires no previous encounter with an antigen in order to be able to recognise it and does not develop a memory [62]. It is non-specific, and all antigens are attacked with an equal probability [8].

Innate immunity incorporates physical barriers such as skin (also referred to as mucosal immunity), cellular components, that is, the phagocytic system, natural killer cells and soluble components comprising proteins, neutrophils, dendritic cells, cytokines and symbiotic microbes, such as normal flora [62]. The innate immune system uses receptors encoded in an individual’s germ-line to identify an antigen i.e. the innate immune system is different from one individual to another. The receptors can be found on cells such as macrophages (a phagocyte) , neutrophils, natural killer cells in addition to roaming freely within the lymphatic system.

The components comprising the innate immune system are discussed in the subsections below.

2.2.1 Mucosal Immunity

The most important barrier to pathogens is our skin. Consequently, the skin is the human body’s largest organ. The skin is literally impenetrable to most micro-organisms (if it is not torn). The human body has evolved a number of additional mechanisms to expel a pathogen [8]:

- Pathogens are ejected from the body through ciliary action (coughing and sneezing) and through the flushing action of tears and saliva.
- Sticky mucus in the respiratory and gastrointestinal tracts ensnares many micro-organisms.
- The pH of skin secretions inhibits bacterial growth, because it is less than 7.0. Hair follicles secrete sebum, which contains lactic acid and fatty acids, which, in their turn, prevent certain fungi and bacteria from growing.

- Saliva, tears, nasal secretions and perspiration contain lysozyme, an enzyme that destroys gram-positive bacterial cell walls by inducing cell lysis (bursting).
- Many pathogens are also destroyed in the stomach owing to its mucosa secreting hydrochloric acid and protein-digesting enzymes.

2.2.2 Normal Flora

Normal flora are microbes residing in our body and usually do not cause harm. The human body comprises about 10^{13} cells and 10^{14} bacteria, mostly in the large intestine. There are about 10^3 to 10^4 microbes per cm^2 on the skin (predominantly staphylococcus, aureus, staphylococcus epidermidis, diphtheroids, streptococci, etc). Lactobacilli reside in the stomach and small intestine. Normal flora occupy all of the available ecological niches within the body and produce bacteriocins, defensins, cationic proteins and lactoferrin, which destroy other bacteria competing for the same niche.

Bacteria, however, can become problematic when they overrun ecological niches not destined for them, for example, if staphylococcus gains entry into the body through a cut in the skin [8].

2.2.3 Phagocytes

Phagocytes include neutrophils, monocytes (occurring in the blood) and macrophages (occurring in the tissue) [62]. A phagocyte attracts, by chemotaxis, and ingests foreign bodies through a process known as phagocytosis [1].

Promonocytes are manufactured in the bone marrow. When they are released into the blood system, they are termed circulating monocytes, which mature into macrophages [8].

2.2.4 Macrophages

Macrophages are strategically situated at interfaces of tissues with blood or cavity spaces [62]. For example, macrophages can be found in the lungs, liver, lining of the lymph nodes and spleen, brain microglia, kidney mesangial cells, synovial cells and osteoclasts [8]. Macropahges are long lived and attack not only diseased cells, but also pathogens living within cells. Once a macrophage has engulfed a cell, the macrophage processes the cell internally and distributes some of the cell's proteins, named epitopes, on its surface in conjunction with some of its own proteins. Other immune cells can then infer the structure of the invading pathogen from the macrophage. Owing to the pivotal role that they perform, macrophages are often termed antigen-presenting cells (APCs).

Macrophages are subdivided into two groups:

- Non-fixed/wandering macrophages roam blood vessels and travel to infection sites to eliminate dead tissue and pathogens. The process by which a macrophage squeezes through a capillary wall to tissue is known as diapedesis/extravasation. Macrophages are attracted to an infection site by the presence of histamines [8, 1].
- Tissue macrophages are monocytes that wander into tissues, become fixed in the tissues and swell to form tissue macrophages. They frequently proliferate and form capsules around foreign particles that cannot be digested, thus preventing the spread of disease [1].

2.2.5 Natural Killer Cells

Natural killer cells travel in the blood and lymph, causing cancer cells and virus-infected cells to lyse (burst) [8].

2.2.6 Neutrophils

Polymorphonuclear neutrophils are phagocytes with no mitochondria and store glycogen for energy. They do not divide, only live between one to four days and comprise 50 to 75 percent of all leukocytes. Neutrophils are a key form of defence against pyogenic (pus-forming) bacteria and are always the first to emerge after the onset of an infection. Macrophages appear several hours later.

2.2.7 Dendritic Cells

Dendritic cells are covered with an entanglement of membranous processes. There are four types of dendritic cells and most of them are highly efficient antigen-presenting cells: Langerhans cells, interstitial dendritic cells, interdigitating dendritic cells and circulating dendritic cells. These cells do their utmost to attract antigens and present them to T-helper cells [8].

Cells within the innate immune system bind to antigens with the aid of highly specialised pattern-recognition receptors. These receptors are genetically encoded and have evolved so that broad clusters of antigens can be distinguished.

Although the importance of innate immunity regarding its role in our survival cannot be overstated, adaptive immunity has generated an immense amount of research interest because of its learning capability.

2.3 Adaptive/Acquired Immunity

Adaptive or acquired immunity differs between people and is dependent on the subset of antigens with which an individual has come into contact. Adaptive immunity is comprised of three key components namely learning, memory, and adaptability [62]. The key component of the adaptive immune system is the lymphocyte.

Lymphocytes can be differentiated into two major types, namely, B-cells and T-cells. Approximately 20 to 50 percent of circulating lymphocytes can be found in peripheral blood, whereas the rest is constrained within the lymph system. Approximately 80 percent of lymphocytes are T-cells, 15 percent are B-cells, and the remainder are undifferentiated cells. The total mass of all lymphocytes is approximately the same as the human brain [8, 1,62].

Although both B-cells and T-cells are produced by stem cells within the bone marrow, there are a number of factors which differentiate them most notably:

- Upon activation, B-cells undergo a clonal selection process.
- B-cells produce antibodies whereas T-cells do not [8].

- T-cells undergo a sensitisation process in the thymus before being released into the blood/lymph system; hence they are called “thymus”/T-cells [8, 1].

T-cells are primarily responsible for cell-mediated immunity, whereas B-cells are primarily responsible for humoral immunity (immunity of the human immune system) [8]. Cell-mediated immunity is discussed in section 2.4, and humoral immunity is discussed in section 2.5.

2.4 Cell-mediated Immunity (Cellular Immunity)

Cellular immunity is achieved through the interaction of various classes of T-cell. T-cells do not have surface immunoglobulin (see section 2.5.1) like B-cells and recognise antigens primarily with their specialised “antibody-like” receptors and other adhesion molecules [62]. The major histocompatibility complex (MHC) is the means through which T-cells can react to antigens and is briefly discussed in section 2.4.1, before the different T-cell classes are discussed in section 2.4.2.

2.4.1 Major Histocompatibility Complex

Unlike B-cells, which can respond to soluble/free-floating antigen, T-cells can rarely do so and can only generally respond to antigens embedded in the MHC. MHC products can be differentiated into two classes:

- Class 1 products have a wide distribution and are present on the surface of all nucleated cells.
- Class 2 products have a more limited distribution on B-cells, macrophages, dendritic cells, Langerhans cells and activated T-cells [62].

An antigen is processed and associated with MHC before encountering T-cells via antigen-presenting cells; for example, macrophages (refer to section 2.2.4 for a general discussion of macrophages). The process is not fully understood, but immunologists have ascertained that, in order to be processed, an antigen must be unfolded, degraded and fragmented.

Antigens subject to exogenous processing undergo endocytosis and degradation in lysosomes and are associated with class 2 MHC products. Conversely, through endogenous processing, an antigen is processed intracellularly (for example, a viral infection), and the resulting peptides are transported to the endoplasmic reticulum by transporter proteins. Once in the endoplasmic reticulum, these peptides are associated with class 2 MHC products and are transported to the cell surface [62].

2.4.2 T-cells

T-cells (thymus cells) migrate to the thymus to mature after they have been created to learn the concept of “self” by undergoing two selection processes: positive selection and negative selection [10]. There are contrasting views on positive and negative selection within immunology; these views are founded on either the avidity hypothesis or the differential signalling hypothesis:

- The avidity hypothesis postulates that the avidity with which a T-cell lymphocyte binds to a self-MHC peptide complex dictates whether the T-cell is positively or negatively selected. The positive selection process destroys all T-cells with a relatively weak avidity to a self-MHC peptide complex. The negative selection process destroys all T-cells with a strong avidity to a self-MHC peptide complex. T-cells thus only have a single paratope, involved in both positive and negative selection [8, 1, 62].
- The differential signalling hypothesis formulated by Cohn [10] postulates that there are two different paratopes on T-cell receptors, namely, anti-r and anti-p. Anti-r paratopes are germline encoded and provide specific recognition of the MHC-peptide complex. Anti-p paratopes are somatically encoded and provide specific recognition of the peptide bound to a MHC molecule. Positive selection and negative selection are the result of qualitatively different interactions with a T-cell’s receptor.

The avidity hypothesis is the most popular hypothesis and is used throughout this thesis. T-cells that have survived both positive and negative selection are released from the thymus [7, 58]. Interestingly, if a foetus’s thymus is removed several months before birth, cellular immunity cannot develop, and the likelihood of organ rejection after a transplant is reduced [1].

There are several different varieties of T-cell: T-helper cells and T-suppressor cells/cytotoxic cells [8, 62]. Each type of T-cell is discussed in further detail below.

2.4.2.1 T-helper Cells

T-helper (TH) cells manage the immune response by secreting lymphokines, chemicals that cause both T-cells and B-cells to grow and divide, attract neutrophils and enhance the abilities of macrophages [8].

2.4.2.2 T-suppressor Cells/Cytotoxic Cells

T-suppressor (TS) cells inhibit the production of killer T-cells when the killer T-cells are no longer needed [8].

2.4.2.3 Killer Cells

The primary task of killer T-cells is to release lymphotoxins, which release results in cell lysis [8]. There are several categories of killer cell depending on MHC restriction, sensitisation requirements, target specificities and responses to cytokines (refer to section 2.4.3). These categories can broadly be simplified into MHC restricted and MHC non-restricted. Each cell delivers a lytic signal through the target-cell membrane [62].

2.4.2.3.1 MHC-restricted Killer Cells

Cytotoxic T-cells are killer cells generated through sensitisation either against cells that express foreign MHC products or against cells modified by viral infections. The life cycle of a cytotoxic T-cell can be in one of three states: a cytotoxic cell upon stimulation, an effector cell, which has differentiated and been specialised, or a memory cell, which can become an effector when it is restimulated [62].

2.4.2.3.2 MHC-non-restricted Killer Cells

MHC-non-restricted killer cells do not require sensitisation like MHC-restricted killers and are often termed natural killers. Natural killer cells comprise 5 to 30 percent of the peripheral blood lymphocytes. These killer cells do not belong to the T- and B-cell classes of lymphocyte and thus do not express either immunoglobulin or T-cell receptors on their surface. Natural killer cells will terminate certain autologous, allergenic and xenogeneic tumour cells even in the absence of class 1 MHC. Natural killer cells have several class 1 MHC receptors on their surface, and these are referred to as killer-cell inhibitory receptors. Interestingly, killer-cell inhibitory receptors have also been observed on T-cells. Immunologists are still uncertain how T-cells with different receptors for the same molecule class (class 1 MHC) can decide whether a T-cell is activated or inhibited [62].

2.4.3 Cytokines

Both T-cells and monocytes secrete cytokines, which influence both close and distant events [62]. Cytokines are also aptly termed the messengers of the immune system and are proteins secreted by cells that act to coordinate an immune response. Cytokines comprise a diverse assortment of interleukins, interferons and growth factors [1].

2.5 Humoral Immunity

Humoral immunity is achieved through the interaction of B-cells and their related antibodies. B-cells comprise 5 to 15 percent of blood lymphocytes and occur in the outer subcapsular cortical area in primary, secondary follicles and medullary cords [62].

Immature B-cells develop in the bone marrow through a process of significant gene rearrangement. Antigens do not play a role in B-cell development and, in fact, their interaction with B-cells can lead to clonal inactivation or tolerance. Immature B-cells leave the bone marrow and enter peripheral lymphoid organs.

B-cells reach maturity when two conditions are met: an antigen binds to its receptors, and a cytokine has been released by a T-helper cell in the vicinity of the immature B-cell. Immature B-cells within the vicinity of the antigen endure differentiation and clonal proliferation/selection and reproduce asexually via mitosis [8, 62].

The result of clonal selection culminates in a number of plasma cells, tailored to match a specific antigen-secreting antibody at a rapid rate, and newly formed memory B-cells [8, 1, 62].

The immune response that is produced when an antigen is first encountered in the human body is aptly termed the primary immune response. The primary immune response is characterised by the presence of an antigen, an initially latent period during which few antibodies are secreted – predominantly IgM – and, then, with the help of T-helper cells, the secretion of a large amount of IgG, IgA or IgE (antibody classes are discussed in more detail in section 2.5.1), resulting in the creation of many memory cells [8, 62]. The subset of B-cells with the highest affinity to the antigen become long-lived or memory B-cells.

The secondary immune response occurs upon subsequent encounters with the same antigen. The primary characteristics of this phase are rapid proliferation of B-cells, rapid differentiation into mature plasma cells and large numbers of antibodies, mainly IgG [62]. Plasma cells release immunoglobulin at an extremely high rate of about 2 000 molecules per second. This process continues for several days until the death of the plasma cells [1].

2.5.1 Antibodies

Antibodies are also called immunoglobulin and are secreted by plasma cells [8]. An antibody can bind with a specific subset of antigens at specific regions – epitopes – of the antigen/s. If an antigen and an antibody combine, they do so with a strong attractive force, because the matching areas on each molecule are relatively large [62].

An important feature of the immune system and the concept upon which immunisation is firmly grounded is the fact that antibodies can cross-react with related antigens if their epitopes are sufficiently similar [62].

Antibodies act in three different ways: they can directly attack an antigen, activate a complement system, which destroys an antigen, or activate the anaphylactic system, which changes the environment around the invading antigen and reduces its virulence [1].

Each immunoglobulin, irrespective of the class to which it belongs, is composed of two heavy and two light polypeptide chains, each with constant and variable domains [8, 1]. Disulfide bonds join the chains, and the molecules bind into a Y configuration [62]. The Y-shaped region can be divided further into a variable (V) region and a constant (C) region.

An antibody binds to small segments of an antigen (called an epitope) through its variable regions (paratopes), and, consequently, a high diversity of amino acids occurs in the V region [8, 62]. Hyper-variable regions within V regions contain idiotypic determinants to which natural antibodies (referred to as anti-idiotype antibodies) bind, creating an important mechanism for the regulation of B-cell responses [62]. The C region contains a relatively constant sequence of amino acids and is different for each class of immunoglobulin (Ig). [62]. Owing to its consistency, the amino-acid sequence of an antibody's C region determines the isotope of the antibody's immunoglobulin class.

There are five main classes of antibodies (see Figure 3), with corresponding subclasses. Each main class serves a different function:

- IgM is the first antibody formed when the body is exposed to a new antigen, that is, during the body's primary response. It consequently protects the intravascular space from disease and serves as an antigen receptor on a B-cell membrane.
- IgG is the major antibody produced during the body's secondary response and protects tissues from bacteria, viruses and toxins.
- IgA is found in mucous secretions (saliva, tears etc.) and provides an early antibacterial and antiviral defence.
- IgD is found in extremely low concentrations on the surface of developing B-cells. IgD is believed to be an important component in the B-cell maturation process.

- IgE is found primarily in respiratory secretions. IgE interacts with mast cells. It is believed that when two IgE molecules encounter an allergen, they deregulate and cause an allergic response.

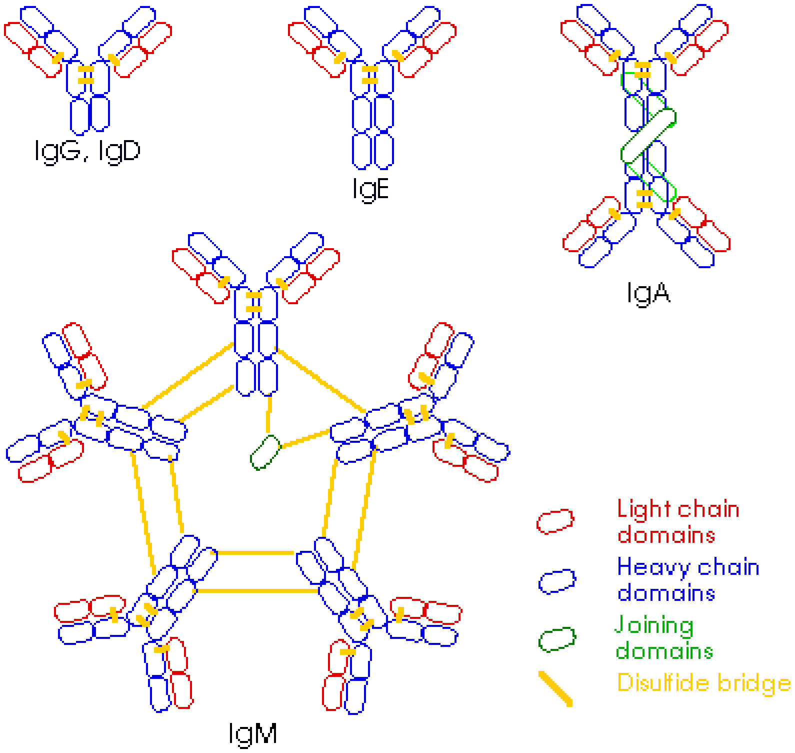


Figure 3. *Immunoglobulin classes (this image was taken from [8])*

The way in which B-cell genes are combined enforces diversity, because of the nature of the generation process. Gene segments are disconnected and must be juxtaposed during B-cell maturation. The potential diversity is further increased by somatic point mutations and joining inaccuracies between the various antibody segments [62].

The detection rate of an antibody molecule is increased further, owing to the number of degrees of flexibility that it attains as a result of freedom of movement, which allows it to bind easily to various regions on the surface of an antigen [8].

An antibody can inactivate an antigen in several ways, most notably by agglutination, precipitation, neutralisation and lysis [1, 62]:

- Agglutination occurs when multiple antigenic agents bind to form a clump of antibodies.
- Precipitation occurs when the antigen-antibody complex becomes insoluble and precipitates.
- Neutralisation occurs when the antibody covers the toxic sites of the antigen.
- Lysis occurs when antibodies directly attack the membranes of foreign cells and causes them to rupture.

Most of the body's protection comes from the complement and anaphylactic systems.

2.5.2 Regulation of Human Immune Responses

Regulation of the immune system is critical to prevent unlimited antibody production, which could result in self-destruction. The human immune system is regulated by a number of factors, namely, the disappearance of the pathogen/foreign substance, the idiotypic network of antibodies and cytokines. Anti-idiotypic activity occurs because the V regions of each antibody's molecules recognise one another. This activity blocks B-cell receptors, thus suppressing not only further activation of the cell, but also the production of idiotypic antibodies [62].

Both cellular and humoral immunity are essential for survival and contain a number of explicit differences, as well as implicit differences. One of the most striking differences between cellular and humoral immunity is persistence. Antibodies only last a few months or years at most, whereas it is believed that sensitised T-cells last almost indefinitely [1].

2.6 Conclusion

The NIS is an intricate network of components, which work collectively to protect the body against the virtually limitless antigens in and around the environment. Pathogens are eliminated primarily through phagocytosis, which is facilitated by the pathogen being coated with antibodies and complement proteins. The immune response is regulated by successfully evolving efficient antibodies, with a high affinity towards the pathogen. Antibodies are manufactured by B-cells, which, in their turn, are aided by T-cells. There are a number of complex chemical reactions involved in and around the environment to support this process.

This chapter provided a brief overview of the innate and adaptive immune systems and the components governing the behaviour of these systems. It is critical to understand the NIS before one delves into AIS theory, since an AIS abstracts a subset of processes occurring within the NIS.

The next chapter defines what an AIS is in more detail and portrays the main research fields within AIS theory.

Chapter 3
Foundations of Artificial Immune Systems

"It is not knowledge, but the act of learning, not possession but the act of getting there, which grants the greatest enjoyment."

- Carl Friedrich Gauss

Artificial immune systems (AISs) emerged in the 1986 as a new computational intelligence paradigm [30]. An AIS can be defined as a system of interconnected components, which emulate a particular subset of aspects originating from the NIS in order to accomplish a particular task within a particular environment/domain. Scientists have achieved great success in finding algorithmic solutions to complex problems by emulating and transposing mechanisms occurring naturally within biological systems to alternative real-world domains. For example, by studying the social behaviour of birds within a flock, scientists have created a class of algorithms termed particle swarm-optimisation algorithms, which have been very successful in solving complex optimisation algorithms [27]. An AIS is by no means an exception and tries to leverage desirable characteristics from the NIS [17, 32]. The key features of a NIS that are of interest to computer scientists have been summarised by both Wierzchoń [87] and Forrest *et al.* [34], and are presented below:

- **Multi-layered protection**: The humoral immune system provides a number of protective layers against antigens namely innate immunity, humoral immunity and cellular immunity. Classical security systems are monolithic and ordinarily address security threats only at a specific level. For example, a system containing only basic password authentication addresses security only at a single level.
- **Distributed detection**: The immune system is autonomous. This is highly desirable within any computer system, because there is no single point of failure.

- **Learning ability**: The NIS is adaptive and develops memory B-cells in response to encountered antigens. Interestingly, the learning ability of AIS algorithms has often been compared to other computational intelligence learning paradigms, such as neural networks [16, 19].
- **Unique localised copies of the detection system**: Each individual lymphocyte within a population of lymphocytes consists of its own protective cells and molecules.
- **Generalisation ability**: Each lymphocyte has the ability to recognise antigens that are structurally similar to previously encountered antigens, the concept on which immunisation is firmly based. The immune system is also able to generate and improve on receptors, clonal selection of B-cell lymphocytes during the primary response, to eradicate antigen that have not yet been encountered.
- **Imperfect detection**: The immune system achieves a very high reliability rate at a low cost, owing to roughly distributed detection (a perfect match is not required between an antibody and an antigen in order for the antibody to recognise the antigen), coupled with the need for minimal communication among immune system artefacts.
- **Self-organisation**: The memory cells of the NIS are organised into an idiotypic network, which changes over time (see [52]).
- **No need for negative examples**: Many learning algorithms require both positive and negative learning examples in order to differentiate correctly between self/non-self patterns, whereas the NIS requires only self-samples.
- **Uniqueness**: Every individual has a unique immune system, which depends on the antigens which an individual has been exposed to in their lifetimes.
- **Explicit symbolic representation:** The knowledge acquired by the NIS is represented explicitly by the structure of receptors on the surface of lymphocytes.

- **Robustness:** In addition to the above characteristics of the NIS, Hofmeyer *et al.* have noted that the natural immune system is robust, owing to its diversity, distributed detection, error tolerance, self-awareness and adaptability [47].

This chapter presents the foundations of AISs and provides a brief overview of the most popular AIS algorithms in section 3.2. Recognition within AISs is then discussed in section 3.4 by focusing on the shape space theory, detector generating techniques, matching functions employed by AISs, and undetectable regions/holes induced by the matching functions. Finally AIS algorithm performance metrics are presented in section 3.5 which discusses traditional AIS performance metrics such as true positives, false negatives etc. in addition to non-traditional AIS performance metrics such as Generalisation and overfitting.

3.1 Artificial Immune-system Foundations

The focal point of most AIS research is on how lymphocytes (B-cells and T-cells) mature, adapt, react and learn in response to a foreign antigen. A taxonomy of the main AIS models was provided by De Castro and Timmis [22], where AIS models were viewed as being either population based or network based [22].

Population-based algorithms such as the negative selection algorithm (NSA) [33] and the clonal selection algorithm (CSA) [21] focus primarily on generating an initial population of lymphocytes, and on improving and refining that population based on techniques emulated from the NIS.

Network-based models [44, 52, 59, 90] are based on anti-idiotypic activity within the NIS, which consequently regulate the population of lymphocytes. The concept of artificial immune networks (AINs) was first proposed by Jerne in 1974. Jerne stated that dynamic behaviour could be observed within the NIS even in the absence of antigens; this suggested that B-cell lymphocytes have the capability of recognising one another [52]. This dynamic behaviour was modelled mathematically using differential equations, continuous models [30] or discrete models (based on a set of iterative equations) [84].

A similar taxonomy was proposed by Dasgupta [15], but his taxonomy does not differentiate between population-based and network-based algorithms. Instead, Dasgupta's taxonomy focuses on each AIS model, including Matzinger's danger theory [60, 61]. By fusing elements of both Dasgupta's and De Castro's taxonomy, a general taxonomy of AIS research models proposed by this thesis is presented in Figure 4.

A short overview of each AIS algorithm is presented in section 3.2

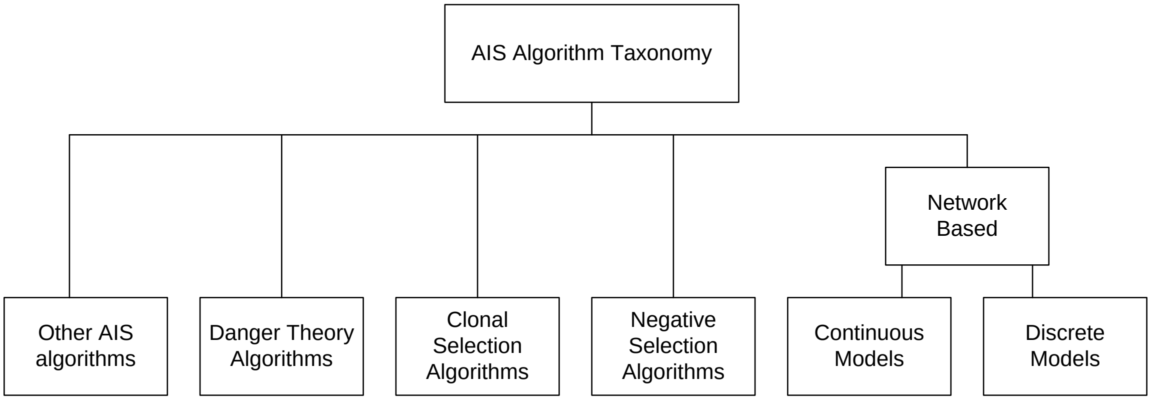


Figure 4. *Taxonomy of artificial immune system algorithms*

3.2 Overview of Different AIS Algorithms

This section provides a brief description of the AIS algorithms illustrated in Figure 4.

3.2.1 Clonal Selection Algorithms

Clonal selection algorithms [20, 21] are inspired by the way in which B-cell lymphocytes adapt in response to an antigen encounter (see section 2.5). AIS researchers generally refer to this process as clonal selection [20, 21]. The main points of the clonal selection process are highlighted below:

- B-cells are activated upon recognising antigens and receiving signals from T-helper cells.
- Upon activation, B-cells rapidly proliferate and mature into plasma cells. The proliferation rate of a B-cell is directly proportional to the affinity between the B-cell and the recognised antigen. This will result in high-affinity B-cells to

produce a large number of clones, whereas low-affinity B-cells will produce a diminutive number of clones.

- The progeny/clones of the parent B-cells then undergo a mutation process. The areas subject to mutation are the portions of the receptor that bind with the antigen. Since lymphocytes are somatic cells, being cells not involved in reproduction, the mutation is termed somatic mutation. Berek and Ziegner [6] found that the somatic mutation is inversely proportional to the affinity between the antigen and the antibody; that is, the higher the affinity is, the lower the mutation rate is, and the lower the affinity is, the higher the mutation rate is. This mutation principle allows the immune system to preserve cells that have a high affinity to the antigen and to improve those cells that have a low affinity to the antigen.
- A selection mechanism existing within the NIS guarantees that offspring that are better than their parents at recognising the antigen are selected as long-term memory cells. Memory cells in effect are a means by which the B-cells retain information about the antigen and can thus react more quickly upon subsequent encounters with the same antigen. Learning does not cease once memory cells have been formed, and each successive encounter with the same antigen improves the immune system’s capability to recognise that antigen.

The first version of an algorithm that was inspired by the clonal selection process, called CLONALG, was presented by De Castro *et al.* [20, 21]. Interestingly when compared to the performance of evolutionary strategies (see [26]), Walter and Garrett [85] found that, in low dimensional landscapes, the CSA performed better than its evolutionary counterpart did. In other studies, CLONALG is viewed as the root of all CSA algorithms, and a number of subsequent improvements have been made to CLONALG, as mentioned below:

- The CLONALG was found by White and Garrett [86] to be extremely good at classifying unseen examples from binary data sets, if given enough data albeit being computationally expensive. A modified CLONALG, called clonal classification (CLONALGAS) was proposed by White and Garrett [86] in an attempt to improve upon the computational complexity exhibited by CLONALG. Although the CLONALGAS algorithm has the same complexity as CLONALG it is able to converge on a solution set in fewer iterations.

- A dynamic clonal selection algorithm was created, by Kim and Bentley [57], to overcome difficulties faced in anomaly detection owing to changing environments. The algorithm was later extended [56] to delete memory detectors that are no longer valid. This resulted in a reduction of the false-positive rate exhibited by the original dynamic clonal selection algorithm.

3.2.2 Negative Selection Algorithms

Negative selection algorithms are inspired by the T-cell maturation process occurring within the NIS. Since the work presented in this thesis is rooted in the negative selection theory, a comprehensive overview of negative selection is given in chapter 4.

3.2.3 Immune Network Algorithms

Immune network algorithms are based on Jerne's network theory [52], which asserts that B-cells are capable of recognising one another. Immune network algorithms attempt to maintain a population of detectors by using a set of mathematical equations and are applied mainly to clustering problems [30, 59, 90]. A well-known immune network algorithm, called AINET, was developed by De Castro and Von Zuben [18] and was shown to be extremely effective at compressing an input space comprising of N antibodies, while still preserving the topology of the problem space.

3.2.4 Danger-theory Algorithms

The danger theory introduced by Matzinger [60, 61] asserts that there must be discrimination within the immune system that goes beyond the classical view of self/non-self discrimination. Matzinger asserts that the immune system in fact discriminates "some self from some non-self". Matzinger's theory is illustrated in Figure 5, which shows four different cells: a B-cell lymphocyte (represented by the symbol B), a T-helper lymphocyte (represented by the symbol Th), a T-killer lymphocyte (represented by the symbol Tk) and an antigen-presenting cell (represented by the symbol APC).

Each immune system artefact in Figure 5 conforms to the following three laws:

- A cell is activated upon receipt of signal 1 and signal 2, that is, if a B-cell produces antibodies when it is activated. A cell dies if it receives signal 1 in the absence of signal 2. A cell ignores the receipt of signal 2 if it did not receive signal 1.
- Signal 2 can be accepted only from antigen-presenting cells, or, for B-cells, from T-helper cells. B-cells can also act as APCs for long-lived T-cells. Signal 1 can originate from any cell.
- Activated cells revert to a resting state after some finite time period.

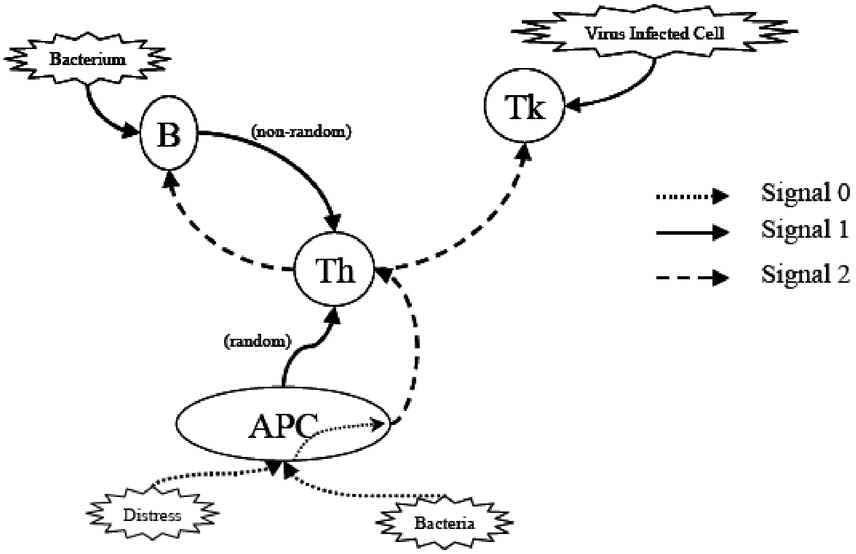


Figure 5. *Illustration of Matzinger's danger theory (this image was taken from [3])*

Greensmith *et al.* implemented an AIS utilising Marzinger's danger theory which emulates the behaviour of dendritic cells (refer to section 2.2.7 for a discussion on dendritic cells) [42, 43]. Their algorithm, termed the Dendritic Cell Algorithm (DCA), differs from other AIS algorithms in the sense that is inspired by the innate immune system as opposed to the adaptive immune system.

An example of a NSA that implements a simplified form of the signalling concept illustrated in Figure 5 was is by Hofmeyer and Forrest [48]. In their system, detectors can be in one of several states (randomly created, immature, mature and naïve, activated or dead). When a detector is activated by a foreign agent for the first time, the detector transitions from a mature state to an activated state. Thereafter, there is a fixed-time waiting period for a signal, originating from a human operator, in the form of an e-mail, to confirm that the foreign agent is indeed an anomaly. If the signal is received before a particular time period, then the detector will be retained by the system as a memory detector; otherwise the detector is purged from the system.

3.2.5 Other Immune Algorithms

Each of the AIS algorithms discussed thus far subscribes to a particular niche within the NIS and only emulates a subset of mutually exclusive functionality from the NIS. There is a subclass of AIS algorithms, called hybrid AIS algorithms. These algorithms try to emulate as much of the NIS as possible in order to derive from the benefits of the NIS in its entirety. An example is the multilevel immune learning algorithm (MILA), created by Dasgupta *et al.* [14]. MILA creates three different types of ALC, namely, T-helpers, T-suppressors and B-cells, and thus draws on both negative and clonal selection theory.

3.3 Generation of Detectors within an AIS

Most AIS algorithms generate random ALCs. This, however, is not reflective of what happens within the NIS. The NIS stores the genetic material for an individual antibody in seven separate libraries. An antibody molecule is produced by randomly selecting genetic components from each of the individual libraries. Hightower *et al.* [46, 65] wrote a genetic algorithm to evolve gene libraries (refer to [37] for more information on genetic algorithms) to study the characteristics that both gene libraries and ALCs generated from gene libraries exhibit, and found that:

- Gene libraries became increasingly dissimilar as they evolved.

- If gene libraries were exposed to a much greater proportion of antigens, the gene libraries evolved more rapidly and generated ALCs of a better quality. That is, the ALCs generated from the gene libraries were able to recognise a greater proportion of antigens.

Their algorithm was modified by Hofmeyer *et al.* [48] through the addition a mutation step and found that the mutation step increased the speed at which gene libraries converged to a specific solution. In another follow-up study, Opera and Forrest [66] found that an increase in the size of the genome of an antibody library increased the survival probability of an individual by increasingly smaller amounts. It is argued by Opera and Forrest that antibodies produced by antibody libraries reflect different antigen clusters. In a separate study, Forrest and Perelson showed how the degree of generalisation (generalisation is discussed in section 3.5.2) in an evolved antibody population can be controlled [31]. To date, very little work has been done on using gene libraries in conjunction with AIS algorithms to generate detectors. The reason is probably that such an AIS algorithm would be more complex than its random counterpart. That is, the algorithm would first need to evolve a set of gene libraries before being able to execute. An example of a recent AIS algorithm, which uses gene-libraries to classify e-mails, can be found in [76].

A large percentage of the AIS algorithm classes illustrated in Figure 4 are rooted in a number of similar key concepts. Each of these concepts is discussed in section 3.4, where a stronger emphasis is placed on concepts that are applicable to binary problem spaces, as opposed to real-valued problem spaces.

3.4 Recognition within an Artificial Immune System

Lymphocyte cells in the NIS (T-cells or B-cells) present receptor molecules on their surface responsible for recognising the antigenic fragments displayed by the pathogens. Pattern recognition within the NIS occurs on a molecular level, and covalent bonds form between the antigen and T-cells and B-cell antibodies.

In order to model this aspect, most AIS algorithms employ the shape-space concept proposed by Perelson and Oster [69], which allows a quantitative description of the interactions of receptor molecules and antigens. Perelson's and Oster's shape-space theory is discussed in detail in section 3.4.1.

3.4.1 Shape-space Theory

A population of N individuals (cell receptors) can be represented as a finite volume, V, containing n points (where n is the dimensionality of an individual/artificial lymphocyte). Each individual has a volume, V_r, surrounding it such that any complementary antigen that lies within V_r is recognised by the individual. The term V_r is called the detection area, and its size depends on a parameter, r, known as the affinity threshold. In this shape space, an artificial lymphocyte/receptor molecule is depicted as a vector, $\mathbf{x}$, with coordinates $(x_1, x_2, \dots, x_n)$, whereas an antigen/non-self artefact is depicted as a vector, **y**, with coordinates $(y_1, y_2, \dots, y_n)$. Perelson's and Oster's [69] shape-space theory is illustrated in Figure 6, which portrays two antibodies and their associated detection regions.

The shape-space theory enables AIS researchers to represent antigens and immune system artefacts (lymphocytes and antibodies) mathematically. Building on the shape-space theory, researchers within the AIS community have studied a variety of approaches in which the affinity between an antigen and an antibody can be captured. The most popular approaches are discussed in section 3.4.2.

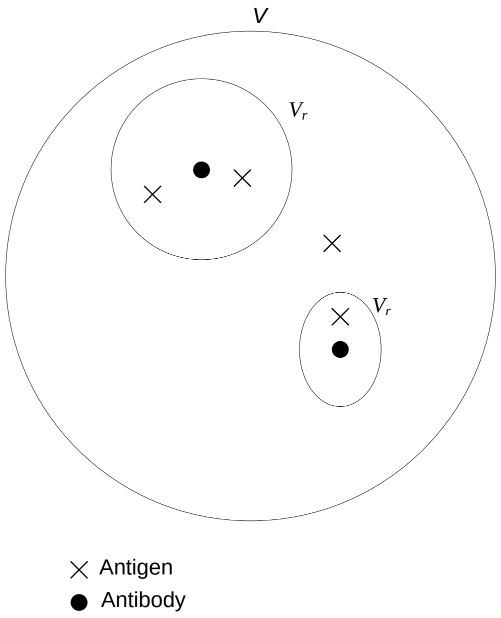


Figure 6. *Shape-space theory*

3.4.2 Affinity Threshold

The affinity-threshold functions presented in this section attempts to mimic the bonding process that occurs withinin the NIS. When an antigen and receptor bind covalently, they do so with varying degrees of strength depending on how well the receptor's V region can recognise the antigen. The bonding process that matching rules emulate is diagrammatically depicted in Figure 7 (refer to section 2.5.1 for a more detailed description of the bonding process). Take note that in figure 8 the terms Ab and TCR refers to antibody and T-cell receptor respectively.

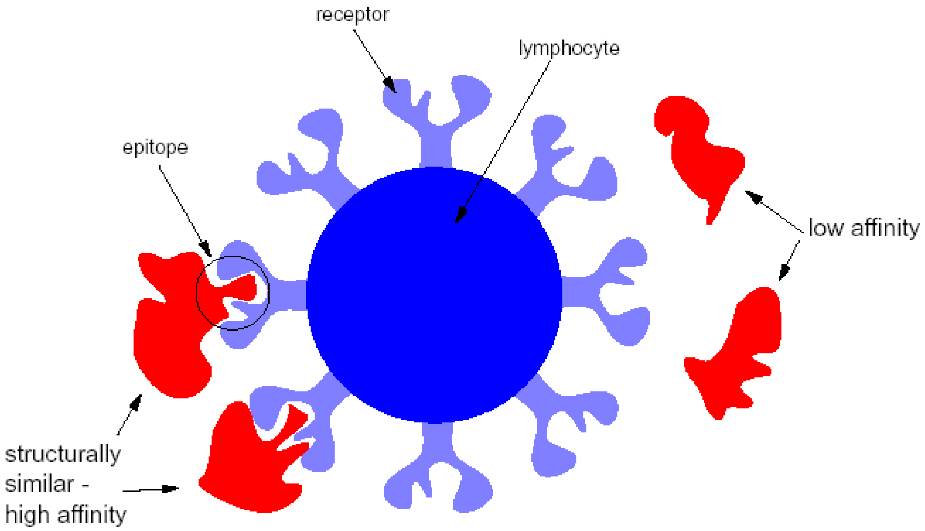


Figure 7. *Antigen/antibody bonding (this image was taken from [49])*

Given an artificial lymphocyte, **x**, and an antigen, **y**, a number of matching rules can be defined to determine whether **x** and **y** match.

3.4.2.1 Hamming-distance Rule

The hamming distance (HD) between two binary vectors is the number of corresponding bits that differ. For example, if $\mathbf{x} = (1,0,0,1)$ and $\mathbf{y} = (1,1,0,1)$ then the hamming distance between **x** and **y**, $f_{HD}(\mathbf{x}, \mathbf{y})$, is 1.

3.4.2.2 *r*-Contiguous Bits Rule

The r-contiguous bits (RCBITS) rule states that two binary vectors match if they have identical bits in at least r contiguous positions [68].

For example; if $\mathbf{x} = (1,0,1,0,0,0,0,0)$ and $\mathbf{y} = (0,1,1,0,0,1,1,1)$ then the number of r contiguous bits between **x** and **y**, $f_{RCBITS}(\mathbf{x}, \mathbf{y})$, is 3.

The RCBITS rule is a very popular matching rule, since it is conceptually simple and lends itself equally to both mathematical and statistical analysis [87].

3.4.2.3 *r*-Chunks rule

The r-Chunks (RCHK) matching rule, as conceptualised by Balthrop *et al.* [5], was inspired by the RCBITS rule and matching rules for classifier systems developed by Holland [50]. RCHK detectors are specified by a window of size r in which all r bits in the window must match the given string in question. The remaining bit positions are termed "don't cares" and are ignored. An RCHK detector is depicted as a vector, $\mathbf{x}$, of length r and a starting position w (that is, detection starts at position w and ends at position $w + r - 1$).

For example; if $\mathbf{x} = (1,0,0,1)$ and $\mathbf{y} = (1,0,0,1,0,0,0,0)$ then $f_{RCHK}(\mathbf{y}, \mathbf{x}, 1) = 4$, where $f_{RCHK}(\mathbf{y}, \mathbf{x}, 1)$ is the application of the RCHK rule to vectors $\mathbf{x}$, $\mathbf{y}$, and $w = 1$.

A number of experiments were performed by Balthop *et al.* where they concluded that the RCHK rule performed better than the RCBITS rule for their data set [5]. It was shown that the RCHK rule subsumes the RCBITS rule by Esponda *et al.* [29].

From each of the detection rules presented thus far (HD, RCBITS and RCHK) it is evident that a trade-off between the number of detectors and their affinity threshold exists. Larger affinity thresholds result in more specific matching, whereas smaller affinity thresholds result in more generic matching. In other words the best values for these parameters are problem dependant and should be fine tuned for each new problem domain [47].

An additional trade-off, called holes or undetectable strings, exists for the RCHK and RCBITS rule. The origins of holes are discussed further in section 3.4.2.4.

3.4.2.4 Holes induced by the *r*-Contiguous Bits and *r*-Chunks Rule

A critical difference between the RCHK and RCBITS matching rule noted by Balthrop *et al.* [5] was the number of undetectable strings that they induced. These undetectable strings are called "holes" by D'haeseleer *et al.* [11]. An example of holes is illustrated in Figure 8, where the detection region, V_r, or each detector, is shown as a circle in dark grey, an antigen is shown in light grey and holes are indicated as white space.

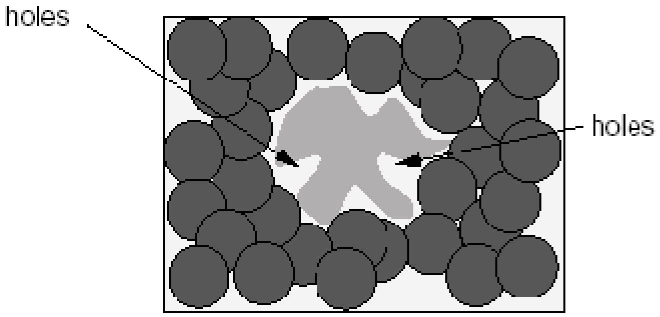


Figure 8. *Undetectable regions induced by holes (this image was taken from [47])*

Holes do not exist merely because of the limitations of the matching rules used, but also because of similarity existing between self and non-self cells. If was noted by Hofmeyer [49] that in reality, self and non-self are distributed at great distances from each other. The fact that that holes can exist for any approximate matching rule, even within the NIS, because binding between receptors and peptides is approximate [47].

Two different types of holes were identified by Balthrop *et al.* [5]: length-limited holes and crossover holes. Crossover holes and length-limited holes are discussed in section 3.4.2.5 and section 3.4.2.6, respectively.

3.4.2.5 Crossover Holes

A crossover hole, $\mathbf{h}$, occurs when all possible windows (the specified r contiguous positions of a RCHK detector see section 3.4.2.3) within $\mathbf{h}$ are crossovers (defined below) of adjacent windows within a particular set of vectors (where the set is either a self-set or a non-self set depending on the AIS algorithm being employed).

Given a set, S, of self strings and two vectors, $\mathbf{u}$, $\mathbf{v} \in S$, a crossover occurs between two adjacent windows, $\mathbf{w}_i = (v_i, v_{i+1}, \dots, v_{i+r-1})$, and, $\mathbf{w}_{i+1} = (u_{i+1}, u_{i+2}, \dots, u_{i+r})$, whenever bits $v_j = u_j \, \forall_j : i + 1 \leq j \leq i + r - 1$ [5].

To illustrate this concept, consider a self-set, $S = \{1001{,}1100\}$, an affinity threshold, $r = 2$, and a function, $W(s, r)$, which returns a set of all possible windows of length r, occurring within a string $s \in S$. A graph, G, can be constructed by applying $W(s,r)$ to each $s \in S$ as follows: the set of windows returned by the application of $W(s,r)$ to s are added as interconnected nodes to an individual level in G pertaining to the index of s in S (where the index of the first string in S is one). Such a graph is depicted in Figure 10. In Figure 10 whenever two windows crossover (according to the crossover definition) they are connected with a bold line.

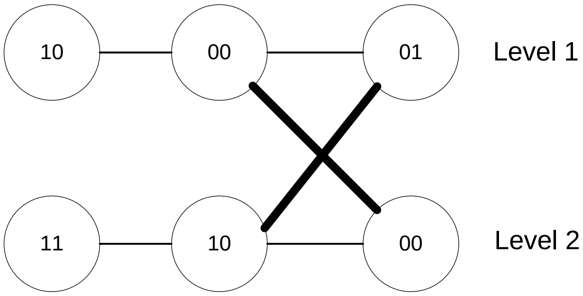


Figure 9. *Crossover-window graph*

By traversing the paths in the graph in Figure 9 from the leftmost nodes to the rightmost nodes (on each level starting at level 1), the following set of strings, $S^{'}$, can be generated: $\{1001,1100,1000,1101\}$. The subset $S^{''} \in S'$ consists of crossover holes: $\{1000,1101\}$.

3.4.2.6 Length-limited Holes

Length-limited holes are holes that arise in full-length detectors, for example detectors which employ the RCBITS detection rule. A length-limited hole is defined as a string, $\mathbf{h'}$, which contains at least one window of r bits not present within the self-repertoire and for which a detector cannot be generated [5].

A length-limited hole was illustrated by Balthrop *et al.* [5] by means of the following example:

Let $S = \{110,010\}$ and $r = 2$. Then it would be impossible to generate a detector for a non-self string defined as 011 because the generated detector would match a self-string.

It should be noted that length-limited holes occur when the variance between self and non-self is particularly low [5].

3.4.2.7 Overcoming Holes

Holes reduce the overall detection capability of an AIS algorithm and are hence problematic. The RCBITS rule induces both length-limed and crossover holes, whereas the RCHK rule can induce only crossover holes [5].

When one attempts to overcome holes, for instance in the RCBITS rule, it is tempting to choose a value of r that is equal to n (the length of a particular antigen), because if $r = n$, then it is impossible for length-limited holes to occur. Such a stringent value for r however, carries an unfortunate consequence, in that the artificial lymphocyte loses its generalisation capability, because it has become too specialised. It has been speculated by Hofmeyer and Forrest [47] that MHC plays an important role within the NIS to protect a population of artificial lymphocytes/detectors from holes. In Hofmeyer and Forrest's view, MHC is a mechanism through which a single protein can be represented in a different way. Hofmeyer and Forrest went further to conclude that, because different representations induce different holes, leveraging multiple representations will reduce the overall number of holes.

Hofmeyer and Forrest implemented the MHC mechanism by defining a permutation mask, $\mathbf{m} = (1, \dots, m_n)$, where each $m_i \in \{1, \dots, n\}$ specifies a new position for bit number i. The function $f_{PERMUTE}(\mathbf{w}, \mathbf{m})$ applies a permutation mask to a vector, $\mathbf{w}$. The permutation function is applied by generating a single random permutation mask, $\mathbf{m}$, for the entire global population of artificial lymphocytes/detectors. Each antigen, $\mathbf{y}$, is first processed by $f_{PERMUTE}(\mathbf{y}, \mathbf{m})$, before being introduced to the population of detectors/artificial lymphocytes [47].

The analogy of a permutation mask is illustrated in Figure 10, which shows three different detector sets comprising different V_r regions, because each detector set employs a different permutation mask.

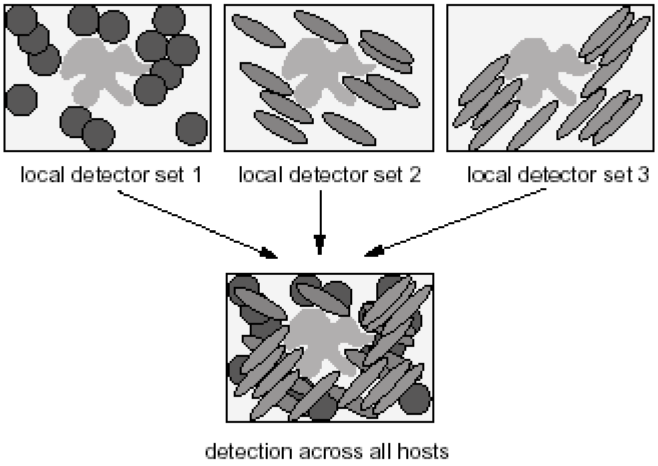


Figure 10. *Graphical illustration of how permutation masks change the shape of detectors (this image was taken from [47])*

A detailed study of the effect of Hofmeyer and Forrest's permutation mask used in conjunction with the NSA was performed by Stibor *et al.* [80]. It was found that randomly generated permutation masks changed the shape and distribution of the entire data set, thus distorting its semantic meaning and resulting in artificial lymphocytes being randomly distributed within the search space, as opposed to being concentrated around self-regions. Furthermore, Stibor *et al.* also doubted whether permutation masks were appropriate at reducing the number of holes within negative selection algorithms by abstracting diversity.

Contrary to the view of Stibor *et al.*, Esponda *et al.* [29] showed that the NSA, under the RCBITS rule augmented by permutation masks could recognise the same set of languages of that of the NSA under the HD rule, showing that permutation masks in fact did reduce holes induced by the RCBITS rule.

The fundamental difference between the views of Esponda *et al.* and Stibor *et al.* is caused by the approach in which they investigated the efficacy of the RCBITS rule. In the former case, Esponda *et al.* approached the investigation with mathematics, whereas in the latter case Stibor *et al.* approached the investigation empirically.

It is the view of this thesis that both Stibor *et al.* and Esponda *et al.* are correct in that permutation masks do eradicate holes induced by the RCBITS rule if the permutation induced by the permutation mask is meaningful. That is, the permutation mask should select both adjacent and non-adjacent bits where a relationship exists between the values of the attributes (in the order induced by the permutation). For example, if a relationship exists between attributes (1, 3, 5, 2, 4) of a five-dimensional problem space, then it is logical to create a permutation mask that induces such a permutation on the entire self repertoire of strings before generating detectors (by utilizing the NSA).

It was argued by Esponda *et al*. [29] that such a permutation is indeed very difficult, if not computationally expensive, to infer. This is in fact not true, and an approach to generate meaningful permutation masks is illustrated by this thesis.

The first part of this chapter discussed how AIS algorithms imitate the affinity relationship between antigens and artificial lymphocytes by exploiting Perelson's and Oster's shape-space theory in conjunction with a matching function. The final part of this chapter focuses on how researchers typically classify the performance of an AIS algorithm.

3.5 AIS-algorithm Performance Metrics

A variety of performance measures currently exist for different AIS-algorithm classes (see Figure 4 for a taxonomy of different AIS classes). Since the work presented by this thesis resides within negative selection theory, the most popular metrics applicable to negative selection theory are presented here. This section also formally introduces two additional metrics which measure the amount of generalisation and overfitting within an AIS. Take note that since the NSA is a classification algorithm, thus the metrics presented in this section are also applicable to a variety of other AIS algorithms which perform one-class classification. Applications of AIS algorithms are by no means limited to classification and have been applied to area such as data cluster, image compression and job scheduling [18, 44].

3.5.1 Popular Negative Selection-theory Performance Measures

The most popular metrics employed by AIS researchers to report on the performance of NSAs and a number of other AIS algorithms are false positives, true positives, false negatives and true negatives:

- False positives (FPs) occur when self-patterns are incorrectly classified as non-self.
- True positives (TPs) occur when self-patterns are correctly classified as self.
- False negatives (FNs) occur when non-self patterns are classified as self.
- True negatives (TNs) occur when non-self patterns are correctly classified as non-self.

These measures can be combined in a more meaningful way to create two additional metrics, termed the detection rate (DR) and false-alarm rate (FR), defined as [82]:

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

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

3.5.2 Generalisation

Generalisation within AIS literature is defined as the comprehensive set of strings that a generated detector is activated by [5]. Generalisation is defined by this thesis as the ability of an AIS algorithm to correctly classify patterns that were not included in the original training set as either being self or non-self patterns, in other words $FP + FN = 0$ for the test set. This concept can easily be demonstrated by means of Perelson and Oster's shape-space theory [69] in Figure 11. Each antibody in Figure 11 has a detection region of size V_r within the entire shape space, V, where V_r is the antibody's generalisation region. The closer the antigen is to the antibody, that is, the centroid of regionV_r, the stronger the affinity of the bond between the antibody and antigen is. The generalisation capability of the AIS in Figure 11 is the union of each antibody's detection region $V_{r.}$

It should be noted that the size of an antibody's detection region, V_r, is a function of the magnitude of the affinity threshold, r, utilised by the antibody's detection rule. For example, if an AIS algorithm uses the HD rule, then greater r values equate to smaller detection regions. Thus by reducing the generalisation capability of an individual artificial lymphocyte and, consequently, the entire AIS algorithm. Pseudocode for an algorithm that can estimate the average Generalisation capability of an AIS is presented by this thesis in the figure below.

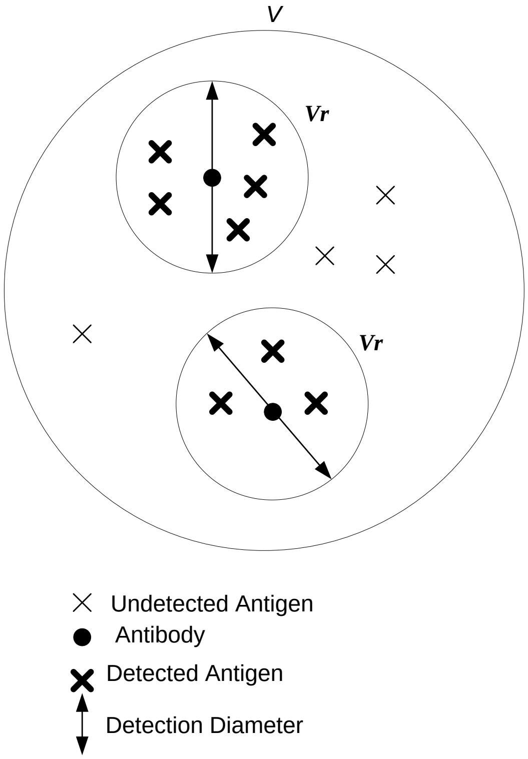


Figure 11. *Graphical depiction of generalisation in terms of the shape-space theory*

The ability of an immune system to generalise is the primary reason for vaccines having been so successful in protecting human beings against a wide plethora of antigen. The premise on which vaccination is based is that in order to protect the human body against a particularly dangerous antigen, $\mathbf{x}$, a harmless structurally similar antigen, $\mathbf{y}$, is introduced into the immune system to allow the immune system to develop antibodies against $\mathbf{y}$. By doing so, the human body will naturally be able to expel an encounter of $\mathbf{x}$, because of its Generalisation capability. Forrest *et al.* [5] noted, however, that generalisation within an AIS must be strictly controlled to ensure that the number of false positives generated by an AIS are sustained at an acceptable level. Overfitting is a term that is strongly related to generalisation and is discussed in section 3.5.3.

Given a particular AIS algorithm A' and a set N_S comprising non-self vectors, the average generalisation capability of A' over N_S can be estimated using the following algorithm:

```
Randomly partition $N_S$ into two disjoint subsets $D_{TRAIN}$ and $D_{TEST}$ such that
$N_S = D_{TRAIN} \cup D_{TEST}$ where $|D_{TRAIN}| = 0.7 * |N_S|$ and $|D_{TEST}| = 0.3 * |N_S|$
Train $A'$ using $D_{TRAIN}$ ;
Let $C$ be a set of artificial lymphocytes generated as an output of $A'$;
Let $g$ (a measure of the average generalisation of an AIS algorithm):= 0;
for each artificial lymphocyte $\mathbf{x}_i \in C$ do
    for each $\mathbf{y}_j \in D_{TRAIN}$ do
        if $\mathbf{x}_i$ is activated by $\mathbf{y}_j$ then
            $g \coloneqq g + 1$;
        end
    end
end
$g \coloneqq \frac{g}{|C|}$;
```

Figure 12. *Pseudocode for estimation of the average generalisation within an artificial immune system*

3.5.3 Overfitting

Overfitting is defined in this thesis as a phenomenon that occurs within an AIS when a large number of detectors memorise the same set of training patters. A direct consequence of overfitting is that the detection capability of the resultant AIS is degraded.

The degree of overfitting, within an AIS, is a function of the magnitude of the affinity threshold, r, used by a particular matching function, in conjunction with how the AIS algorithm distributes its resultant artificial lymphocytes. To illustrate the concept further, two different overfitting scenarios are discussed using Perelson and Oster's shape-space theory.

It is the view of this thesis that the first scenario occurs when two conditions are met:

- The affinity threshold employed by a particular AIS algorithm is sufficiently large such that the detection region, V_r, of each artificial lymphocyte generated by the AIS encompasses a broad area within the resultant shape space, V. Take note that an explicit assumption is being made that a single affinity threshold is utilised as an input parameter by the AIS algorithm in question, which, in turn will generate ALCs with the same affinity threshold provided as an input parameter to the AIS algorithm. If the assumption does not hold true, i.e. each ALC generated by the AIS algorithm has different affinity thresholds, then the condition occurs, when the detection regions of several ALCs encompass a broad area within the resultant shape space, due to the affinity thresholds employed by the ALCs being sufficiently large.
- The majority of artificial lymphocytes are distributed by the AIS algorithm in such a manner that they are in close proximity to one another within the shape space, V.

A direct consequence of the first scenario is that there is a high degree of intersection between a set of artificial lymphocytes and their respective detection regions such that each intersection reduces the efficacy of an individual artificial lymphocyte (see Figure 13 for a graphical illustration).

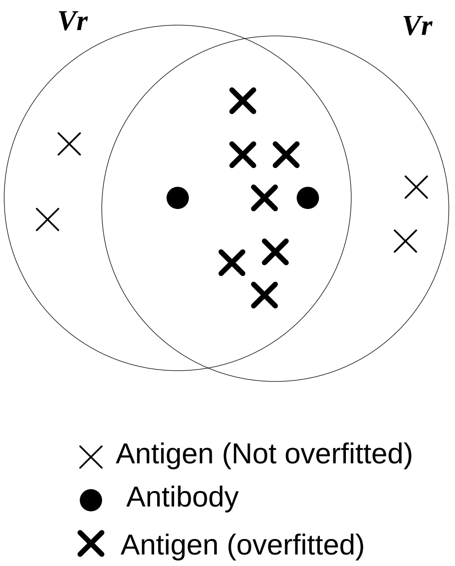


Figure 13. *Overfitting scenario 1: large affinity threshold with two structurally similar artificial lymphocytes, that is, the artificial lymphocytes are in close proximity to one another*

Conversely, a second overfitting scenario can occur when the following two conditions are met:

- The affinity threshold employed by a particular NSA is very small such that the detection region, V_r, of each artificial lymphocyte encompasses a small area within the resultant shape space, V, resulting in highly specialised artificial lymphocytes (with each artificial lymphocyte typically being able to match a maximum of two self/non-self vectors).
- A large number of artificial lymphocytes are generated by the AIS algorithm to ensure that the AIS algorithm still exhibits an acceptable detection rate.

A direct consequence of the second scenario is that the resultant distribution of detectors is localised almost entirely on patterns within the training set. Hence the AIS does not capture a good fit of the training data (refer to Figure 14 below).

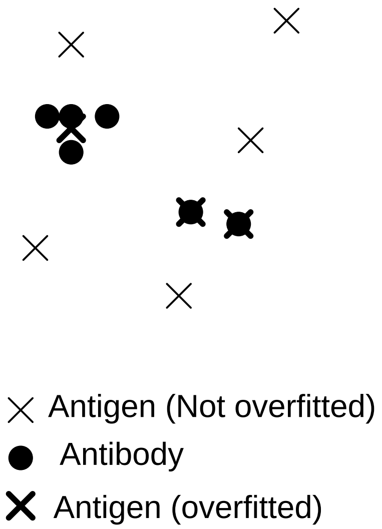


Figure 14. *Overfitting scenario 2: small affinity threshold with a number of structurally similar artificial lymphocytes*

Overfitting within an AIS resulting from overlapping detectors (see overfitting scenario in Figure 14) induces an additional undesirable condition studied by Smith *et al.* [78], which can be explained as follows, given two artificial lymphocytes $\mathbf{x}_1$ and $\mathbf{x}_2$:

Let each antibody have an affinity threshold of r. The shape-space theory stipulates that these artificial lymphocytes can be represented by circles with a radius r. Suppose that the two circles intersect such that a large proportion of the area of $\mathbf{x}_2$ is contained within the area of $\mathbf{x}_1$. This signifies that $\mathbf{x}_1$ can recognise a large proportion of antigens that can be recognised by $\mathbf{x}_2$.

A direct consequence of the above association is that an antigen that lies in the intersection area between $\mathbf{x}_1$ and $\mathbf{x}_2$ will be eradicated by $\mathbf{x}_1$, thereby leaving $\mathbf{x}_2$ with a smaller time window in which to develop memory cells (see the clonal selection theory in section 3.2.1). In fact, a situation could arise whereby $\mathbf{x}_2$ does not produce any memory cells at all. This concept is known as antigenic sin, which many researchers regard as the sole reason for vaccines possibly interfering with one another and thus being counter effective [79].

The extent to which antigenic sin applies to an AIS algorithm depends on a number of factors:

- The mechanisms employed by the AIS algorithm, in other words whether or not the algorithm has a clonal selection component.
- The degree of parallelism for which the algorithm caters.
- The distributed nature of the algorithm.
- The degree of competition between artificial lymphocytes, that is, whether each lymphocyte is given an equal opportunity to undergo the clonal selection process.

Similar to the algorithm presented in Figure 12, an algorithm is presented by this thesis in Figure 15 which provides an estimate of the average amount of overfitting exhibited by an AIS algorithm A' over a data set N_S .

Both overfitting and generalisation within the AIS domain occur because of the associative nature of immunological memory. Overfitting and generalisation within AIS algorithms are very difficult to avoid entirely, although a number of AIS algorithms, especially network-based algorithms such as the AINET algorithm [18], have built-in mechanisms to address overfitting and generalisation.

Given a particular AIS algorithm A' and a N_S comprising non-self vectors, the average overfitting exhibited by A' over N_S can be estimated using the following algorithm:

```
Randomly partition $N_S$ into two disjoint subsets $D_{TRAIN}$ and $D_{TEST}$ such that
$N_S = D_{TRAIN} \cup D_{TEST}$ where $|D_{TRAIN}| = 0.7 * |N_S|$ and $|D_{TEST}| = 0.3 * |N_S|$
Train $A'$ using $D_{TRAIN}$;
Let $C$ be a set of artificial lymphocytes generated as an output of $A'$;
Let $o$ (a measure of the average overfitting of an AIS algorithm) $\coloneqq 0$;
Let $Y_i$ be set of antigens detected by an artificial lymphocyte $\mathbf{x}_i$ where $\mathbf{x}_i \in C$ and
$Y_i = \{y_i \in N_S | f_{HD}(\mathbf{x}_i, \mathbf{y}_i) = true\}$;
for each artificial lymphocyte $\mathbf{x}_i \in C$ do
    Let $o_j$ (the number of non-self patterns overfitted by $\mathbf{x}_i$) := $|Y_i|$;
    for each artificial lymphocyte $\mathbf{x}_j \in C$ where $j \neq i$ and $j > i$ do
        $Y_i = Y_i - (Y_i \cap Y_j)$;
    end
    if $|Y_i| > 0$ then
        $o_j = o_j - |Y_i|$;
    end
end
Once $o_j$ has been calculated for each artificial lymphocyte $\mathbf{x}_i \in C$, calculate $o$ using:
$o \coloneqq \frac{\sum_{\mathbf{x}_i \in C} o_i}{|C|}$;
```

Pseudocode for estimating the average overfitting exhibited by an AIS algorithm

Figure 15. *Pseudocode for estimation of the average overfitting within an artificial immune system*

3.6 Conclusion

The biological processes within the NIS bear numerous favourable characteristics, which artificial immune systems extract and emulate. This, in turn, enables many complex problems that exist within the natural world to be addressed and resolved through the use of AIS algorithms. Despite being a fairly young field, a wide variety of different AIS algorithms is in existence today and has seen a number of successful applications [7, 13, 28, 35, 44, 58, 64, 72, 77, 89]. AIS algorithms have also inspired researchers in other fields to incorporate immune-system theory into their own research paradigms [7, 73].

This chapter discussed the mathematical foundations on which AIS algorithms are based by exploring:

- Perelson’s and Oster’s shape-space theory.
- Detection rules used by AIS algorithms.
- AIS performance metrics (it should be noted that the performance methods most applicable to negative selection theory have been the focal point of this chapter).
- A high-level taxonomy of different AIS algorithms, as well as a brief description of each AIS algorithm class.

The next chapter provides an in-depth view of negative selection algorithm theory, the most popular variants of negative selection algorithms, and how researchers reason about the efficacy of negative selection algorithms.

Chapter 4
The Negative Selection Algorithm

"Any sufficiently advanced technology is indistinguishable from magic."

- Arthur C. Clarke

This chapter discusses the mechanics of the NSA proposed by Forrest *et al.* in section 4.1. The original mathematical equations derived by Forrest *et al.* to quantify the number of resources required by the NSA to exhibit an acceptable detection rate and failure rate (see section 3.5.1) is then discussed in section 4.2.1. Next, the short comings of Forrest *et al.'s* original algorithm is highlighted followed by a broad overview of approaches undertaken by several AIS researchers to address the short comings (see sections 4.3 to 4.4). The chapter concludes by discussing a real-valued NSA (see section 4.5).

4.1 Background on Negative Selection and Positive Selection

Immature T-cells are subjected to both a negative and a positive selection process before being released into the blood stream (refer to section 2.4.2). The NSA was inspired by the negative selection process occurring within the NIS and is conceptually illustrated in Figure 16. The main concept behind the NSA is to generate a set of candidate detectors, C, such that $\forall \mathbf{x}_i \in C$ and $\forall \mathbf{z}_p \in S$ $f_{MATCH}(\mathbf{x}_i, \mathbf{z}_p) < r$ [33].

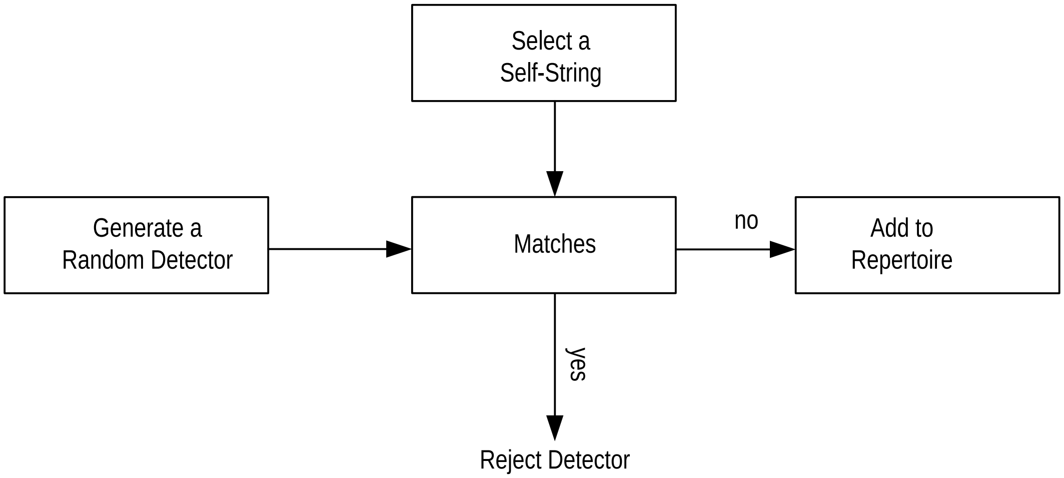


Figure 16. *High-level overview of the negative selection algorithm*

Pseudocode for Forrest's NSA is given in Figure 17.

```
Let counter, $n_c$, be the number of self-tolerant artificial lymphocytes to train;
Let $C$ be an empty set of self-tolerant ALCs.
Create a training set, $D_{TRAIN}$, comprised of self patterns;
while $|C| \neq n_c$ do
    Randomly generate an ALC, $\mathbf{x}_i$;
    matched := false;
        for each self pattern, $\mathbf{z}_p \in D_{TRAIN}$ do
            if $f_{MATCH}(\mathbf{x}_i, \mathbf{z}_p)$ is greater than the affinity threshold $r$ then
             matched:= true;
             break;
        end
    end
    if   matched = false then
     Add $\mathbf{x}_i$ to $C$;
  end
end
```

Figure 17. *Negative selection algorithm pseudocode*

Forrest *et al.*'s original NSA uses a single global affinity threshold, r, in conjunction with the RCBITS matching rule for each individual ALC [33] within the population of ALCs, C. The affinity threshold is determined through a process of trial and error, whereby the threshold yielding the best system performance is chosen as the target affinity threshold. A general framework to aid in choosing an optimum value for r in conjunction with the RCBITS rule was provided by Timmis *et al.* [4], as follows:

- Create a self training test, $D_{TRAIN},$ and a self test set, $D_{TEST},$ from a set of self-strings S such that $S = D_{TRAIN} \cup D_{TEST}.$
- Generate the required number of detectors, n_r (the exact value for n_r is determined mathematically: see section 4.2), for each possible value of $r = 1, r = 2, \ldots, r = n,$ where n is the dimensionality of each ALC. Take note that n_r signifies the total number of random detectors that would need to be generated by the NSA in order to create n_c self-tolerant detectors.
- Run the NSA using D_{TRAIN}, and test the resultant population of generated ALCs, C, using D_{TEST} to obtain the values for FP, FN, TP and TN.
- Once a test has been executed for each individual value of r, use the value for r that yielded the highest TP and TN rate whilst maintaining an acceptable FP and FN rate.

An equally viable alternative to negative selection is positive selection, which can be viewed as the inverse of negative selection. In positive selection, detectors that are not self-reactive are eliminated. A minimal amount of work has been conducted on the merits of positive selection as opposed to negative selection. A formal framework to address this issue has been developed by Esponda *et al.* [29]. The framework analyses the tradeoffs between positive and negative selection with regard to the number of detectors needed to achieve a certain coverage of a problem space V. In comparing the two detection schemes within a particular problem domain, the framework considers:

- The matching rules used by the algorithms (refer to section 4.2).
- The generalisation properties induced by negative and positive selection algorithms (refer to section 3.5.2).

- The redundancy properties induced by both negative and positive selection algorithms. In this thesis, the terms redundancy and overfitting mean the same thing and can be used interchangeably (refer to section 3.5.3).
- Diversity (the variance between detectors) within positive and negative selection algorithms (refer to section 3.4.2.7).

The work in this thesis concentrates on negative selection, since negative selection employs similar mechanisms to positive selection, and examines and builds on each facet of the framework created by Esponda *et al* [29].

4.2 Effect of Matching Functions

One of the notable advantages of the NSA over many of the other AIS algorithms is that, besides being theoretically simple, the NSA allows any matching function to be employed (although this statement is true for a large majority of AIS algorithms it is not true for all AIS algorithms, as discussed in section 4.3). Different matching functions, however, induce different detection regions for each ALC, $\mathbf{x}_i \in C$, and thus have a direct influence on the performance of the NSA (when the performance of the NSA is assessed using the metrics defined in section 3.5).

This section discusses two of the most prominent analyses performed by several AIS researchers on how matching functions influence the performance of the NSA. The original analysis performed by Forrest *et al.* [33], which is rooted in probability theory, is discussed in section 4.2.1. An alternative analysis performed González *et al.* [38], which employs a simple technique to visualize the shape space generated by the NSA utilising a particular matching rule, is then discussed in section 4.2.2.

4.2.1 Analysis of the NSA and RCBITS Rule performed by Forrest *et al.*

When the NSA was first introduced by Forrest *et al.* [33], the RCBITS rule was utilised due to its simplicity and the ease at which it lends itself to mathematical analysis. Noting that the NSA is probabilistic, Forrest *et al.* [33] derived five equations that can be used to determine how many self tolerant detectors, n_c, need to be generated by the NSA in order to protect a set of self strings, S, with a certain failure probability, P_f. Each of these equations are presented below (see equations 4.3 to 4.7). The equations derived by Forrest *et al.* are based on the probability, P_M , that two random strings match in at least r positions, which was defined by Percus *et al.* [67, 68] as:

$$P_M \approx {n_{alph}}^{-r}\left[\frac{(n-r)(n_{alph}-1)}{n_{alph}+1}\right] \tag{4.1}$$

where n_{alph} is the number of symbols contained within the alphabet of the strings (for example for a binary string n_{alph} is 2) and n is the length of a string. It should be noted that this approximation is only good if $n_{alph} \ll 1$. If this constraint is not satisfied, then the exact equation must be used [45, 72]. Equation (4.1) exhibits two characteristics: there is a linear increase in P_M as n increases, and there is an exponential decrease in P_M as r increases.

Similar to equation (4.1), Wierzchoń [87] derived an equation to calculate the probability that two random binary strings have a hamming distance of r i.e. this equation can be used if the HD rule is employed with the NSA:

$$P_M = 2^{-n}\sum_{i=r}^{n}\binom{n}{i} \tag{4.2}$$

It should be noted that equation (4.2) is presented merely for purposes of completeness and that equations (4.3) to (4.7) assume that P_M is calculated using equation (4.1).

The probability of a random string/ALC not matching any self-strings within S is given by Forrest *et al.* [33] as:

$$P_S = (1-P_M)^{|S|} \tag{4.3}$$

The probability that n_c self-tolerant detectors fail to detect an antigen is given by Forrest *et al.* [33] as:

$$P_f = (1-P_M)^{n_c} \tag{4.4}$$

where n_c is equal to the number of self-tolerant ALCs to train.

If P_M is small and n_c is large, then,

$$P_f \approx e^{-P_M|S|} \tag{4.5}$$

The number of self-tolerant detectors, n_c, needed to attain a certain failure probability, P_f, and matching probability, P_M, is given by Forrest *et al.* [33] as :

$$n_c = \frac{-\ln\left(P_f\right)}{P_M} \tag{4.6}$$

The number of initial ALCs, n_r, before censoring (i.e. applying the NSA to the detectors to verify whether any of them are not self-tolerant), needed to generate n_c detectors is given by Forrest *et al.* [33] as:

$$n_r = \frac{-\ln\left(P_f\right)}{P_M(1-P_M)^{|S|}} \tag{4.7}$$

A number of reasons are given by Forrest *et al.* [33] to illustrate why the NSA is desirable, namely:

- The performance of the algorithm is tuneable in the sense that a desired P_f can be selected and both n_r and n_c can be determined as a function of $|S|$. The implication of this is that it is possible to determine the exact number of detectors, n_c, needed to protect a particular set of self-strings, S, without having to guess suitable values for these parameters through a trial and error process. Furthermore, the maximum number of random strings, n_r, which need to be generated in order to create n_c self-tolerant detectors can also be determined upfront.
- The size of the detector set does not grow if both P_M and P_f are fixed. The implication of this is that a set of n_c self-tolerant detectors only needs to be generated once if the problem space remains constant (i.e. the definition of self and non-self does not change).
- There is an exponential increase in the detection probability as the number of independent NSAs increases.

- Detection is symmetric, implying that the same amount of protection afforded to self by the detector set is afforded to the detector set by self, because changes to both self and the detector set utilise an identical matching function.

One of the major disadvantages of the NSA is that if $P_{M,} P_{f,}$ and n_c are fixed, then an exponential increase in n_r can be observed. Forrest *et al.* state that this can also be viewed in a positive light, in the sense that if such a set of n_c detectors were generated by a supercomputer, then it is highly unlikely that a change to self would go undetected [33].

Interestingly, based on a study of the suitability of the NSA for network intrusion detection, Kim and Bentley [55] cited this factor as a primary reason for the NSA having failed to perform effectively. Due to the NSA suffering from a severe scaling problem Kim and Bently concluded that the NSA should rather be used as a filter for invalid detectors and not for the generation of detectors.

Although the approach provided by Forrest *et al.* [33] works well in predicting the behaviour of the NSA with the RCBITS rule, it is not a trivial matter to define mathematical equations to predict the P_M, P_f, n_c and n_r associated with a particular matching function employed within the NSA. González *et al.* [38] defined a much simpler methodology to study the efficacy of various detection rules within the context of the NSA, as discussed in section 4.2.2.

4.2.2 Visualisation of the Shape Space Generated by a Matching Rule

The method proposed by González *et al.* attempts to visualise the shape space defined by Perelson and Oster [69]. In terms of Perelson and Oster's shape-space theory, ALCs should be distributed throughout the entire shape space such that the detection region of each ALC is able to detect a number of structurally similar antigen [69]. The shape space, however, is rarely two-dimensional and, consequently, the process of distributing ALCs throughout the shape space is not a trivial process. If, however, a mechanism existed to map an n-dimensional problem space to 2 dimensions it would be much easier to distribute the ALCs in a more effective manner.

There are fortunately a number of algorithms that can be used to map multidimensional data to a lower dimensionality [63, 74, 83]. For example, Sammon's mapping attempts to preserve inter-pattern distances by minimising an error criterion that differentiates between distances amid points in the original data set and distances amid points in the new data set. By preserving the distance between points in the original data set and the distance between points in the new data set, the algorithm preserves the topology/dynamics of the original shape space [75]. With regard to AIS algorithms, Sammon's mapping holds the following two disadvantages:

- Sammon's algorithm is computationally expensive.
- The resultant mapping rendered by Sammon's mapping will be intuitive only for distance-based detection rules, such as the HD rule.

González *et al.* [38] developed a much more elegant and computationally inexpensive way to visualise Perelson and Oster's shape space, as defined below:

Any point (x, y) taken from a problem space corresponding to the domain $[0.0,1.0]^2$ can be mapped to a binary string, $b_0,b_1...b_7,b_8,b_9..b_{15,}$ of length 16 where the first eight bits encode the integer value $[255.x + 0.5]$ and the last eight bits encode the integer value $[255.y + 0.5]$

As an example of how the mapping can be applied, González *et al.* [38] considered a single detector, 1000000010000000, and generated the areas covered by the detector using:

- The RCBITS rule, with $r = 4$ (leftmost image in Figure 18).
- The RCHK rule, with $\mathbf{x}_i = 0011$ and $w = 7$ (central image in Figure 18).
- The HD rule, with $r = 8$ (rightmost image in Figure 18).

The grey areas in Figure 18 represent areas covered by the detector, 1000000010000000, similarly the white areas in Figure 18 represent areas not covered by the detector. The mapping defined by González *et al.* [38] can thus be used to graphically view the amount of the shape space covered by one or more detectors employing a particular detection rule.

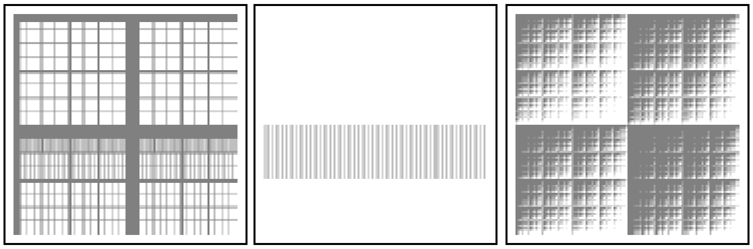


Figure 18. *Areas covered by a detector 1000000010000000 (this image was taken from [75])*

An interesting point noted by González *et al.* [38] was that the relation between the detector and the proximity of the detector's related detection region was not congruent with the natural proximity relation in a real-valued two-dimensional space, making it difficult to achieve an optimal distribution of a set of detectors. By using the mapping to study the results of binary matching rules with regard to different training sets, González *et al.* [38] drew the following conclusions:

- The binary matching rules studied, namely, RCBITS, RCHK, and HD, cannot produce a good generalisation of the self-space, resulting in poor coverage of the non-self space. The reason that the binary matching rules do not produce a good generalisation of the self-space is that they are not able to accurately capture the affinity relation employed in the real space within the self/non-self (binary space).
- The matching rule used by NSA needs to be chosen in such a way that the affinity relationship between points in the problem space is preserved when the relationship is transposed to the self/non-self space.

This section discussed how the effect of matching functions on performance of the NSA, introduced by Forrest *et al.* [33], can be studied both mathematically (section 4.2.1) and visually (section 4.2.2). The next section discusses several hybrid detector-generating techniques introduced by AIS researchers to overcome the severe scaling problem exhibited by the NSA.

4.3 Alternative Detector-generating Techniques

A number of issues inherent to the original NSA were noted by Timmis *et al.* [4], namely

- It is time consuming to generate a large number of candidate detectors and examine each individual candidate detector to determine whether or not it is self reactive (activated by a self string) and whether or not the detector can be added to the repertoire of detectors, C.
- The number of detectors required to afford a relatively satisfactory protection level increases exponentially as the size of the self-set increases if the failure probability is fixed. This sentiment has been echoed by Kim and Bentley [55]. This problem is exacerbated when the length of the binary detectors increases.
- No process is incorporated to ensure that there is no redundancy. This has a direct consequence on the overfitting and generalisation behaviour of C, especially because of n_c being fixed.

Several variations of the NSA have come into existence in an attempt to minimise some of the issues mentioned above. Each of the alternative NSAs mentioned below have a major disadvantage, being that the only detection rule that can be employed by these algorithms is the RCBITS detection rule. This section discusses some of the most popular variations of the NSA starting with the linear time detector-generating algorithm in section 4.3.1, followed by the greedy detector-generating algorithm in section 4.3.2, an algorithm based on the discriminative power of a detector in section 4.3.3, and the NSMutate algorithm in section 4.4.

4.3.1 Linear Time Detector-generating Algorithm

The linear time detector-generating algorithm developed by D’haeseleer *et al.* [11] obtained its name from the fact the algorithm executes in linear time with respect to the size of the input, $|S|$, where n, the length of an ALC, and r, the affinity threshold, are constant. D’haeseleer *et al.* [11] used the following terminology within the algorithm.

The algorithm consists of two distinct phases. Phase 1 is concerned with solving a counting recurrence, whereas phase 2 generates the actual detector strings.

Let s a binary string of length n, for example, 10111.

Let $\hat{s}$ denote a string without its leftmost bit, for the above example, $\hat{s} = 0111$.

Let $\check{s}$ denote a string without its rightmost bit, for the above example, $\check{s} = 1011$.

Let $s.b$, where $b \in \{0,1\}$ denote a binary string with b appended to the end of the string. For the above example, if $b = 0$, then $s.b = 101110$.

Let $b.s$, where $b \in \{0,1\}$ denotes a binary string with b appended to the beginning of the string. For the above example, if $b = 0$, then $b.s = 010111$.

A template of order r is a string of length n consisting of $n - r$ blank symbols. A template is specified by using the symbol, $t_{i,w}$, where w denotes a string composed of r bits and i denotes the starting position of the w bits.

A right completion is a string s, such that all the right blanks are replaced by valid bits.

A left completion is a string s, such that all the left blanks are replaced by valid bits.

Let S_r be a set of all possible binary strings of length r. S_r can easily be calculated by constructing a truth table of all binary strings of length r.

Let $\mathbf{M}_i[s]$ be the number of right completions of $t_{i,w}$ unmatched by any string within the self-set S where the rows of $\mathbf{M}$ are equal to all of possible binary strings of length r, and the columns of $\mathbf{M}$ are equal to all possible positions of $t_{i,w}$; $n - r + 1$ in total.

Figure 19. *Terminology used by the linear time detector-generating algorithm*

Phase 1: Counting a recurrence relation

Pseudocode for phase 1 is given in Figure 20. Figure 21 illustrates how $\mathbf{M}$ is created and initialised using phase 1 of the linear time detector-generating algorithm on a finite set, S, of self strings. Figure 21 shows two tables: the table on the left, S, is comprised of the entire self-set of strings and the table on the right represents the matrix, $\mathbf{M}_i[s]$.

Create a matrix, **M**, such that the rows of the matrix are equal to all possible bit patterns of size r and the columns of the matrix are set to all of the valid window positions for the bit patterns, $n - r + 1$ in total;

The matrix is initialised as follows:

```
  for each template $t_{i,w} \in S_r$ do
      for $i = n - r + 1$ to 1 do
          if $\exists\, s \in S: t_{i,w}$ matches $s$ then
              $\mathbf{M}_i[s] := 0$;
          else
              $\mathbf{M}_i[s] := 1$;
      end
    for each string $t_{i,w}$ where $1 \le i \le (n - r + 1)$ do
        if $\exists\, s \in S: t_{i,w}$ matches $s$ then
            $\mathbf{M}_i[s] := 0$;
        else
          Update $\mathbf{M}_i[s]$ using the following recurrence relation: $\mathbf{M}_{i+1}[\hat{s}.0] + \mathbf{M}_{i+1}[\hat{s}.1]$;
        end
    end
  end
end
```

Figure 20. *Pseudocode for phase 1 of the linear time detector-generating algorithm*

In Figure 21, $r = 3$, and the S_r column was generated by constructing a truth table for all binary strings of length 3. Each row of the S_r column corresponds to a window, w, or r

bits. Each row within a particular column $t_{i,w}$ of the $\mathbf{M}_i[s]$ table represents the number of right completions $t_{i,w}$ unmatched by any string within the self-set, S.

S
001110
001101
001111
010001
010101
011100
011111
100001
110001
110100

S_r	$t_{1,w}$	$t_{2,w}$	$t_{3,w}$	$t_{4,w}$
000	0	0	0	1
001	0	0	2	0
010	0	3	0	1
011	0	0	0	1
100	0	0	1	0
101	3	0	2	0
110	0	3	0	0
111	3	0	0	0

Figure 21. *Application of phase 1 of the linear time detector-generating algorithm*

Phase 2: Generation of strings unmatched by S

Once the recursive process has been completed for $\mathbf{M}_{n-r+1}[.]$ to $\mathbf{M}_1[.]$, each cell within $\mathbf{M}$ denotes the number of unmatched bit strings starting with the r bit pattern. It should be noted that "." is merely a placeholder representing any string $s \in S$. The matrix $\mathbf{M}_1[.]$ is in effect a partitioning of the space of unmatched strings in $\mathbf{M}_1[.]$ for each initial string beginning with the bit pattern specified by ".".

Each subsequent column from $\mathbf{M}_2[.]$ to $\mathbf{M}_{n-r+1}[.]$ can then be viewed as a further partitioning of the space. Following this reasoning the total number of detector strings that are unmatched by a particular bit string s is given by

$$T = \sum_{s \in S_r} \mathbf{M_1}[s] \tag{4.9}$$

Pseudocode for phase 2 of the linear time detector-generating algorithm is given in Figure 22.

```
A random detector, $\mathbf{x}_k$, can be created as follows:

Construct $S_r$ so that $S_r$ is equal to all possible binary strings of length $r$;

Calculate $T = \sum_{s \in S_r} \mathbf{M}_1[s]$;

Randomly select a number, $k \in \{1, \dots, T\}$;

Find a string $s_1$ such that $P_1 = \sum_{s < s_1} \mathbf{M}_1[s] < k \leq Q1 = \sum_{s \leq s_1} \mathbf{M}_1[s]$;

Set the first $r$ bits of $\mathbf{x}_k$ equal to $s_1$;

for $i = 1$ to $n - r + 1$ do
   if $k$ falls in the partition $(P_i, P_i + \mathbf{M}_{i+1}[\hat{s}_i.0]]$ then
       $b := '0'$;
      $\mathbf{x}_k := \mathbf{x}_k.b$;
    else if $k$ falls in the partition $(P_i + \mathbf{M}_{i+1}[\hat{s}_i.0], Q_i]$, then
      $b := '1'$;
      $\mathbf{x}_k := \mathbf{x}_k.b$;
   end
  if $b = 0$ then
    $P_{i+1} := P_i$;
    $Q_{i+1} := Q_i + \mathbf{M}_{i+1}[\hat{s}_i.0]$;
 end
     else if $b = 1$ then
         $P_{i+1} := P_i + \mathbf{M}_{i+1}[\hat{s}_i.0]$;
       $Q_{i+1} := Q_i$;
   end
end
```

Figure 22. *Pseudocode for phase 2 of the linear time detector-generating algorithm*

To illustrate how a detector is generated by phase 2, consider Figure 23 (note that Figure 23 was generated using the information from Figure 21). The following notation is used in Figure 23 to represent a window: <*window*> (<*number of self-binary strings s matched by the window*>,{<*implicit detector numbers detected by the window*>}, where "<" and ">" are placeholders.

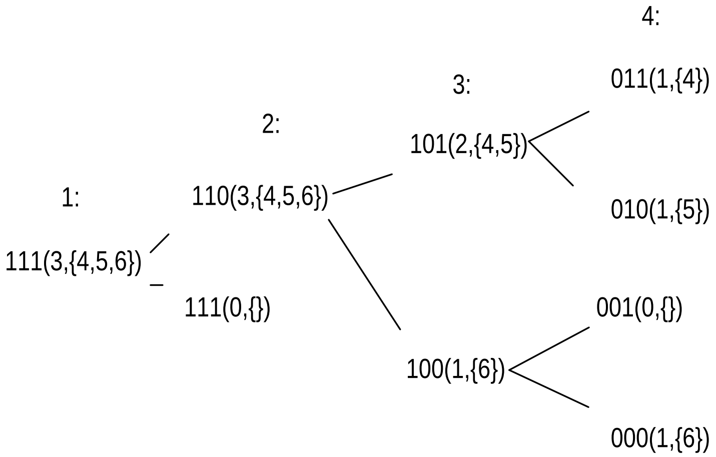


Figure 23. *Implicit portioning used by phase 2 of the linear time detector-generating algorithm*

Detector 4, in Figure 23, can be generated by tracing a route from the root of the tree to the leaf that detects detector 4. The resultant detector is created by constructing a string comprising of the first r bits of the root, followed by the last bit of each node visited along the path traced from the root node to the leaf node, which detects detector 4. That is, detector 4, 111011, is constructed by taking 111, the first r bits of the root, and the last bit from nodes: 2, 3 and 5 respectively.

4.3.2 Greedy Detector-generating Algorithm

The greedy detector-generator algorithm of D'haeseleer *et al.* [11] aims to spread detectors as far apart as possible to achieve maximal coverage of the problem space. The greedy detector-generating algorithm is similar to the linear time detector-generating algorithm in that it consists of two phases. The greedy detector-generating algorithm creates two matrices, $\mathbf{D_S}$ and $\mathbf{D_C}$, based on a set of self-strings, S, and a set of detectors, C, respectively. The process followed to create a **D** matrix is given in Figure 24.

```
Create a matrix $\mathbf{M}$ such that the rows of the matrix are equal to all possible bit patterns of size $r$ and the columns of the matrix are set to all of the valid window positions for the bit patterns, $n - r + 1$ in total;
Create a matrix $\mathbf{M}'$ equal to $\mathbf{M}$ (the $\mathbf{M}'$ represents the number of non-matching left completions for a template $t_{i,s}$);
Populate $\mathbf{M}$ using the algorithm in Figure 20;
Populate $\mathbf{M}'$ using the algorithm in Figure 20 (Note that the recurrence relation:
$\mathbf{M}_{i+1}[\hat{s}.0] + \mathbf{M}_{i+1}[\hat{s}.1]$ in Figure 20 must be replaced with: $\mathbf{M}_{i-1}[0.\check{s}] + \mathbf{M}_{i-1}[1.\check{s}]$
Create an empty matrix $\mathbf{D}$ structurally similar to $\mathbf{M}$ (each entry, $\mathbf{D}_i[s]$, represents the number of unmatched fully specified bit strings corresponding to the template represented by $i$);
Populate $\mathbf{D}$ using $\mathbf{D}_i[s] = \mathbf{M}_i[s] \times \mathbf{M}'_i[s]$ ;
```

Figure 24. *Pseudocode for generating a D matrix*

Pseudocode for phase 1 of the greedy detector-generating algorithm is given in Figure 26.

```
Construct an empty matrix $\mathbf{D_S}$ based on $S$ (the $\mathbf{M}$ and $\mathbf{M}'$ matrices in the algorithm in Figure 24 are constructed by using $S$);
Populate $\mathbf{D_S}$ using the algorithm in Figure 24;
Construct an empty matrix $\mathbf{D_C}$ based on $C$ (the $\mathbf{M}$ and $\mathbf{M}'$ matrices in the algorithm in Figure 24 are constructed by using $C$);
Populate $\mathbf{D_C}$ using the algorithm in Figure 24;
```

Figure 25. *Pseudocode for phase 1 of the greedy detector-generating algorithm*

Once the **D** matrices have been created, detectors are generated according to the algorithm given in Figure 26 to achieve maximal coverage.

```
for each detector to be generated do
begin
    Select the largest entry in $\mathbf{D_C}$ ( if there is a tie between entries, an entry is selected at random);
    Starting at this entry/template, transverse the row, $\mathbf{D_C}$, corresponding to the selected entry, from both the left and the right; 0 or 1 is  added to the template each time depending on which template contains the highest number of unmatched strings (note that a template may be selected only if it is a valid detector template, that is, if it contains a 0 entry in the array$\mathbf{D_S}$);
    Add the generated detector to $C$;
    Update matrix $\mathbf{D_C}$ by:
        Setting the entries that match the newly generated detector in the $\mathbf{M}$ and $\mathbf{M}'$ matrices used to construct $\mathbf{D_C}$ to 0.
        Re-generate $\mathbf{D_C}$ by executing the portion of the algorithm given in Figure 24 that generates  the $\mathbf{D_C}$ matrix;
end
```

Figure 26. *Pseudocode for phase 2 of the greedy detector-generating algorithm*

The linear time detector-generating algorithm and the greedy detector-generating algorithm were studied by D'haeseleer *et al.* [11] under a number of different conditions, with the following conclusions being drawn:

- Large alphabet sizes, m, make it increasingly difficult to choose an optimal value for the affinity threshold, r.
- Large values of n and r increase the computational complexity of both the linear time detector-generating algorithm and the greedy detector-generation algorithm.
- To minimise the number of holes produced by both algorithms, r must be chosen such that $|S| \leq 2^r$.

- A lower bound on the number of detectors that need to be generated by the linear time detector-generating algorithm is given by equation (4.6). Interestingly, the lower bound defined for the number of detectors needed by the greedy detector-generating algorithm is defined by $N_C \geq \frac{1-P_f}{P_M}$. An in-depth theoretical analysis of the linear time detector-generating algorithm and the greedy detector-generating algorithm can be found in [12].

4.3.3 Discriminative Power of a Detector under the RCBITS Rule

Wierzchoń [87] defines the discriminative power of a detector as the number of unique strings detected by a detector using the RCBITS rule. The discriminative power of a detector can be found by counting the number of unique strings recognised by each, $t_{i,w}$, induced by a detector, **x**. As an example, consider the detector 001101 and let $r = 3$.The detector induces the following templates:

$t_{1,001} = 001 ***$, $t_{2,011} =* 011 **$, $t_{3,110} =** 110 *$, and $t_{4,101} =*** 101$.The first template, $t_{1,001} = 001 ***$, recognises 2^{n-r} unique strings. The second template, $t_{2,011} =* 011 **$, matches strings $s_1 = 0011 **$ and $s_2 = 1011 **$. However, s_1 is also recognised by $t_{1,001}$. Hence the total number of strings recognised by $t_{2,011}$ is halved.

Following this reasoning, Wierzchoń [87] found that the discriminative power of a receptor is equal to:

$$2^{n-r-1}.(2+n-r) \tag{4.10}$$

Two additional characteristics affecting the discriminative power of a detector are the number of holes induced by the RCBITS rule and the number of detectors that cannot be produced because they would be self-reactive. Wierzchoń created two algorithms to calculate each measure. These are discussed in sections 4.3.3.1 and 4.3.3.2 respectively.

4.3.3.1 Counting the Number of Holes in a Self-set

The RCBITS rule induces a number of holes (see section 3.4.2.4). Wierzchoń [87] created an algorithm to count the maximum number of holes that can exist within a self-set, S, by dividing S into two subsets:

- S_{rself}: *A* set of all windows/templates matching at least one self-string, $s \in S$.
- $S_{rnonself}$: A set of all windows or templates that do not match a single string, $s \in S$. Note that candidate detectors are constructed from $S_{rnonself}$.

Using S_{rself} a graph, G, is constructed using the algorithm in Figure 27.

```
Create an empty graph $G$;
for each string $s \in S_{rself}$ do
    Add each distinct template, $t_{1,w}$, induced by $s$ to level 1 of $G$;
end
for $i = 1$ to $(n - r)$ do
    for every $t_{i,w}$ at level $i$ do
        Create a left child for $t_{i,w}$ as $t_{i+1,\hat{w}.0}$;
        if $t_{i+1,\hat{w}.0} \notin G$ then
            Add $t_{i+1,\hat{w}.0}$ to $G$;
        end
        Create a right child for $t_{i,w}$ as $t_{i+1,\hat{w}.1}$;
        if $t_{i+1,\hat{w}.1} \notin G$ then
            Add $t_{i+1,\hat{w}.1}$ to $G$;
        end
        Create an edge between $t_{i,w}$ and $t_{i+1,\hat{w}.0}$;
        Create an edge between $t_{i,w}$and $t_{i+1,\hat{w}.1}$;
    end
end
```

Figure 27. *Pseudocode for creating a graph of all possible templates induced by a self-set S*

Figure 28 shows an example of the algorithm presented in Figure 27 applied to the binary string 001101.

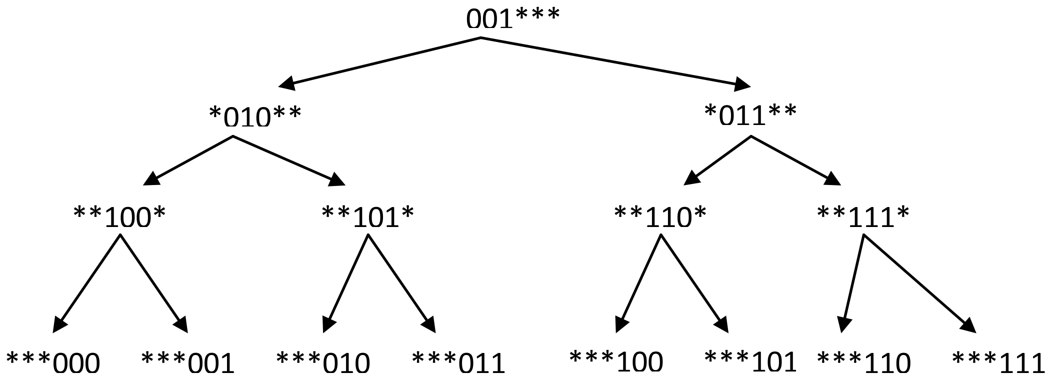


Figure 28. *A binary tree constructed for 001101*

The number of self-strings induced by S_{rself} can be found by counting the number of distinct paths from each root in G to each leaf. The number of holes in S is then equal to $|S|$, i.e. the number of strings induced by S_{rself}.

4.3.3.2 “FindIneffective” Procedure

Owing to the distribution of self-strings within the problem space, a situation can arise in which a valid detector cannot be constructed from a root template chosen from $S_{rnonself}$, because all the leafs of the template belong to self. To illustrate this, consider the set of self-strings,

$$S = \{001110,001101,001111,010001,010101,011100,011111,100001,110100\}.$$

Using the algorithm in Figure 27 for the binary template, $t_{1,1000} \in S_{rnonself}$, to construct a tree, Figure 29 shows paths that will generate valid detectors (using solid edges) and invalid paths (using dashed lines).

An algorithm, called the FindIneffective algorithm, which uses $S_{rnonself}$ to determine which binary templates $t_{1,w} \in S_{rnonself}$ cannot be used to generate valid detectors was developed Wierzchoń [87]. The FindIneffective procedure is outlined in Figure 30.

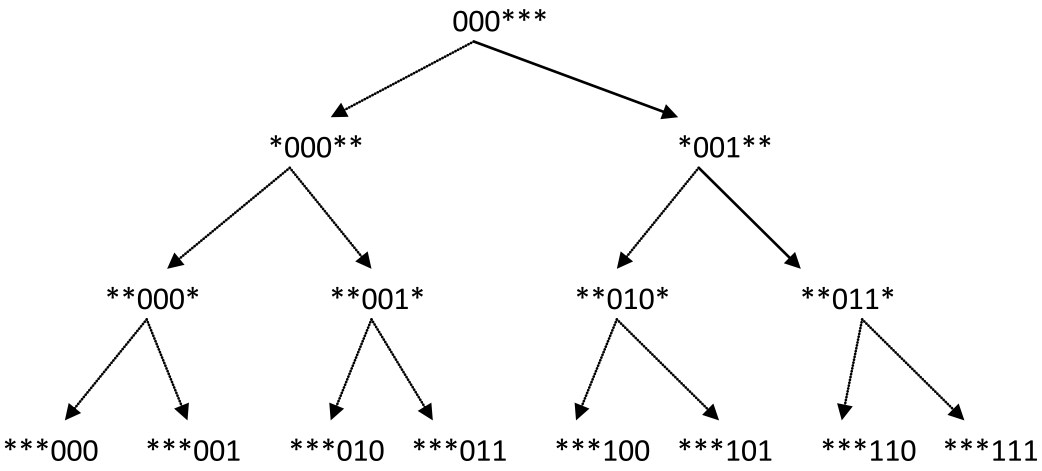


Figure 29. *Binary tree constructed for the non-self template $t_{1,000}$*

```
for each $t_{i,w} \in S_{rself}$ do
  Check the parent $t_{i-1,v}$ of $t_{i,w}$;
  if the parent $t_{i-1,v} \notin S_{rself}$, but its children are members of $S_{rself}$ then
      add $t_{i-1,v}$ to $S_{rself}$;
      remove $t_{i-1,v}$ from $S_{rnonself}$;
  end
  Check the children $t_{i+1,v}$ of $t_{i,w}$;
  if a child $t_{i+1,v} \notin S_{rself}$, but both its parents are $\in S_{rself}$ then
      add $t_{i+1,v}$ to $S_{rself}$;
      remove $t_{i+1,v}$ from $S_{rnonself}$;
  end
end
```

Figure 30. *FindIneffective procedure*

Wierzchoń's methods [88] can be combined to generate a set of detectors which maximises the discriminative power of a receptor.

4.4 Negative Selection with Mutation Algorithm

The objective of the negative selection with mutation algorithm (NSMutate) proposed by De Castro and Timmis [23], is to increase the speed at which the NSA converges on a set of detectors through the introduction of a mutation step (recall from section 4.2.1 that if P_M , P_f , and n_c are fixed, then there is an exponential increase in n_r). The algorithm mutates self-reactive detectors for several iterations until they are no longer activated by any $s \in S$. Each candidate detector has a detector lifetime indicator determining the number of mutation attempts that can be made before the detector is finally discarded. Pseudocode for the algorithm is given in Figure 31.

The mutation is adaptive and is proportional to the affinity between the candidate detector and the matching self-element. The higher the affinity (between the detector and the self-element) is, the more the detector is mutated and, similarly, the lower the affinity is, the less the detector is mutated. This allows the detector to take larger or smaller jumps within the search space, depending on the fitness of the detector. The mutation probability can be calculated by using

$$P_{mutate} = 1 - \frac{f_{MATCH}(\mathbf{x}, \mathbf{z})}{f_{MAX}(\mathbf{x})} \tag{4.11}$$

where $\mathbf{x}$ is a detector, $\mathbf{z}$ is a self string/vector, $f_{MAX}(\mathbf{x})$is the maximum affinity of $\mathbf{x}$, and $f_{MATCH}(\mathbf{x}, \mathbf{z})$ represents any matching function (for example the RCBITS rule). The mutation step can be performed using a random mutation algorithm as summarised in Figure 32 or an inorder mutation algorithm as summarised in Figure 33.

```
Let counter $n_c$ be the number of self-tolerant artificial lymphocytes to train;
Let $C$ be an empty set of self-tolerant ALCs;
Let mutationSteps be the number of times that a self-tolerant detector is mutated;
Create a training set $D_{TRAIN}$ consisting of self-patterns;
while $|C| \neq n_c$ do
    Randomly generate an ALC, $\mathbf{x}_i$;
    matched := false;
      for each $p \in \{1, \ldots, |D_{TRAIN}|\}$ do
          if $f_{MATCH}(\mathbf{x}_i, \mathbf{z}_p)$ is greater than the affinity threshold $r$ then
           matched:= true;
              for $i = 1$ to mutationSteps do
                  Mutate $\mathbf{x}_i$;
                  if $\forall\, \mathbf{z}_j \in \{1, \ldots, p\}: f_{MATCH}(\mathbf{x}_i, \mathbf{z}_j) < r$ then
                   break;
              end
          end
      end
     if matched = false then
        Add $\mathbf{x}_i$ to $C$;
     end
  end
```

Figure 31. *Pseudocode for the NSMutate algorithm*

```
Let $\mathbf{x}_{ij}$ be the $j^{th}$ element of a detector $\mathbf{x}_i = (x_1, x_2, \ldots, x_n)$;
Let $U(0,1)$ be a uniform random number between 0 and 1;
for $j = 1$ to $n$ do
   if $U(0,1) \leq P_{mutate}$ then
      $\mathbf{x}_{ij} := !\mathbf{x}_{ij}$, where ! denotes the Boolean NOT operator;
   end
end
```

Figure 32. *Random mutation algorithm for a binary string*

```
Select mutation points, $\mathcal{E}_1, \mathcal{E}_2 \sim U(1, \dots, n)$;
 for $j = \mathcal{E}_1$ to $\mathcal{E}_2$ do
     if $U(0,1) \leq P_{mutate}$ then
         $\mathbf{x}_{ij} \coloneqq !\mathbf{x}_{ij}$;
   end
end
```

Figure 33. *Inorder mutation algorithm of a binary string*

The fact that the time complexity of the NSMutate algorithm degrades as P_{mutate} becomes larger was noted by De Castro *et al.* [23]. An additional restriction is also imposed on the algorithm by using a binary alphabet, i.e. if $P_{mutate} = 1.0$ then each mutation performed on a particular detector will merely flip the detector to its inverse and back. It was also found by De Castro *et al.* [23] that the time complexity of the algorithm increased as a detector's life-time indicator increases. The benefit of the NSMutate algorithm is that it reduces the number of candidate detectors generated when it is computationally more expensive to generate a detector than to mutate a detector.

In their analysis of the NSMutate algorithm, Ayara *et al.* [4] came to the following conclusions:

- The number of candidate detectors increases exponentially as the size of the self-set increases.
- If the self-set is randomly distributed within the search space, there is an equal probability of the random detector mutating either towards or away from the self-set. This results in the performance of the NSMutate algorithm being equivalent to the original NSA.
- The NSMutate algorithm is more tuneable than the NSA, and good performance can be obtained by tuning the algorithm's parameters for a specific data set.

The algorithms discussed thus far include the original NSA by Forrest *et al.* and some of its variants.

Although these algorithms are quite different from each other, the problem space in which they reside is the same; that is, binary representations are used for both self and non-self strings. The next section briefly discusses the real-valued negative selection algorithm.

4.5 Real-valued Negative Selection Algorithm

The original NSA was developed for binary spaces and performs very well if the problem space is categorised. However, the NSA does not perform well when a real-valued problem is mapped to a binary space. The primary reason for this deficiency is that the relation between the detector and the proximity of the detector's detection region is not congruent with the natural proximity relation in the real-valued space [75]. The real-valued NSA was informally proposed by Ebner *et al.* [25], but Gonzalez *et al.* were the first to apply the real-valued NSA [80, 82].

Each detector is represented as a hyper-sphere in an n-dimensional real-valued space, where the centre of the hyper-sphere is indicated by the coordinates of $\mathbf{x}$ and the radius of the hyper-sphere is indicated by r, the affinity threshold. The affinity-matching function between a detector $\mathbf{x}$ and an antigen $\mathbf{y}$ is represented by the Euclidean distance, where $\mathbf{x}$ matches $\mathbf{y}$ if the Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$ is less than r.

The algorithm generates detectors in the same manner as the original NSA presented in Figure 17, except that each detector is randomly generated within $[1{,}0]^n$ and $f_{EUCLIDEAN}(\mathbf{w}, \mathbf{m})$ is used to determine the Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$. An alternative real-valued NSA, the *V*-detector algorithm, which generates variable size detectors, was developed by Ji and Dasgupta [53]. For a detector $\mathbf{x} \in C$, the radius r of $\mathbf{x}$ is decided by the Euclidean distance to the closest self-sample by the *V*-detector algorithm.

Pseudocode for the V-detector algorithm is given in Figure 34.

Ji and Dasgupta [53] showed that if the number of detectors, n_c, remained constant, then the V-detector algorithm has the same complexity as the real-valued NSA. The experiments performed by Ji and Dasgupta [53] concluded that the V-detector algorithm has a much better DR and FR rate (see section 3.5.1) than the real-valued NSA. The following advantages provided by the V-detector algorithm over the real-valued NSA were noted:

- The algorithm generates fewer detectors to cover a particular problem space.
- The algorithm is better at covering holes. That is, detectors with smaller affinity thresholds cover holes, whereas detectors with larger affinity thresholds cover large regions of the non-self space.
- The coverage estimate, $c_{0,}$ can be used to predict the algorithm's performance because it provides a means to explicitly specify the percentage of the problem space that must be protected/covered by the resultant set of generated detectors.

When Stibor *et al.* [81] compared the efficacy of the V-detector algorithm to statistical anomaly detection techniques (the Parzen-Window and one-class support vector machine techniques), they discovered a number of interesting facts about the termination conditions of the V-detector algorithm stated by Ji and Dasgupta [53]. With reference to the pseudocode of the V-detector algorithm in Figure 34, two termination conditions are reached when $k > \frac{1}{1-msc}$ or when $l \geq \frac{1}{1-c_0}$:

- The termination condition $k > \frac{1}{1-msc}$ is never satisfied, because k will never have a value greater than 1 which thus invalidates the condition as a termination condition.
- The probability of $l \geq \frac{1}{1-c_0}$ being satisfied is decreased when $|S|$ increases or when r increases.

```
Let $c_0$ be the estimated coverage (percentage of data points covered) of the problem space by a set of detectors $C$;
Let $msc$ be the maximum coverage (percentage of data points covered) of the problem space by a set of detectors $C$;
while $|C| \leq n_c$ do
    $k := 0$;
    $l := 0$;
    $r_{temp} := \infty$;
    Randomly generate an ALC, $\mathbf{x}_i \in [\mathbf{1}, \mathbf{0}]^{\boldsymbol{n}}$;
    for each $\mathbf{x}_j \in C$ do
        if $f_{EUCLIDEAN}(\mathbf{x}_i, \mathbf{x}_j) \leq$ the radius of $\mathbf{x}_j$ , $r$, then
           $l := l + 1$;
            if $l \geq \frac{1}{1-c_0}$ then
           return $C$;
       end
     end
  end
  for each self pattern, $\mathbf{z}_p \in S$ do
       if $f_{EUCLIDEAN}(\mathbf{x}, \mathbf{z}_p) - r_s \leq r$ then
          $r_{temp} := f_{EUCLIDEAN}(\mathbf{x}, \mathbf{z}_p) - r_s$;
  end
 if $r > r_s$ then
      Set the affinity threshold $r$ of $\mathbf{x}_i$ equal to $r_{temp}$;
      Add $\mathbf{x}_i$ to $C$;
 end
     else
        $k := k + 1$;
   end
    if $k > \frac{1}{1-msc}$ then
     exit the algorithm;
   end
end
```

Figure 34. *Pseudocode for V-detector algorithm*

4.6 Conclusion

The original NSA developed by Forrest *et al.* [33] was prolific in the sense that it inspired an acute interest in AIS and is one of the most popular AIS algorithms to date. The algorithm is conceptually simple and draws from the process of T-cell maturation in the thymus within the NIS.

The main advantages of the NSA are that self-data are sufficient to train the algorithm, and the algorithm does not impose a specific detection rule. One of the major pitfalls of the original algorithm (which used the RCBITS rule) is the severe scaling problem, that is, when $P_{M\,,}$ $P_{f\,,}$ and n_c are fixed, an exponential increase in n_r has been observed. This effectively translates to a scenario in which a large number of redundant detectors will need to be created in order to create n_c detectors that guarantee acceptable P_M and P_f values.

This chapter provided an in-depth overview of:

- The original NSA, including a mathematical analysis of the properties exhibited by the original NSA.
- A class of deterministic NSAs (the linear time detector-generating algorithm and the greedy detector-generating algorithm), which exploit the mathematical properties induced by the RCBITS rule within a binary shape space.
- Algorithms to examine the discriminative power of a receptor under the RCBITS rule operating within a binary shape space, in addition to counting the number of holes induced by the RCBITS rule.
- A version of the NSA that employs a mutation step in an attempt to mutate a self-reactive detector away from the self-space.
- A means with which to visualise the effect of binary matching rules within the NSA.
- A discussion of how randomly generated permutation masks affect the NSA.

- A discussion of real-valued NSA, as well as a variant of the real-valued NSA, called the V-detector algorithm.

The next chapter discusses a new affinity matching function, the feature-detection rule, which infers relationships between the constituents of an antigen to decide whether the antigen is matched by a particular detector. The thesis will show that when used as the affinity matching function of the original NSA, introduced by Forrest et *al.* [33], the feature-detection rule yields superior performance over the RCBITS, HD and RCHK affinity matching functions (see chapter 6).

Chapter 5
The Feature-Detection Rule

“If I have seen further it is by standing on the shoulders of giants.“

- Sir Isaac Newton

The feature-detection rule derives its name from the fact that it infers relationships between the constituents of an antigen to decide whether the antigen is matched by a detector.

The feature-detection rule is discussed in section 5.1. The mathematical properties associated with the feature-detection rule is then discussed in section 5.2. This is followed by a discussion of the interrelationship between the feature-detection rule and other affinity-matching functions discussed in this thesis (refer to section 3.4.2). This chapter concludes with a discussion on positional-bias introduced by the feature-detection rule and how it is addressed in this thesis.

5.1 Matching under the Feature-detection Rule

The feature-detection rule differs vastly from the RCHK, HD, and RCBITS rules in that it uses the interrelationships between antigen fragments to decide whether an antigen is detected by a candidate detector. To illustrate the characteristics of the feature-detection rule, consider the following fictitious study:

Suppose that an experimental study was carried out on a sample of individuals who had developed cancer and a sample of individuals who had not developed cancer in their lifetimes. The purpose of the study was to develop an algorithm that could deduce whether a person had indeed developed cancer by looking at an instance of the attributes/characteristics captured by the experimental study. Assume that the study interviewed each individual within the sample and captured the following data:

- Attribute 1: Does the individual smoke?
- Attribute 2: Does the individual work more than 60 hours per week?
- Attribute 3: Does the individual have a companion?
- Attribute 4: Does the individual drink alcohol more than six times per week?
- Attribute 5: Does the individual take frequent holidays?
- Attribute 6: Does the individual sleep at least eight hours per day?
- Attribute 7: Does the individual eat five portions of fruit and vegetables each day?
- Attribute 8: Does the individual exercise less than three times per week?

Some of the attributes are relevant to the problem that the algorithm is trying to solve, whereas others are irrelevant. It is also possible that a group of attributes and the value that each attribute carries are relevant to the outcome of the problem.

Consider Figure 35, which depicts such a group of attributes and their relative values (note that each attribute is a binary attribute). Figure 35 represents a particular individual having cancer if

- the individual smokes,
- the individual drinks alcohol more often than six times per week,
- the individual sleeps less than eight hours per day, and
- the individual does not exercise three times per week.

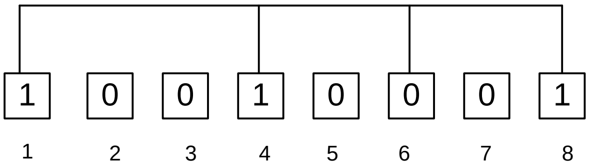


Figure 35. *Overview of the feature-detection rule*

The feature-detection rule refers to the characteristics described in the above example as features. From the above, the values of features 1, 4, 6 and 8 are relevant to the outcome of the problem, whereas the values of features 2, 3, 5 and 7 are irrelevant to the outcome of the problem. If an AIS algorithm were used to generate detectors to determine whether an individual has cancer, based on the values of the 8 attributes, it would in fact be more effective to:

- pre-process each self-string, comprised of all 8 features, into a shorter self-string comprising of features 1, 4, 6, and 8, and
- generate detectors, of length 4, against the pre-processed self-strings by employing the RCBITS rule.

The above example illustrates the premise upon which the feature-detection rule is based.

The feature-detection rule is applied by:

- pre-processing a string/vector into a shorter string/vector consisting of features relevant to the outcome of the problem (relevant features can be determined by a domain expert or by using mathematical techniques such as principle component analysis) currently under consideration (see the definition for f_{SELECT} below); and
- applying the RCBITS rule to the newly generated string/vector (consisting of relevant features) and a detector, **x** to determine whether the affinity between **x** and the newly generated string/vector is greater than r, the affinity threshold.

The definition of the feature-detection function, $f_{FEATURE}$, is formally stated below:

Let $f_{SELECT}(\mathbf{w}_1, \mathbf{p})$ be a function such that, given a binary vector $\mathbf{w}_1 = (w_1, w_2, \dots, w_n)$ and a vector of integer positions $\mathbf{p} = (p_1, p_2, \dots, p_{n'})$, where $n' \leq n$, the function constructs a vector $\mathbf{w}_1' = \left(w_{p1}, w_{p2}, \dots, w_{pn'}\right)$ by using the positions stipulated in **p**. In other words the feature-detection rule generates a vector $\mathbf{w}_1'$ by selecting elements of $\mathbf{w}_1$ as dictated by $\mathbf{p}$. For example if $\mathbf{w}_1 = (1,0,1,0,1)$ and $\mathbf{p} = (1,3,5)$ then $f_{SELECT}(\mathbf{w}_1, \mathbf{p}) = (1,1,1)$.

Now consider an antigen, $\mathbf{y} = (y_1, y_2, \dots, y_n)$ (binary vector), a detector $\mathbf{x} = (x_1, x_2, \dots, x_{n'})$ (binary vector) and a vector **p** (integer vector) of dimensionality n', where the dimensionality of the problem space/antigen is n, the dimensionality of a detector is n', $n' \leq n$ and $\mathbf{p}$ comprises a subset of feature positions of $\mathbf{y}$, that is, $\mathbf{p} = (y_i, y_{i+1}, \ldots, y_j)$**,** where each y_i appears only once and $i \geq 1$ and $j \leq n$.

Antigen $\mathbf{y}$ and detector $\mathbf{x}$ match under the feature-detection rule, $f_{FEATURE}$, if $f_{RCBITS}(\mathbf{x}, f_{SELECT}(\mathbf{y}, \mathbf{p})) \geq r$. In other words antigen $\mathbf{y}$ and detector $\mathbf{x}$ match if there are r contiguous features.

From Figure 35, consider the example where if

$$\mathbf{x} = (1,1,0,1)$$
$$\mathbf{y} = (1,0,0,1,0,0,0,1)$$
$$\mathbf{p} = (1,4,6,8)$$
$$r = 2$$

Then

$$\begin{aligned}&f_{RCBITS}(\mathbf{x}, f_{FEATURE}(\mathbf{y}, \mathbf{p}))\\&= f_{RCBITS}\left((1,1,0,1), f_{FEATURE}((1,0,0,1,0,0,0,1), (1,4,6,8))\right)\\&= f_{RCBITS}((1,1,0,1), (1,1,0,1))\\&= 4\end{aligned}$$

The following can be stated for the example above assuming that the feature-detection rule is not used:

- Depending on which features are related and whether they occur in close proximity to one another, the RCBITS rule would not be a good choice of an affinity-matching function between $\mathbf{y}$ and $\mathbf{x}$. For example, if an exclusive relationship existed between feature 1 and feature 8, then only one $\mathbf{x} = (1,0,0,1,0,0,0,1)$ with $r = 8$ would be able to detect $\mathbf{y}$. But such a detector is too specific to antigen, $\mathbf{y}$, and overfits $\mathbf{y}$ (refer to section 3.5.3). The same applies to the RCHK rule, since the RCHK rule subsumes the RCBITS rule [29].
- The HD rule has the ability to capture the relationship between different features only if the difference among the features with no relationship between $\mathbf{y}$ and $\mathbf{x}$ is less than a particular threshold, r. To illustrate what is meant by this statement, suppose that there is a relationship between feature 1 and feature 8 and no relationship between features 2, 3, 4, 5, 6 and 7, and that $r = 2$. The HD rule is

based upon the number of bits that differ, thus feature 1 and feature 8 must be equal in both $\mathbf{y}$ and $\mathbf{x}$ and at most 2 features between feature 4, 5 ,6 and 7 may not be different.

The next section discusses the matching probability, P_M, induced by the feature-detection rule.

5.2 Matching Probability and Discriminative Power of the Feature-detection Rule

The discriminative power of the feature-detection rule leverages the concepts described in section 4.3.3. Consider the following scenario, where

$$\begin{aligned} \mathbf{x} &= (1,1,0,1) \\ \mathbf{y} &= (1,0,0,1,0,0,0,1) \\ \mathbf{y}' &= (1,4,6,8) \\ r &= 3 \end{aligned}$$

The total number of strings that can be matched by $\mathbf{x}$ can be depicted by constructing two trees for the two templates induced by $\mathbf{x}$, i.e. $t_1 = 1 ** 1 * 0 **$ and $t_2 =*** 1 * 0 * 1$, as shown in Figure 36. The total number of strings that can be matched by t_1 can be calculated by noting that each '*' can represent either 0 or 1, meaning that the template can match 2^5 strings (this can be confirmed by counting the number of leafs shown in the trees of Figure 36). The same applies to t_2.

If the logic presented above is followed, then theoretically this means that $\mathbf{x}$ should match $2^5 + 2^5 = 64$ strings. However this is not the case, because the total number of unique strings that the conjunction of both templates can match is in fact $2^5 + 2^4 = 48$. This is because t_2 can detect only half of the strings that template t_1 detects (owing to overlapping strings). The number strings that overlap between templates t_1 and t_2 are given in bold in Figure 36 (at the end of the chapter).

The reason for the reduction in the discriminative power of the detectors is very similar to the argument presented by Wierzchoń [87] (refer to section 4.3.3). That is, $t_1 = 1 ** 1 * 0 **$ induces two templates, $t_1{}' = 1 ** 1 * 0 * 1$ and $t_1'' =** 1 * 0 * 0$.

Similarly, $t_2 =*** 1 * 0 * 1$ induces two templates, $t_2{}' = 0 ** 1 * 0 * 1$ and

$t_2'' = 1 ** 1 * 0 * 1$. However, since $t_1' = t_2''$, the number of strings that t_2 can detect is effectively halved. This argument can be generalised to any arbitrary detector **x**.

The probability that a detector, **x**, matches an antigen, **y**, can then be calculated by noting that a detector **x** of length n' induces $n' - r + 1$ templates. The first template can recognise 2^{n-r} strings and each subsequent template ($n' - r$ in total) can recognise only $\frac{2^{n-r}}{2}$ strings. There are 2^n strings in total. Therefore,

$$P_M = \frac{2^{n-r} + \frac{(n'-r).2^{n-r}}{2}}{2^n}$$

$$= \frac{2^{n-r} + (n'-r).2^{n-r-1}}{2^n} \qquad (5.1)$$

To compare the matching probability of the feature-detection rule with the other detection rules discussed in this thesis (refer to section 3.4.2), the matching probabilities for the RCBITS and RCHK rules are repeated below.

For the RCBITS rule:

$$P_M = \frac{2^{n-r} + \frac{(n-r).2^{n-r}}{2}}{2^n}$$

$$= \frac{2^{n-r} + (n-r).2^{n-r-1}}{2^n} \qquad (5.2)$$

The matching probability, P_M, of the RCHK rule can easily be calculated by noting that the length of a detector under the RCHK rule is equal to the affinity threshold, r. Thus, each detector of length r can recognise 2^{n-r}strings.

For the RCHK rule:

$$P_M = \frac{2^{n-r}}{2^n} \tag{5.3}$$

From equations (5.1) and (5.2), the matching probability, P_M, under the feature-detection rule is greater than P_M under the RCBITS rule if:

$$\frac{2^{n-r} + (n' - r).2^{n-r-1}}{2^n} > \frac{2^{n-r} + (n-r).2^{n-r-1}}{2^n}$$
$$\therefore 2^{n-r} + (n' - r).2^{n-r-1} > 2^{n-r} + (n-r).2^{n-r-1}$$
$$\therefore (n' - r) > (n - r)$$
$$\therefore n' > n$$

Furthermore, from equations (5.1) and (5.3) the matching probability, P_M under the feature-detection rule is greater than P_M under the RCHK rule if:

$$\frac{2^{n-r} + (n' - r).2^{n-r-1}}{2^n} > \frac{2^{n-r}}{2^n}$$
$$\therefore 2^{n-r} + (n' - r).2^{n-r-1} > 2^{n-r}$$
$$\therefore (n' - r).2^{n-r-1} > 0$$
$$\therefore (n' - r) > 0$$
$$\therefore n' > r$$

Interestingly, P_M for the feature-detection rule is calculated in a similar manner to P_M for the RCBITS rule (see equations (5.1) and (5.2)). The feature-detection rule is therefore expected to suffer from the same scaling problems as the NSA (from a purely mathematical viewpoint).

The matching probability, P_M, for the HD rule was given by equation (4.2) (repeated below for the reader's convenience),

$$P_M = 2^{-n} \sum_{i=r}^{n} \binom{n}{i} \tag{4.2}$$

From equations (5.1) and (4.2), the matching probability, P_M, under the feature-detection rule is greater than P_M under the HD rule if:

$$\frac{2^{n-r} + (n' - r).\,2^{n-r-1}}{2^n} > \frac{\sum_{i=r}^{n} \binom{n}{i}}{2^n}$$

It can be shown empirically that for any $n \leq 100000, n^{'} \leq 10000, r \leq 10000$, the matching probability,P_M under the feature-detection rule is greater than the matching probability P_M under the HD rule only if the following conditions hold true:

- $r = 0$, and $n^{'} \geq 1$, resulting in the left hand side of the equation being equal to a factor of 2^n and the right hand side of the equation being equal to 2^n; or
- values of r are sufficiently small enough and values of n' are sufficiently large enough (being almost equal to n) resulting in the left hand side of the equation being greater than or equal to 1.0.

Taken note that there is no significance attached to the number 100000, apart from the fact that it is large enough to cover most of the scenarios under which either the HD rule or feature-detection rule would ever be applied. A mathematical proof needs to be generated, so that the statement holds true for any arbitrary number, and is not within the scope of this thesis.

When the feature-detection rule is applied, a value of n' should be chosen such that $n^{'} > r$ and $n^{'} < n$, because

- if $n^{'} = r$, then the feature-detection rule is equivalent to the RCHK rule; and
- if $n^{'} = n$, then the feature-detection rule is equivalent to the RCBITS rule.

If $n^{'} > r$ and $n^{'} < n$ (which will generally be the case) then:

- P_M, under the HD rule will be greater than P_M under the feature-detection rule for most cases; and

- P_M, under the RCHK rule (the RCHK rule subsumes the RCBITS rule) will be greater than P_M under the feature-detection rule for all cases.

This effectively means that the probability of a randomly generated detector matching a self-string under the feature-detection rule is lower than that of a detector using either the RCHK , RCBITS, or HD rules.

5.3 Placing the Feature-detection Rule into Context

The most common affinity-matching functions used in the NSA, namely, the RCBITS and the RCHK rules, induce holes, which can be overcome by using a permutation mask (refer to sections 3.4.2.4 and 3.4.2.7) against an antigen to reorder the bits of the antigen. Even though permutation masks are a mathematically feasible means of overcoming both crossover and length-limited holes, major flaws exist in the way in which they are implemented. Permutation masks are currently implemented by generation of an individual random permutation mask and application of the individual random permutation mask to a population of detectors generated under the NSA. A consequence of the application of permutation masks in this manner is that their benefits are occluded by what appears to be a shattering of the entire self-space (by randomly changing the shape of the entire self-space with a randomly generated permutation mask). The shattering of the self-space attributed to permutation masks by Stibor *et al.* [80] can be explained as follows:

- A permutation mask changes the form of a shape space. Using a single random permutation mask for an entire set of detectors generated under the NSA is equivalent to taking a wild guess by trying to infer a single and meaningful alternate representation for the entire problem space. There may in fact be multiple representations of the problem space that are relevant to the problem at hand. For example, there may be multiple relationships involving entirely different subsets of features within a particular problem domain. If the RCBITS rule and the RCHK rule are considered in the same context as the problem presented in section 5.1, then two things would immediately become evident: (1)

the RCBITS rule and RCHK rule can find relationships only between features that are adjacent, and (2) a random permutation of the problem space will in fact increase the efficacy of both the RCBITS and the RCHK rule, because the result of the permutation in respect of the problem space can render two non-adjacent attributes adjacent. More random permutations are thus equivalent to finding more relationships between non-adjacent features.

- It was stated by Stibor *et al.* [80] that finding a meaningful permutation mask to apply to an entire set of detectors is computationally expensive and, in fact, infeasible. Although their statement does hold true, the problem of finding a meaningful permutation mask can be approached from another angle: Following the argument presented in the previous point, the RCBITS rule and the RCHK rule should actually be viewed as affinity-matching functions, which exploit the relationships that exist between adjacent features. Relationships between non-adjacent features can be discovered by the application of a random permutation mask to an individual detector. Following the argument of Stibor *et al.* [80], the aim should be to discover several permutation masks based on the conjunction of the problem space and the features of an individual detector, meaning that the problem is more computationally expensive than previously thought. Finding a set of meaningful features, $\mathbf{p}$, to use in the feature-detection rule is equivalent to finding a meaningful permutation mask to apply to a particular problem domain. With the only difference being that the length of $\mathbf{p}$, n', is less than or equal to the length of an artefact (an antigen, detector or self-string), n, resident within a particular problem domain.

- The approach used in this thesis, which is surprisingly simple and works exceptionally well, is to couple the generation of a random permutation mask to the generation of each individual detector under the NSA. In other words, a detector is generated together with a random permutation mask and is checked against the entire self-set to ensure that the detector is not activated by a self-string before being added to the resultant repertoire of detectors. This means that the NSA is tasked with learning a permutation mask for each detector being generated. The same approach is used when utilising the feature-detection rule in

> the NSA, i.e. the generation of meaningful features (features whose values have an impact on the outcome of the particular problem under consideration) is coupled to the generation of each individual detector. The NSA is thus performing feature extraction in addition to ascertaining whether or not a detector is activated by self.

From the above and equations (5.1) and (5.2) it is evident that if $r = n'$, then a detector generated under the feature-detection rule is equivalent to a detector generated under the RCHK rule (the RCHK rule is used because it subsumes the RCBITS rule [29]) , where each detector under the RCHK rule has a random permutation mask. But, if $n' > r$, then the discriminative power of a single detector under the feature-detection rule is equal to $n' - r$ RCHK detectors. The feature-detection rule is computationally less expensive than the RCHK rule (where each detector has a random permutation mask), because the latter case needs to generate a random detector and a random permutation mask, as well as to apply the permutation mask to an antigen, before presenting the antigen to a candidate ALC. If $r < n'$ then one detector generated under the feature-detection rule is equivalent to $n' - r$ RCHK detectors. Since the feature-detection rule is equivalent to the RCHK rule, with each detector having a random permutation mask in the worst case (when $r = n'$), it follows that the feature-detection rule cannot induce either length-limited or crossover holes.

The next section explores how the feature-detection rule introduces positional bias.

5.4 Positional Bias introduced by the Feature-Detection Rule

The feature-detection rule is a more efficient form of the RCBITS rule, in that the feature-detection rule applies the RCBITS rule to a set of features **p** of an artefact (which is an element of the problem domain), as opposed to all of the attributes comprising the artefact. The RCBITS rule however, introduces positional bias as discussed in this chapter and by Freitas *et al.* [36]. To illustrate this, consider the example presented in Figure 35. Based on Figure 35, the values of features 1, 4, 6 and 8 are relevant to the outcome of the problem i.e. feature 1 must bear a value of 1, feature 4 must bear a value of 1, feature 6 must bear a value of 0 and feature 8 must bear a value of 1. Now if an affinity threshold of $r = 4$, is employed by the feature-detection rule and the features are ordered in the manner presented above i.e. in the order of feature 1, feature 4, feature 6 and feature 8, then it is evident that there is positional bias, because if the features were selected in a different order, for example in reverse order, then the same detector that would have matched the features presented in their original sequence i.e. 1101, will not match the features if they were presented in reverse order 1011.

Now consider how the feature-detection rule is applied within the context of the thesis, the features together with the resultant detector (each detector has its own set of features) are generated by employing the NSA i.e. the features are selected randomly against a randomly generated detector, meaning that regardless of the order in which the features are presented, the detector will only be activated, under the feature-detection rule, if its attributes match the re-ordered features. Thus if the feature-detection rule is applied within the context of the NSA, it does not introduce positional bias.

To illustrate this example, consider the following scenario (with reference to Figure 35) with two randomly generated detectors: $\mathbf{x}_1 = (1,1,0,1)$ with a position vector, $\mathbf{p}_1 = (1,4,6,8)$ and $\mathbf{x}_{2=}(1,0,1,1)$ with a position vector, $\mathbf{p}_2 = (8,6,4,1)$**,** an affinity threshold, $r = 4$ and an antigen $\mathbf{y} = (1,0,0,1,0,0,0,1)$. Regardless of the fact that both $\mathbf{x}_1$ and $\mathbf{x}_2$ select the same set of features, albeit in a different order, both $\mathbf{x}_1$ and $\mathbf{x}_2$ are activated by antigen $\mathbf{y}$.

5.5 Conclusion

This chapter formally introduced the feature-detection rule and also argued that permutation masks are not being applied correctly to the NSA, resulting in the pivotal role that permutation masks play in reducing length-limited and crossover holes (induced by both the RCHK and RCBITS rule) being misconstrued. The feature-detection rule is computationally less expensive than both the RCHK rule or the RCBITS rule and has the following additional interesting properties:

- If $r = n'$, then the feature-detection rule is equivalent to the RCHK rule (with a random permutation mask applied to each detector).

- If $r < n'$, then each individual detector under the feature-detection rule is equivalent to $n' - r$ RCHK detectors. A value of $r < n'$ should ideally be used when the feature-detection rule is applied to a particular problem space to benefit from an increase in the discriminative power. Furthermore, if $r < n'$, then fewer detectors generated under the feature-detection rule would be needed to cover a particular problem space, as opposed to detectors generated under the RCHK rule.
- If $n = n'$, then the feature-detection rule is equivalent to the RCBITS rule (with a random permutation mask applied to each detector).
- Due to the fact that the feature-detection rule utilises the RCBITS rule it introduces positional bias, which in turn is overcome by coupling the coupling the generation of the feature position vector, **p**, with each detector generated by the NSA.

The next chapter, conducts a number of experiments using the feature-detection rule, the RCHK rule with and without random permutation masks for each detector, and the HD rule.

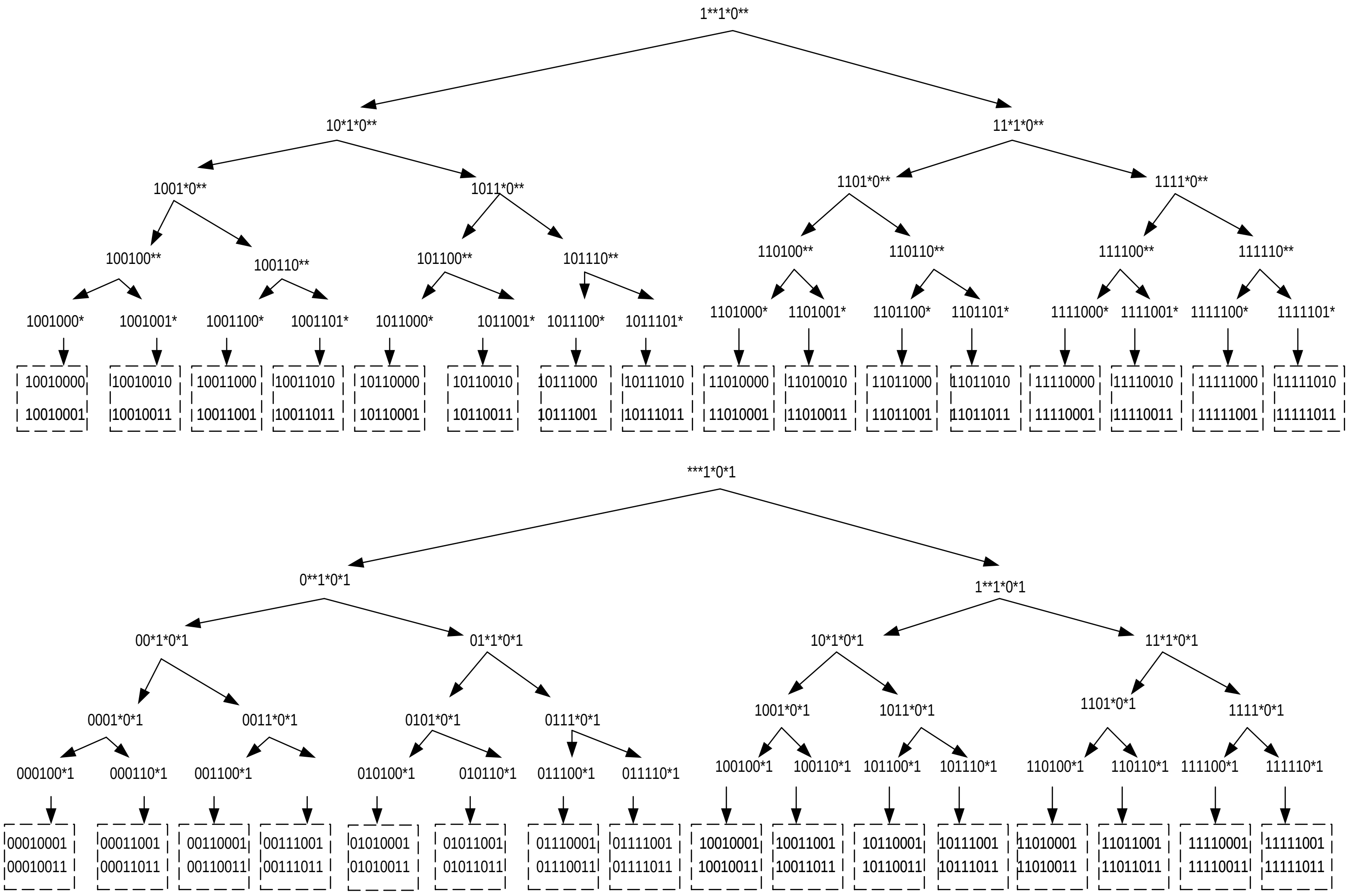


Figure 36. *Discriminative power of template 1**1*0** and ***1*0*1 induced by **x**. Note that the number of overlapping strings is shown in bold.*

Chapter 6
Experimental Results

"Facts are stubborn things, but statistics are more pliable."

- Mark Twain

The objectives of this chapter are to validate the concepts presented in the previous chapter by:

- demonstrating the efficacy of the feature-detection rule in contrast to the RCHK rule (with no permutation masks) and the HD rule;
- demonstrating the efficacy of the feature-detection rule in contrast to the RCHK rule (where each detector has a random permutation mask). For the purposes of this chapter, scenarios in which a random permutation mask was used in conjunction with the RCHK rule are denoted as RCHK (MHC). Conversely, scenarios in which MHC masks were not used in conjunction with the RCHK rule are denoted as RCHK (No MHC);
- demonstrating how the application of an individual global MHC mask applied to a set of already generated detectors impedes the performance of the set of detectors. This point is important because it will validate the assertion made by this thesis that MHC masks are being applied incorrectly within the context of the NSA. For the purpose of this chapter, scenarios in which a global MHC mask is applied to a pre-generated set of detectors is denoted as RCHK (Single global MHC); and
- demonstrating that the performance (detection rate and false-alarm rate) of the feature-detection rule is equivalent to the RCHK rule (where each detector has a random permutation mask) at worst case.

The procedure used to conduct the experiments and scenarios tested by each experiment is presented in section 6.1, followed by an empirical analysis of the results produced in section 6.2. Finally, the conclusions inferred by the experiments are presented in section 6.3.

6.1 Experimental Procedure

An experiment is comprised of five scenarios where a scenario pertains to a particular high-level objective:

- Training a set of detectors with the NSA utilising the feature-detection rule.
- Training a set of detectors with the NSA utilising the HD rule.
- Training a set of detectors with the NSA utilising the RCHK rule with no permutation mask, denoted by RCHK (No MHC).
- Training a set of detectors with the NSA utilising the RCHK rule with a single global permutation mask, denoted by RCHK (Global MHC). Take note that the test sets within this particular scenario are executed by: firstly generating a set of detectors and then applying a single global permutation mask to the generated detector set.
- Training a set of detectors with the NSA utilising the RCHK rule where each detector has its own randomly generated permutation mask, denoted by RCHK (MHC).

A scenario is comprised of a number of test groups. Each test group utilises a different set of parameters (e.g. different values of r) to test the scenario. A test group in turn is comprised of several tests. Each test is executed with the parameters stipulated by its test group in addition to using a particular target population size, n_c. That is, different tests have different n_c values. The last test executed within a test group has the largest n_c value and is called the target test. The objective of a target test is to measure the performance of the NSA given a maximum n_c value in conjunction with the parameters pertaining to the target test. Test groups are compared to one another within the same scenario and across scenarios based on the results achieved by target tests.

For each test executed within a particular test group, the TP, FP, FN, OC, GC, DR, FR, n_c, and the actual population size are recorded. The overfitting count and generalisation count, OC and GC, respectively, are used to infer the current spread of the detectors and whether there are too many or too few detectors.

The actual population size is a critical metric used over and above the DR and FR metrics to quantify the performance of a detection rule within a particular scenario. The actual population size is the resultant number of detectors produced by the NSA given a target of n_c detectors as input. It is often the case that, given a target n_c, the NSA can take an inordinate amount of time to generate n_c detectors which are self-tolerant (not activated by self), depending on the complexity of the data set and the detection rule used.

In an attempt to minimise the time taken by the NSA to generate a candidate set of detectors, the implementation of the NSA employed in this thesis attempts to generate n_c detectors at most $1000 * n_c$ times, where after the resultant data set is returned. This means that if the actual population size is less than n_c, then it reflects that the NSA failed to generate n_c detectors within an acceptable time frame.

The relationship between experiments, scenarios, test groups and tests are illustrated in Figure 37.

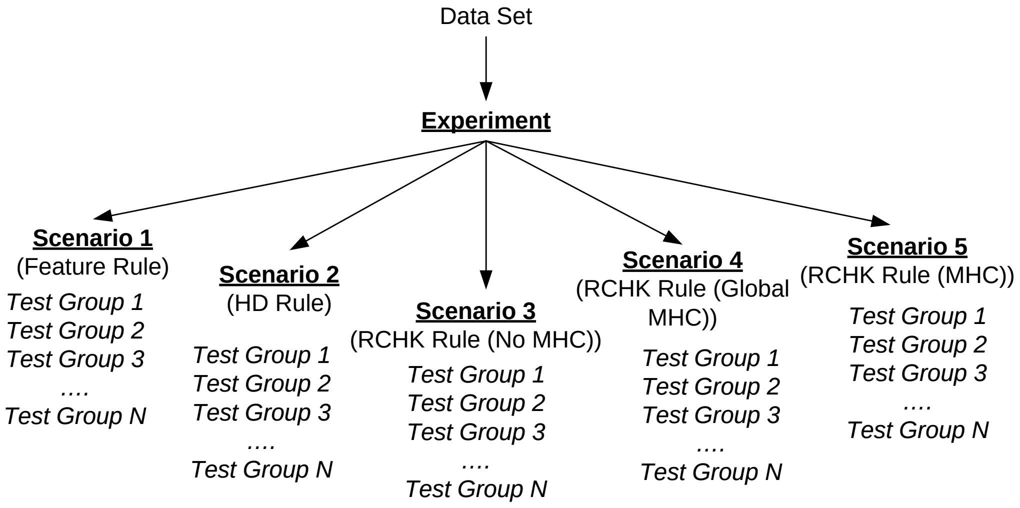


Figure 37. *Relationship between Experiments, Scenarios and Tests*

Each experiment is performed on its own individual data set. The data sets become increasingly more complex for each successive experiment performed in this chapter. The same data sets originally used by Graaff [41] to test the efficacy of an AIS comprised of evolved detectors, are used by this thesis. The data sets were collected by Graaff from the UCI Machine Learning Repository and converted to binary strings [2]. The data were manipulated further by grouping some of the data sets into subsets comprising self and non-self strings, respectively. The way in which a particular data set is fragmented into a self and non-self set is explained in each section pertaining to a particular experiment (see section 6.2).

A training set is created based on the self-data by randomly selecting 70% of the original self-set. The test set comprises the remaining 30% of the original self-set and a non-self set associated with the training data. To ensure that the results are statistically significant, each test within a particular test group is executed 30 times, and the result of each metric records the mean value of the metric and the standard deviation of the metric from the mean. Take note that a new test set and training set is randomly created for each new execution of a test pertaining to a particular test set.

The parameters used to test the NSA within a particular scenario were chosen by applying the framework suggested by Timmis *et al.* to choose an optimum affinity threshold, r, for the NSA under the RCBITS rule [4] (see section 4.1). Due to the mathematical similarity between the RCHK rule (regardless of how a permutation mask is applied to a detector or set of detectors) and the feature-detection rule, (see Chapter 5) the same set of affinity thresholds are used across all of the test groups in scenarios 1, 3, 4 and, 5 (see Figure 37) for each particular experiment.

Conversely, due to the differences in the mechanisms employed by the RCHK rule/feature-detection rule and the HD rule to determine whether two binary strings are activated by one another, the same affinity threshold, r, cannot be reused to compare the performance of the HD rule to either the RCHK rule or the feature-detection rule. Instead, the performance of the HD rule is optimised (by choosing an optimal r value) and scenarios are compared to one another within a particular experiment based on each scenario’s best performing test group. The best performing test group has the greatest average DR minus FR value for its target test (the last test executed within the test group). Take note that the worst performing test group has the lowest average DR minus FR value for its target test.

To enhance the readability of the tests results presented in this chapter, the following annotations are used when presenting the results of a scenario:

- The table row corresponding to the best performing test group for the entire scenario is highlighted in green.
- The table row corresponding to the worst performing test group for the scenario is highlighted in red.
- Table rows corresponding to target tests are outlined in bold.

6.2 Empirical Analysis of Results

This section discusses the empirical analysis and results pertaining to five experiments. Each experiment leverages its own data set and the complexity of each data set increases in each successive experiment.

6.2.1 Car Evaluation Experiment

The car evaluation data set imposes a valuation of cars based upon three characteristics, namely; price, technical characteristics, and comfort. The price factor includes the price of the car and the cost of maintaining the car. The technical factor addresses the safety of the car. The comfort factor is concerned with the car's carrying capacity and the size of the car's boot in terms of luggage capacity. The data set comprises 1728 patterns, distributed between four classes: good, acceptable, unacceptable and very good. Graaff converted each pattern into a binary string of length 13 [41]. Each element within a binary string was treated as an individual attribute by the NSA. The same applies for the rest of the experiments conducted.

The data sets were processed further to create a single self-set and non-self set as follows:

- Acceptable.self: Contains 384 patterns relating to the acceptable class.
- Acceptable.non-self: Contains all the patterns related to the unacceptable, good and very good classes. The set contains 1 344 patterns in total.

The results of each scenario are reported as per the list below:

- The results of scenario 1, the car-evaluation data set, under the feature-detection rule are tabulated in Table 1 and Table 2.
- The results of scenario 2, the car evaluation data set, under the HD rule are tabulated in Table 3.
- The results of scenario 3, the car evaluation data set, under the RCHK rule with no permutation mark are tabulated in Table 4.
- The results of scenario 5, the car evaluation data set, under the RCHK rule with a single global permutation mask are tabulated in Table 5.
- The results of scenario 4, the car evaluation data set, under the RCHK rule with each detector having its own random permutation mask are tabulated in Table 6.

Table 1. *Car Evaluation Data Set test results under Feature-Detection Rule (Part 1)*

Detector Length	r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
3	2	50	1.0±0.0	0.34689826±0.34201074	0.0±0.0	0.6531017±0.34201074	0.8347826±1.8666307	10.369735±10.594484	0.6270664±0.13540047	0.0±0.0	44.0±1.5811388
3	2	150	1.0±0.0	0.6±0.1726341	0.0±0.0	0.39999998±0.17263411	2.9078918±0.9240834	10.660208±2.3427403	0.7235007±0.09341315	0.0±0.0	129.6±5.3197746
3	2	500	1.0±0.0	0.7890819±0.0	0.0±0.0	0.21091811±0.0	5.6470976±2.90152	9.25022±3.299672	0.8258198±6.664002E-8	0.0±0.0	434.0±9.617692
3	2	1500	1.0±0.0	0.7890819±0.0	0.0±0.0	0.21091811±0.0	6.223884±1.4062006	7.6081543±1.412988	0.8258198±6.664002E-8	0.0±0.0	1316.8±3.7682889
3	**2**	**5000**	**1.0±0.0**	**0.7890819±0.0**	**0.0±0.0**	**0.21091811±0.0**	**7.5739183±1.2592841**	**7.99926±1.2543238**	**0.8258198±6.664002E-8**	**0.0±0.0**	**4363.4±24.337214**
3	3	50	1.0±0.0	0.596526±0.12940393	0.0±0.0	0.40347394±0.12940393	10.868±5.5392036	31.584±8.982258	0.71814245±0.067923285	0.0±0.0	50.0±0.0
3	3	150	1.0±0.0	0.7786601±0.02546163	0.0±0.0	0.22133994±0.025461629	22.699999±6.0601187	32.896004±6.9612565	0.8190795±0.016341219	0.0±0.0	150.0±0.0
3	3	500	1.0±0.0	0.78908193±6.282881E-8	0.0±0.0	0.2109181±1.5707203E-8	32.237797±3.3406363	35.834732±3.353956	0.82581985±1.2565762E-7	0.0±0.0	499.9±0.3162278
3	3	1500	1.0±0.0	0.78908193±6.282881E-8	0.0±0.0	0.2109181±1.5707203E-8	32.539207±3.4591494	33.789494±3.4547026	0.82581985±1.2565762E-7	0.0±0.0	1499.9±0.31622773
3	**3**	**5000**	**1.0±0.0**	**0.78908193±6.282881E-8**	**0.0±0.0**	**0.2109181±1.5707203E-8**	**35.188686±1.7760925**	**35.575317±1.7741834**	**0.82581985±1.2565762E-7**	**0.0±0.0**	**4999.3±0.6749486**
5	3	50	1.0±0.0	0.5081886±0.29454103	0.0±0.0	0.49181142±0.29454106	3.7136116±2.306355	19.036303±10.862264	0.6882226±0.11434687	0.0±0.0	44.8±1.9235383
5	3	500	1.0±0.0	0.7890819±0.0	0.0±0.0	0.21091811±0.0	16.109928±3.9183753	19.831661±4.034572	0.8258198±6.664002E-8	0.0±0.0	450.4±6.8774996
5	3	1500	1.0±0.0	0.7890819±0.0	0.0±0.0	0.21091811±0.0	19.708607±0.74742484	21.046814±0.7451664	0.8258198±6.664002E-8	0.0±0.0	1360.4±7.7006493
5	**3**	**5000**	**1.0±0.0**	**0.7890819±0.0**	**0.0±0.0**	**0.21091811±0.0**	**19.958355±0.7083699**	**20.372671±0.70495766**	**0.8258198±6.664002E-8**	**0.0±0.0**	**4519.4±11.371016**
5	4	50	1.0±0.0	0.63374686±0.11454241	0.0±0.0	0.36625308±0.11454241	10.588±7.9717956	30.128002±12.498653	0.73605597±0.061627515	0.0±0.0	50.0±0.0
5	4	150	1.0±0.0	0.7851116±0.0045076576	0.0±0.0	0.21488833±0.0045076692	21.321331±2.5096369	31.186666±3.1381807	0.82312995±0.003050299	0.0±0.0	150.0±0.0
5	4	500	1.0±0.0	0.853598±0.038760547	0.0±0.0	0.14640197±0.038760547	25.8664±2.7545605	29.307201±2.7550654	0.87307054±0.028708184	0.0±0.0	500.0±0.0
5	4	1500	0.99304354±0.015555247	0.8357321±0.052239012	0.0069565214±0.015555255	0.16426799±0.052238993	26.642132±1.9909208	27.821331±2.0269582	0.85943145±0.039744098	0.0069565214±0.015555255	1500.0±0.0
5	**4**	**5000**	**1.0±0.0**	**0.9012407±0.006658271**	**0.0±0.0**	**0.0987593±0.006658264**	**27.915562±1.0472664**	**28.2996±1.0362086**	**0.91014445±0.0055605234**	**0.0±0.0**	**5000.0±0.0**
5	5	50	0.99304354±0.015555247	0.44367248±0.105355375	0.0069565214±0.015555255	0.5563276±0.10535537	4.0439997±4.106882	15.935999±7.0568886	0.6434466±0.048531227	0.0069565214±0.015555255	50.0±0.0
5	5	150	0.99304354±0.015555247	0.73995036±0.05360606	0.0069565214±0.015555255	0.26004964±0.053606056	7.2666664±1.8918655	15.106667±2.3714223	0.7937865±0.03230375	0.0069565214±0.015555255	150.0±0.0
5	5	500	0.99304354±0.00952562	0.8913151±0.026342487	0.0069565214±0.009525609	0.10868486±0.026342494	11.9428±1.8103461	15.1224±1.8506056	0.9017253±0.021868793	0.0069565214±0.009525609	500.0±0.0
5	5	1500	0.9756522±0.019829132	0.92208445±0.023051558	0.024347825±0.019829137	0.07791563±0.02305156	14.355867±0.65841585	15.532534±0.64935255	0.92651767±0.019433595	0.024347825±0.019829137	1500.0±0.0
5	**5**	**5000**	**0.9373913±0.037603036**	**0.955335±0.009448863**	**0.0626087±0.03760304**	**0.044665016±0.0094488505**	**15.0174±0.4538391**	**15.3915205±0.4594334**	**0.9545604±0.0091885645**	**0.0626087±0.03760304**	**5000.0±0.0**
6	4	50	1.0±0.0	0.60347396±0.115129516	0.0±0.0	0.39652604±0.11512952	8.992±6.9031096	25.904001±11.719168	0.7199381±0.058926884	0.0±0.0	50.0±0.0
6	4	150	1.0±0.0	0.77568233±0.07078705	0.0±0.0	0.22431763±0.07078705	19.342667±6.890137	29.178665±7.89573	0.8189026±0.045930147	0.0±0.0	150.0±0.0
6	4	500	1.0±0.0	0.83076924±0.038055178	0.0±0.0	0.16923077±0.038055167	25.48743±2.9862356	28.7329±2.952363	0.8559801±0.02753249	0.0±0.0	498.6±0.5477225
6	4	1500	1.0±0.0	0.8843673±0.03223898	0.0±0.0	0.11563276±0.032238968	26.255238±2.3171303	27.501959±2.3159204	0.89693344±0.025141811	0.0±0.0	1494.2±1.9235384
6	**4**	**5000**	**0.99304354±0.015555247**	**0.9091811±0.010759049**	**0.0069565214±0.015555255**	**0.09081886±0.010759046**	**26.78363±1.3870422**	**27.171045±1.3761488**	**0.916337±0.00830487**	**0.0069565214±0.015555255**	**4985.2±5.540758**

Table 2. *Car Evaluation Data Set test results under Feature-Detection Rule (Part 2)*

Detector Length	r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
6	5	50	0.9895652±0.0155552495	0.48486352±0.09490463	0.010434782±0.015555255	0.5151365±0.09490463	4.808±2.3366044	18.248±3.6769986	0.65980554±0.042294133	0.010434782±0.015555255	50.0±0.0
6	5	150	0.99304354±0.015555248	0.7602977±0.064123645	0.0069565214±0.015555255	0.23970222±0.06412363	8.904±1.5526508	16.880001±1.7105168	0.80729735±0.03969928	0.0069565214±0.015555255	150.0±0.0
6	5	500	0.98782605±0.022675486	0.8818858±0.030268885	0.0121739125±0.022675486	0.118114136±0.030268883	16.0232±2.7464504	19.307598±2.7844162	0.8938341±0.022675406	0.0121739125±0.022675486	500.0±0.0
6	5	1500	0.97913045±0.046665765	0.9032258±0.017546088	0.020869564±0.046665765	0.09677419±0.017546073	17.633867±0.42715433	18.814533±0.40394115	0.91025317±0.013802266	0.020869564±0.046665765	1500.0±0.0
6	**5**	**5000**	**0.973913±0.022169646**	**0.93349874±0.016214706**	**0.026086956±0.02216965**	**0.066501245±0.0162147**	**18.41776±0.6698696**	**18.80188±0.6677639**	**0.9363154±0.013973414**	**0.026086956±0.02216965**	**5000.0±0.0**
6	6	50	0.99304354±0.012160102	0.33548385±0.07719209	0.0069565214±0.012160101	0.6645161±0.07719209	1.926±1.5390054	11.256±3.2695878	0.60025096±0.027921198	0.0069565214±0.012160101	50.0±0.0
6	6	150	0.97913045±0.026628824	0.6332506±0.05175745	0.020869564±0.026628833	0.36674935±0.051757444	3.5826669±1.0072635	9.775333±1.4100066	0.72812307±0.031449553	0.020869564±0.026628833	150.0±0.0
6	6	500	0.93391305±0.04922419	0.885608±0.019935371	0.06608696±0.049224187	0.114392065±0.019935373	6.6879997±0.74012125	9.6934±0.761829	0.8908706±0.018771568	0.06608696±0.049224187	500.0±0.0
6	6	1500	0.82260865±0.03693806	0.9513647±0.010733585	0.17739132±0.03693805	0.048635233±0.010733584	8.4504±0.43172964	9.581266±0.44280118	0.94435966±0.010974495	0.17739132±0.03693805	1500.0±0.0
6	**6**	**5000**	**0.6808696±0.07088719**	**0.9806452±0.010960652**	**0.31913048±0.0708872**	**0.019354839±0.01096064**	**9.122801±0.26813093**	**9.4779±0.26626307**	**0.97288924±0.014271054**	**0.31913048±0.0708872**	**5000.0±0.0**
10	3	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
10	3	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
10	3	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
10	3	1500	1.0±0.0	0.03126551±0.06991181	0.0±0.0	0.96873456±0.0699118	0.0±0.0	38.4±85.86501	0.5084791±0.018959912	0.0±0.0	0.2±0.44721362
10	**3**	**5000**	**1.0±0.0**	**0.101240695±0.13872944**	**0.0±0.0**	**0.89875925±0.13872942**	**0.0±0.0**	**121.6±166.89159**	**0.5289814±0.039721936**	**0.0±0.0**	**0.4±0.5477226**
10	10	50	0.9765218±0.009211725	0.03796526±0.020597028	0.023478258±0.009211738	0.9620348±0.020597033	0.02±0.02108185	1.1520001±0.31243846	0.5037774±0.005812775	0.023478258±0.009211738	50.0±0.0
10	10	150	0.9504348±0.015902458	0.12928039±0.03405224	0.04956522±0.015902454	0.8707196±0.03405224	0.051999997±0.024904758	1.06±0.14190939	0.52201927±0.010087935	0.04956522±0.015902454	150.0±0.0
10	10	500	0.83652174±0.03424709	0.35186106±0.06168415	0.16347827±0.03424709	0.64813894±0.06168414	0.1844±0.038225062	1.1304±0.10664915	0.56411344±0.021933494	0.16347827±0.03424709	500.0±0.0
10	10	1500	0.5895652±0.034247085	0.67270476±0.019901024	0.41043478±0.03424708	0.32729527±0.019901032	0.4256±0.04601363	1.0663334±0.064504385	0.6428601±0.018071339	0.41043478±0.03424708	1500.0±0.0
10	**10**	**5000**	**0.2±0.029842405**	**0.97493804±0.009094671**	**0.79999995±0.029842408**	**0.025062034±0.009094668**	**0.7968999±0.023988204**	**1.0832398±0.025059715**	**0.886607±0.049325854**	**0.79999995±0.029842408**	**5000.0±0.0**
13	5	50	1.0±0.0	0.5315136±0.17376284	0.0±0.0	0.4684863±0.17376283	24.988972±15.050846	87.13451±25.017529	0.68877673±0.08259607	0.0±0.0	12.6±4.3931766
13	5	150	1.0±0.0	0.817866±0.049147803	0.0±0.0	0.182134±0.049147803	49.800087±8.985213	84.64915±12.932692	0.8471082±0.035493888	0.0±0.0	40.2±3.4928498
13	5	500	0.9947826±0.0077776364	0.89478904±0.020582072	0.005217391±0.0077776276	0.10521092±0.020582072	70.09715±2.367076	82.91542±2.612662	0.9046537±0.016435819	0.005217391±0.0077776276	132.4±6.7675695
13	5	1500	0.9547826±0.033339623	0.9290323±0.024476629	0.045217387±0.03333963	0.07096775±0.024476616	79.727295±4.0189977	84.32462±4.05154	0.9317395±0.020666404	0.045217387±0.03333963	394.8±16.39207
13	**5**	**5000**	**0.9060869±0.060993653**	**0.95732003±0.015436562**	**0.09391304±0.06099366**	**0.042679902±0.015436558**	**83.91581±5.068424**	**85.39886±5.041089**	**0.95521414±0.015067842**	**0.09391304±0.06099366**	**1270.0±29.53811**
13	13	50	0.9956522±0.0061487616	0.008933002±0.00408572	0.004347826±0.0061487537	0.99106693±0.0040857196	0.0±0.0	0.194±0.040055517	0.50115156±0.001862451	0.004347826±0.0061487537	50.0±0.0
13	13	150	0.9808696±0.013471242	0.019602977±0.0064547937	0.019130435±0.013471246	0.98039705±0.0064547956	0.0019999999±0.003220306	0.16866668±0.036925018	0.50010407±0.0032387571	0.019130435±0.013471246	150.0±0.0
13	13	500	0.946087±0.016801605	0.061042182±0.013347325	0.05391304±0.016801596	0.93895787±0.013347325	0.0050000004±0.0028674419	0.16160001±0.019409047	0.5018785±0.040656333	0.05391304±0.016801596	500.0±0.0
13	13	1500	0.81217396±0.06137814	0.1776675±0.023312738	0.18782608±0.06137814	0.8223325±0.023312742	0.0150666665±0.0023136735	0.16833332±0.010343482	0.49632487±0.021494161	0.18782608±0.06137814	1500.0±0.0
13	**13**	**5000**	**0.5121739±0.036307175**	**0.466005±0.028311532**	**0.48782605±0.036307167**	**0.53399503±0.02831153**	**0.044±0.002315167**	**0.16905999±0.0052008964**	**0.48935682±0.026420228**	**0.48782605±0.036307167**	**5000.0±0.0**

Table 3. *Car Evaluation Data Set test results under HD Rule*

r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
3	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
3	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
3	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
3	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
3	**5000**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.5±0.0**	**0.0±0.0**	**0.0±0.0**
5	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
5	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
5	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
5	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
5	**5000**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.5±0.0**	**0.0±0.0**	**0.0±0.0**
6	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
6	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
6	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
6	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
6	**5000**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.5±0.0**	**0.0±0.0**	**0.0±0.0**
9	50	0.9808696±0.020413697	0.44540945±0.10752848	0.019130433±0.020413699	0.5545906±0.10752848	11.349294±3.6571944	49.605957±8.747039	0.64163774±0.044072818	0.019130433±0.020413699	18.5±4.326918
9	150	0.97304356±0.022988265	0.6401984±0.057768982	0.026956523±0.022988265	0.35980147±0.057768967	30.950256±3.9141574	52.782677±5.165424	0.73144644±0.027498888	0.026956523±0.022988265	57.4±6.040603
9	500	0.9713044±0.024951106	0.7059554±0.03527202	0.028695654±0.024951117	0.29404464±0.03527202	42.928288±3.6040816	50.841377±3.5777516	0.7683508±0.017654179	0.028695654±0.024951117	198.9±11.779926
9	1500	0.98086965±0.0114483535	0.69181144±0.030114787	0.019130435±0.011448358	0.3081886±0.030114796	49.190403±2.0722256	51.97506±2.0802689	0.76136017±0.016379708	0.019130435±0.011448358	575.6±28.682941
9	**5000**	**0.97217387±0.017294412**	**0.70148885±0.02874916**	**0.027826086±0.017294416**	**0.29851115±0.028749157**	**50.592796±0.41329703**	**51.38978±0.40452722**	**0.76553756±0.014868999**	**0.027826086±0.017294416**	**1990.2±75.09372**
10	50	0.93826085±0.027937569	0.50595534±0.09179612	0.061739128±0.027937578	0.4940447±0.09179614	5.6523905±1.7752867	20.49006±3.2997348	0.65722585±0.04359591	0.061739128±0.027937578	49.6±0.6992059
10	150	0.85739124±0.047486726	0.76625305±0.053204462	0.1426087±0.047486722	0.23374692±0.053204447	11.577742±1.3957705	20.23759±1.7036465	0.7873149±0.03791639	0.1426087±0.047486722	149.1±0.73786473
10	500	0.7408696±0.06543282	0.9171216±0.016258936	0.25913042±0.06543282	0.08287841±0.01625894	17.58783±1.1701149	21.022486±1.1942682	0.8997275±0.015591858	0.25913042±0.06543282	497.6±1.3498971
10	1500	0.70869565±0.049403075	0.9610421±0.0070233005	0.29130435±0.049403075	0.038957816±0.007023303	20.340313±0.6993385	21.553984±0.70822	0.9478022±0.009184392	0.29130435±0.049403075	1494.0±2.0548048
10	**5000**	**0.6695652±0.0765788**	**0.9625311±0.009390746**	**0.33043483±0.07657881**	**0.037468985±0.009390747**	**21.028452±0.40354848**	**21.39462±0.404416**	**0.9473878±0.009549535**	**0.33043483±0.07657881**	**4979.6±5.3789716**
13	50	0.99217397±0.009569619	0.0037220842±0.0026802071	0.007826087±0.009569608	0.9962779±0.002680201	0.0±0.0	0.134±0.049035136	0.4989584±0.0024793262	0.007826087±0.009569608	50.0±0.0
13	150	0.9756522±0.016801592	0.022332506±0.0064069205	0.024347825±0.016801596	0.97766745±0.0064069265	6.6666666E-4±0.002108185	0.18133333±0.032325506	0.4994554±0.0046378346	0.024347825±0.016801596	150.0±0.0
13	500	0.94956523±0.017773582	0.05459057±0.013182281	0.05043478±0.017773576	0.9454095±0.013182269	0.0040±0.0026666666	0.167±0.02091783	0.50107944±0.0054825	0.05043478±0.017773576	500.0±0.0
13	1500	0.8217392±0.027877375	0.16848636±0.012125286	0.17826086±0.027877368	0.8315136±0.012125287	0.015666667±0.002901681	0.16753332±0.008526211	0.49693838±0.010881279	0.17826086±0.027877368	1500.0±0.0
13	**5000**	**0.5121739±0.05932477**	**0.45756823±0.026056552**	**0.48782605±0.05932477**	**0.5424318±0.026056565**	**0.044460002±0.0027921316**	**0.17174±0.0038976343**	**0.48446384±0.03272004**	**0.48782605±0.05932477**	**5000.0±0.0**

Table 4. *Car Evaluation Data Set test results under RCHK (no MHC) Rule*

r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
3	50	1.0±0.0	0.6937965±0.09147324	0.0±0.0	0.30620348±0.09147323	115.282555±22.017618	153.52303±21.84598	0.7689472±0.05351272	0.0±0.0	37.1±3.5103025
3	150	1.0±0.0	0.7573201±0.051212616	0.0±0.0	0.24267991±0.051212635	125.97691±18.168451	140.51581±18.225851	0.8059033±0.03210646	0.0±0.0	111.0±7.055337
3	500	1.0±0.0	0.78908193±6.282881E-8	0.0±0.0	0.2109181±1.5707203E-8	137.8957±7.8437357	142.72743±7.8088846	0.82581985±1.2565762E-7	0.0±0.0	368.2±9.658617
3	1500	1.0±0.0	0.78908193±6.282881E-8	0.0±0.0	0.2109181±1.5707203E-8	143.77979±2.669157	145.45421±2.6706285	0.82581985±1.2565762E-7	0.0±0.0	1102.0±11.728408
3	**5000**	**1.0±0.0**	**0.78908193±6.282881E-8**	**0.0±0.0**	**0.2109181±1.5707203E-8**	**143.24014±2.0761178**	**143.73682±2.066644**	**0.82581985±1.2565762E-7**	**0.0±0.0**	**3677.5±23.453262**
5	50	1.0±0.0	0.7372209±0.04793715	0.0±0.0	0.26277915±0.047937155	30.039429±15.00694	56.68343±17.318945	0.7929034±0.02927119	0.0±0.0	49.6±0.51639783
5	150	1.0±0.0	0.78908193±6.282881E-8	0.0±0.0	0.2109181±1.5707203E-8	36.578262±5.2656755	47.476902±5.208559	0.82581985±1.2565762E-7	0.0±0.0	148.4±1.3498971
5	500	1.0±0.0	0.78908193±6.282881E-8	0.0±0.0	0.2109181±1.5707203E-8	49.72654±3.2125247	53.08362±3.2111003	0.82581985±1.2565762E-7	0.0±0.0	492.9±2.5582113
5	1500	1.0±0.0	0.78908193±6.282881E-8	0.0±0.0	0.2109181±1.5707203E-8	50.119057±2.1898417	51.234295±2.1824021	0.82581985±1.2565762E-7	0.0±0.0	1481.6±5.929212
5	**5000**	**1.0±0.0**	**0.78908193±6.282881E-8**	**0.0±0.0**	**0.2109181±1.5707203E-8**	**51.182255±1.0891833**	**51.5172±1.0886565**	**0.82581985±1.2565762E-7**	**0.0±0.0**	**4931.9±7.7093015**
6	50	1.0±0.0	0.48982638±0.0646242	0.0±0.0	0.5101737±0.064624205	16.765999±5.802858	32.424004±7.511584	0.6632675±0.028363917	0.0±0.0	50.0±0.0
6	150	0.9965218±0.010999221	0.6679901±0.041830197	0.0034782607±0.010999227	0.3320099±0.04183021	27.314667±2.2658448	34.827335±2.3699522	0.750663±0.024870014	0.0034782607±0.010999227	150.0±0.0
6	500	0.9904348±0.015582234	0.7645161±0.027667465	0.009565217±0.015582237	0.23548386±0.027667461	29.090122±2.496486	31.875097±2.5462694	0.8082434±0.01858833	0.009565217±0.015582237	499.8±0.421637
6	1500	0.9869565±0.02100199	0.78982633±0.0011986304	0.013043478±0.021001996	0.21017368±0.0011986234	30.661438±1.4157709	31.720022±1.412055	0.82439727±0.0022904724	0.013043478±0.021001996	1498.9±0.99442893
6	**5000**	**0.9843478±0.022377117**	**0.79081887±0.0031058844**	**0.015652174±0.022377111**	**0.20918112±0.0031058716**	**31.001312±0.57114166**	**31.327793±0.5722672**	**0.8246962±0.0033092082**	**0.015652174±0.022377111**	**4997.6±1.9550505**
10	50	0.9652174±0.022074703	0.06699751±0.008271298	0.034782607±0.022074705	0.9330025±0.0082712965	0.033999998±0.036575645	1.882±0.28696883	0.5084339±0.005985348	0.034782607±0.022074705	50.0±0.0
10	150	0.9026086±0.019573256	0.19181141±0.027006773	0.09739131±0.019573266	0.8081886±0.027006771	0.21399999±0.057387568	2.0053334±0.28641334	0.52766114±0.006963673	0.09739131±0.019573266	150.0±0.0
10	500	0.767826±0.036903907	0.49280396±0.035329193	0.23217389±0.03690391	0.50719607±0.0353292	0.5418±0.029506305	1.9546001±0.06463608	0.6023965±0.015774054	0.23217389±0.03690391	500.0±0.0
10	1500	0.49304348±0.034062587	0.7531017±0.020933082	0.5069565±0.0340626	0.24689826±0.020933086	1.0709333±0.055873916	1.8901335±0.05978133	0.66627073±0.021158325	0.5069565±0.0340626	1500.0±0.0
10	**5000**	**0.18434784±0.04256036**	**0.8992556±0.011997655**	**0.8156522±0.042560354**	**0.10074441±0.011997661**	**1.6117799±0.030573247**	**1.9298999±0.03208311**	**0.6413342±0.055760883**	**0.8156522±0.042560354**	**5000.0±0.0**
13	50	0.99304354±0.0068592303	0.007692308±0.0029707667	0.0069565214±0.006859223	0.99230754±0.0029707565	0.0039999997±0.00843274	0.178±0.04467164	0.5001809±0.0019476732	0.0069565214±0.006859223	50.0±0.0
13	150	0.98347837±0.011915827	0.022332508±0.0042175516	0.016521737±0.011915829	0.9776676±0.004217551	0.0013333333±0.0028109136	0.17±0.017284831	0.5014668±0.0030785354	0.016521737±0.011915829	150.0±0.0
13	500	0.9260869±0.019765481	0.055583127±0.014805394	0.07391304±0.01976548	0.9444169±0.014805384	0.0068000006±0.003910101	0.17160001±0.018130392	0.49507746±0.007524564	0.07391304±0.01976548	500.0±0.0
13	1500	0.82521737±0.036307167	0.17220843±0.034743395	0.1747826±0.036307167	0.8277916±0.034743384	0.017199999±0.0036757458	0.16953333±0.01861116	0.49920145±0.018707193	0.1747826±0.036307167	1500.0±0.0
13	**5000**	**0.5156522±0.031231798**	**0.46724564±0.017818842**	**0.48434776±0.031231802**	**0.53275436±0.017818844**	**0.043959998±0.0030343218**	**0.16932±0.0029727637**	**0.4915766±0.017126424**	**0.48434776±0.031231802**	**5000.0±0.0**

Table 5. *Car Evaluation Data Set test results under RCHK (Global MHC) Rule*

r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
3	50	0.5191304±0.08361201	0.61364776±0.23514372	0.48086962±0.08361201	0.38635236±0.23514372	84.67874±18.580166	117.0379±20.995888	0.6118217±0.15022022	0.48086962±0.08361201	35.2±2.7406406
3	150	0.24608696±0.039753493	0.7513648±0.012201222	0.75391304±0.03975348	0.24863525±0.012201216	213.12027±10.482822	230.04013±10.481347	0.49458584±0.038160045	0.75391304±0.03975348	111.0±5.2493386
3	500	0.14869566±0.026707591	0.89578164±0.0	0.85130435±0.026707593	0.104218364±0.0	200.45786±5.9352536	205.78947±5.963186	0.5834547±0.047493592	0.85130435±0.026707593	368.7±5.5186553
3	1500	0.2521739±0.02526898	0.78908193±6.282881E-8	0.7478261±0.025268985	0.2109181±1.5707203E-8	170.44±2.0613496	172.22182±2.0756462	0.5433142±0.025061835	0.7478261±0.025268985	1102.3±17.876429
3	**5000**	**0.4173913±0.038671482**	**0.8138958±6.282881E-8**	**0.5826087±0.038671497**	**0.18610421±1.5707203E-8**	**190.96681±2.5337741**	**191.49844±2.5312166**	**0.69050974±0.019413104**	**0.5826087±0.038671497**	**3660.4±27.407219**
5	50	0.21565218±0.034975316	0.8439206±0.026115589	0.7843479±0.034975313	0.15607938±0.02611557	40.575233±6.4535637	71.84115±7.8994837	0.580101±0.02023211	0.7843479±0.034975313	49.4±0.6992059
5	150	0.19652176±0.026628833	0.7543425±6.282881E-8	0.80347836±0.026628826	0.24565753±3.1414405E-8	58.46186±3.715108	70.28572±3.7058413	0.4427247±0.031756166	0.80347836±0.026628826	148.4±0.96609175
5	500	0.20434782±0.03155295	0.8982631±6.282881E-8	0.7956523±0.03155295	0.10173698±7.853601E-9	74.79267±5.4420023	78.73871±5.4325047	0.6645466±0.033208326	0.7956523±0.03155295	493.8±2.394438
5	1500	0.13913043±0.03330181	0.88585603±6.282881E-8	0.8608696±0.03330181	0.11414392±0.0	76.32446±1.567062	77.731346±1.5794283	0.5421511±0.06115315	0.8608696±0.03330181	1479.2±5.5136194
5	**5000**	**0.18695652±0.025351964**	**0.7940446±6.282881E-8**	**0.8130436±0.025351956**	**0.20595536±1.5707203E-8**	**70.732254±1.0503442**	**71.111725±1.0515862**	**0.47373027±0.036189504**	**0.8130436±0.025351956**	**4931.6±7.80598**
6	50	0.46434784±0.03554715	0.55334985±0.03912945	0.53565216±0.035547156	0.44665012±0.039129447	17.804±5.2073693	37.324±6.207507	0.5098729±0.018941185	0.53565216±0.035547156	50.0±0.0
6	150	0.23565218±0.064090095	0.78808933±0.060952228	0.76434785±0.064090095	0.21191068±0.06095223	30.908±3.5851698	42.385998±3.899064	0.5293514±0.06860177	0.76434785±0.064090095	150.0±0.0
6	500	0.24869564±0.04103267	0.7843672±0.019378515	0.7513043±0.04103267	0.21563277±0.01937851	35.550964±1.4201787	38.991936±1.3572165	0.5332025±0.046470337	0.7513043±0.04103267	499.8±0.42163703
6	1500	0.22608697±0.031486325	0.7925558±0.0031387326	0.773913±0.03148632	0.20744416±0.0031387373	35.546314±0.8531688	36.73696±0.834348	0.5194079±0.032090683	0.773913±0.03148632	1499.6±0.69920594
6	**5000**	**0.19304349±0.040121637**	**0.82133996±0.0**	**0.80695647±0.040121645**	**0.17866005±0.0**	**41.613544±0.70293146**	**41.973175±0.70236254**	**0.5143154±0.052136976**	**0.80695647±0.040121645**	**4997.9±1.1005049**
10	50	0.95739126±0.017123535	0.067741945±0.011931905	0.042608697±0.017123543	0.932258±0.011931891	0.067999996±0.07554249	2.1179998±0.27575353	0.5066322±0.005754671	0.042608697±0.017123543	50.0±0.0
10	150	0.80521744±0.034830876	0.23920596±0.04644023	0.19478261±0.034830883	0.76079404±0.046440236	0.20199999±0.08606917	2.1853333±0.2115154	0.5143868±0.01579484	0.19478261±0.034830883	150.0±0.0
10	500	0.52260864±0.061821405	0.491067±0.035719175	0.47739133±0.061821405	0.508933±0.035719175	0.54800004±0.06809796	2.0128±0.14691253	0.5055878±0.03121478	0.47739133±0.061821405	500.0±0.0
10	1500	0.25652173±0.032600645	0.7692308±0.016166113	0.7434783±0.032600645	0.23076923±0.0161661	1.1761999±0.05903842	2.0384002±0.07190404	0.5250567±0.032152593	0.7434783±0.032600645	1500.0±0.0
10	**5000**	**0.20086959±0.03767001**	**0.88734496±0.017357908**	**0.79913044±0.037670016**	**0.11265508±0.017357906**	**1.8097401±0.07867294**	**2.14048±0.083814576**	**0.6378676±0.0581588**	**0.79913044±0.037670016**	**5000.0±0.0**
13	50	0.9956522±0.0061487616	0.006699752±0.0043845684	0.004347826±0.0061487546	0.99330026±0.0043845647	0.0±0.0	0.172±0.056725457	0.5005891±0.001692027	0.004347826±0.0061487546	50.0±0.0
13	150	0.98±0.016422275	0.020347394±0.006699241	0.02±0.01642228	0.9796526±0.0066992333	0.0026666666±0.0034426518	0.17533334±0.04355215	0.50006235±0.0037003066	0.02±0.01642228	150.0±0.0
13	500	0.9417392±0.02010266	0.055831272±0.008374052	0.05826087±0.020102657	0.94416875±0.008374052	0.0050000004±0.0028674416	0.16279998±0.014823404	0.49931327±0.0051280847	0.05826087±0.020102657	500.0±0.0
13	1500	0.82782614±0.03325132	0.15583126±0.0132237375	0.1721739±0.033251315	0.8441688±0.013223737	0.013866668±0.0027541213	0.15966666±0.010964593	0.49496204±0.010746985	0.1721739±0.033251315	1500.0±0.0
13	**5000**	**0.5269565±0.048189227**	**0.44218364±0.02133931**	**0.4730435±0.048189227**	**0.5578163±0.02133932**	**0.042260002±0.002716289**	**0.16316±0.0046435646**	**0.485006±0.027896196**	**0.4730435±0.048189227**	**5000.0±0.0**

Table 6. *Car Evaluation Data Set test results under RCHK (MHC) Rule*

r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
3	50	1.0±0.0	0.33697274±0.19040091	0.0±0.0	0.6630273±0.19040091	1.1777778±2.141881	23.69969±13.135716	0.6081756±0.06693383	0.0±0.0	17.1±3.1780496
3	150	1.0±0.0	0.669727±0.15380368	0.0±0.0	0.33027297±0.15380368	15.290926±10.740243	36.276398±14.18548	0.76008093±0.080511875	0.0±0.0	53.7±5.9637794
3	500	1.0±0.0	0.78387105±0.016478373	0.0±0.0	0.21612902±0.016478367	28.302176±10.875965	36.99615±11.415833	0.82241267±0.010774263	0.0±0.0	180.6±13.615351
3	1500	1.0±0.0	0.78908193±6.282881E-8	0.0±0.0	0.2109181±1.5707203E-8	33.92102±3.6885273	37.120083±3.6778576	0.82581985±1.2565762E-7	0.0±0.0	548.8±17.319225
3	**5000**	**1.0±0.0**	**0.78908193±6.282881E-8**	**0.0±0.0**	**0.2109181±1.5707203E-8**	**32.78231±1.7324523**	**33.833244±1.7219863**	**0.82581985±1.2565762E-7**	**0.0±0.0**	**1829.4±36.417946**
5	50	1.0±0.0	0.51910675±0.1426016	0.0±0.0	0.4808933±0.14260161	5.526125±2.8074174	22.189848±6.5863657	0.680904±0.06529615	0.0±0.0	44.4±2.7968235
5	150	0.9965218±0.010999221	0.75037223±0.051511005	0.0034782607±0.010999228	0.24962778±0.051511	10.543793±3.572964	20.295738±4.018014	0.8008684±0.031943575	0.0034782607±0.010999228	128.3±4.808557
5	500	0.99739134±0.008249431	0.86873454±0.020510491	0.0026086955±0.00824942	0.1312655±0.020510484	18.482637±1.535586	22.266165±1.6131766	0.8839592±0.016159816	0.0026086955±0.00824942	434.9±6.8223486
5	1500	0.966087±0.026707582	0.9143921±0.019229649	0.033913042±0.026707593	0.08560793±0.019229656	19.938938±1.8637004	21.317707±1.872413	0.91886985±0.016635654	0.033913042±0.026707593	1300.3±13.350407
5	**5000**	**0.9573914±0.031069977**	**0.94317615±0.02361163**	**0.042608697±0.031069975**	**0.056823827±0.023611626**	**21.375202±0.7659717**	**21.805256±0.7687041**	**0.9448045±0.022010047**	**0.042608697±0.031069975**	**4321.7±29.548643**
6	50	0.9913044±0.02459503	0.39727047±0.08277458	0.008695653±0.02459502	0.6027295±0.082774594	3.4858108±2.4067633	14.593817±4.6244574	0.6232509±0.036291003	0.008695653±0.02459502	49.3±1.0593499
6	150	0.97913045±0.014317777	0.705459±0.06397355	0.020869564±0.014317784	0.29454094±0.06397356	6.9157906±2.3304596	14.558668±2.9310434	0.7704673±0.037435643	0.020869564±0.014317784	146.4±2.7968235
6	500	0.92260873±0.04960674	0.905211±0.02413745	0.077391304±0.049606726	0.09478908±0.024137458	11.74383±1.5439423	14.992986±1.6473479	0.907643±0.019667495	0.077391304±0.049606726	489.2±2.529822
6	1500	0.82869565±0.058841296	0.95632744±0.015349893	0.17130435±0.05884129	0.043672454±0.01534989	13.36982±0.87916386	14.557483±0.90283436	0.9503892±0.015446535	0.17130435±0.05884129	1464.8±5.6921
6	**5000**	**0.67565215±0.07435786**	**0.98684865±0.0067247255**	**0.32434782±0.07435786**	**0.013151364±0.006724723**	**14.31147±0.36280334**	**14.682085±0.36298364**	**0.98124427±0.008661475**	**0.32434782±0.07435786**	**4887.9±9.949316**
10	50	0.98086965:0.017773572	0.0662531:0.018644394	0.019130435:0.017773576	0.93374693:0.018644398	0.028:0.019321835	1.626:0.320562	0.5123109:0.007928354	0.019130435:0.017773576	50.0:0.0
10	150	0.94434786±0.021768091	0.16277917±0.047229044	0.055652164±0.021768097	0.83722085±0.04722904	0.10066666±0.080335714	1.5233334±0.3821996	0.5303447±0.0119385645	0.055652164±0.021768097	150.0±0.0
10	500	0.8130435±0.052374814	0.4808933±0.039283015	0.1869565±0.052374814	0.5191066±0.039283004	0.4264±0.10598029	1.7163999±0.20414546	0.6102571±0.023303607	0.1869565±0.052374814	500.0±0.0
10	1500	0.52±0.065304294	0.8434243±0.025666354	0.47999993±0.06530429	0.15657568±0.025666364	0.90139997±0.06195136	1.6992±0.0775513	0.767723±0.033250857	0.47999993±0.06530429	1500.0±0.0
10	**5000**	**0.13652173±0.03758069**	**0.99280393±0.0037813426**	**0.8634783±0.037580684**	**0.007196029±0.0037813499**	**1.3942599±0.04589167**	**1.6879±0.047859453**	**0.9443079±0.036726713**	**0.8634783±0.037580684**	**5000.0±0.0**
13	50	0.99217397±0.006416223	0.007940447±0.0050718645	0.007826087±0.0064162156	0.9920595±0.005071862	0.0019999999±0.0063245553	0.20200002±0.07146095	0.50002706±0.0018899277	0.007826087±0.0064162156	50.0±0.0
13	150	0.98±0.013626262	0.018610422±0.005760293	0.02±0.013626272	0.98138964±0.0057602925	6.6666666E-4±0.002108185	0.14799999±0.018270938	0.49962783±0.0040771035	0.02±0.013626272	150.0±0.0
13	500	0.9330435±0.019680284	0.0617866±0.012293387	0.06695653±0.019680286	0.93821347±0.012293388	0.0066±0.0036575645	0.1698±0.014619621	0.4985872±0.005273877	0.06695653±0.019680286	500.0±0.0

6.2.2 Car Evaluation Data Set Conclusion

The car evaluation data set, re-formatted as a set of binary strings, is the least complex data set, since it comprises only 13 attributes. The results of the best performing test group for each scenario are summarised in Figure 38. The RCHK (MHC) rule has the highest DR value followed by the feature-detection rule (the DR values between the feature-detection rule and the RCHK (MHC) rule differ by 0.01). The feature-detection rule is the best-performing rule, because it has the greatest DR minus FR value. The application of a single global permutation mask RCHK (single global MHC) produced the worst results because it has the lowest DR minus FR value.

A comparison between GC and OC exhibited by the best performing test group for each scenario is summarised in Figure 39. The RCHK (Global MHC) rule has the highest OC and GC values and the feature-detection rule has the lowest OC and GC values. The OC and GC values of the feature-detection rule and the RCHK (MHC) rule are similar. The HD rule is the best performing rule, with reference to generalisation and overfitting, because it has the greatest GC minus OC value. The feature-detection rule is the worst performing rule, because it has the lowest GC minus OC value.

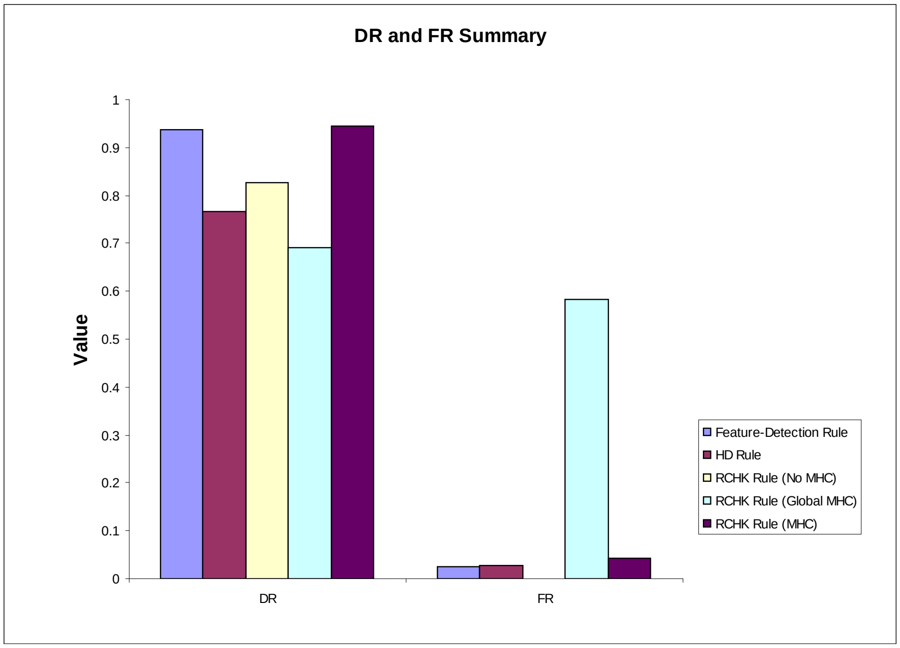


Figure 38. *Car Evaluation Data Set - DR and FR Summary*

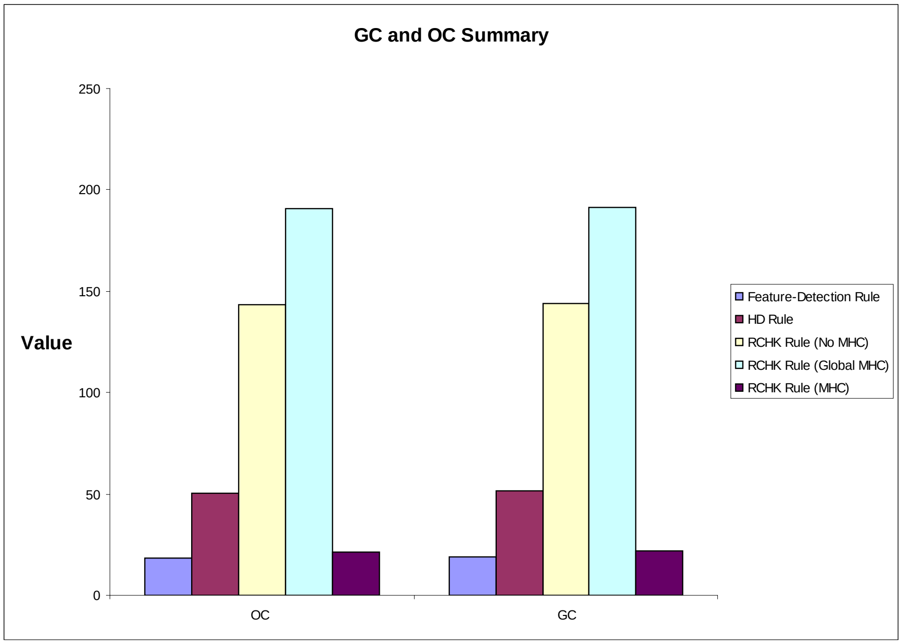


Figure 39. *Car Evaluation Data Set - GC and OC Summary*

6.2.3 Iris Experiment

The iris data set consists of three distinct classes: namely veriscolor, virginica and setosa. The setosa class is linearly separable from the veriscolor and virginica classes, whereas the veriscolor and virginica classes are not linearly separable from one another. Each pattern within the data set comprises four continuous attributes, which was converted into a binary string of length 21 [41]. The data sets were processed further to create a single self-set and non-self set as follows:

- Virginica.self: Contains 50 patterns relating to the virginica class.
- Virginica.non-self: Contains all the patterns related to the veriscolor and setosa classes. The set contains 100 patterns in total.

The results of each scenario are reported as per the list below:

- The results of scenario 1, the iris data set, under the feature-detection rule are tabulated in Table 7 and Table 8.
- The results of scenario 2, the iris data set, under the HD rule are tabulated in Table 9.

- The results of scenario 3, the iris data set, under the RCHK rule with no permutation mark are tabulated in Table 10.

- The results of scenario 4, the iris data set, under the RCHK rule with a single global permutation mask are tabulated in Table 11.

- The results of scenario 5, the iris data set, under the RCHK rule with each detector having its own random permutation mask are tabulated in Table 12.

Table 7. *Iris Data Set under Feature-Detection Rule (Part 1)*

Detector Length	r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
3	2	50	0.97333336±0.036514837	1.0±0.0	0.026666667±0.03651484	0.0±0.0	23.456±2.9529445	26.62±2.9492373	1.0±0.0	0.026666667±0.03651484	50.0±0.0
3	2	150	0.88±0.136626	1.0±0.0	0.120000005±0.13662602	0.0±0.0	27.692001±1.7973523	28.765331±1.8233205	1.0±0.0	0.120000005±0.13662602	150.0±0.0
3	2	500	0.76000005±0.101105005	1.0±0.0	0.24000001±0.101105005	0.0±0.0	28.638±0.3765132	28.965998±0.38916555	1.0±0.0	0.24000001±0.101105005	500.0±0.0
3	2	1500	0.8±0.08164965	1.0±0.0	0.2±0.08164966	0.0±0.0	29.084±0.71462417	29.1852±0.71307117	1.0±0.0	0.2±0.08164966	1500.0±0.0
3	**2**	**5000**	**0.8133334±0.14452988**	**1.0±0.0**	**0.18666667±0.1445299**	**0.0±0.0**	**28.98544±0.4429355**	**29.0168±0.4433243**	**1.0±0.0**	**0.18666667±0.1445299**	**5000.0±0.0**
3	3	50	0.7266667±0.10634209	0.7166667±0.14337209	0.27333337±0.1063421	0.28333336±0.14337209	1.806±1.1346287	3.7419999±1.473151	0.7311622±0.110297926	0.27333337±0.1063421	50.0±0.0
3	3	150	0.5066667±0.233492	0.81333333±0.08635271	0.4933334±0.23349202	0.18666668±0.086352706	2.8713334±0.68007296	3.7160003±0.71504605	0.7226308±0.09954802	0.4933334±0.23349202	150.0±0.0
3	3	500	0.36666667±0.11863421	0.9433333±0.035311654	0.6333334±0.11863421	0.056666665±0.035311665	3.7555995±0.6654341	4.0406±0.6671765	0.8641971±0.0693556	0.6333334±0.11863421	500.0±0.0
3	3	1500	0.24666667±0.077300124	0.95666665±0.022498278	0.75333333±0.07730012	0.043333333±0.022498287	4.128372±0.6369852	4.222845±0.63724047	0.8529293±0.080069505	0.75333333±0.07730012	1499.9±0.31622776
3	**3**	**5000**	**0.19333334±0.08577893**	**0.9700001±0.03314763**	**0.80666673±0.08577893**	**0.030000001±0.033147633**	**4.1348±0.56519085**	**4.1637±0.5659087**	**0.8825758±0.13702841**	**0.80666673±0.08577893**	**5000.0±0.0**
4	3	50	0.9333334±0.08164965	1.0±0.0	0.06666667±0.08164965	0.0±0.0	11.768±3.4825451	14.472±3.6130762	1.0±0.0	0.06666667±0.08164965	50.0±0.0
4	3	150	0.61333334±0.13662602	1.0±0.0	0.38666666±0.13662602	0.0±0.0	14.176±0.40330854	15.086667±0.41856897	1.0±0.0	0.38666666±0.13662602	150.0±0.0
4	3	500	0.6±0.08164967	1.0±0.0	0.4±0.08164967	0.0±0.0	14.988399±0.80000377	15.272±0.78977424	1.0±0.0	0.4±0.08164967	500.0±0.0
4	3	1500	0.5866667±0.11925695	1.0±0.0	0.41333333±0.11925696	0.0±0.0	15.0744±0.5413303	15.165334±0.5451644	1.0±0.0	0.41333333±0.11925696	1500.0±0.0
4	**3**	**5000**	**0.5866667±0.09888265**	**1.0±0.0**	**0.41333333±0.09888265**	**0.0±0.0**	**14.989±0.24474639**	**15.017079±0.24525374**	**1.0±0.0**	**0.41333333±0.09888265**	**5000.0±0.0**
4	4	50	0.7199999±0.15331723	0.63±0.1328231	0.27999997±0.15331724	0.37±0.1328231	1.2980001±0.37976018	3.0740001±0.6209706	0.6645249±0.084139556	0.27999997±0.15331724	50.0±0.0
4	4	150	0.34±0.12746339	0.91±0.07036063	0.66±0.12746339	0.09±0.07036062	2.574±0.7758608	3.3939998±0.8076697	0.8027849±0.14434408	0.66±0.12746339	150.0±0.0
4	4	500	0.120000005±0.075686164	0.99333334±0.014054579	0.88±0.07568616	0.006666667±0.014054568	2.881±0.36610168	3.1372±0.37142393	0.85555553±0.31881985	0.88±0.07568616	500.0±0.0
4	4	1500	0.060000002±0.058373	0.9966667±0.010540934	0.93999994±0.058372993	0.0033333334±0.010540926	3.078467±0.21836653	3.164867±0.21730447	0.6666666±0.47140452	0.93999994±0.058372993	1500.0±0.0
4	**4**	**5000**	**0.04666667±0.044996575**	**1.0±0.0**	**0.9533334±0.044996563**	**0.0±0.0**	**3.0703797±0.13541369**	**3.0962403±0.13556914**	**0.6±0.5163978**	**0.9533334±0.044996563**	**5000.0±0.0**
9	8	50	0.8933333±0.101105	0.22666666±0.092496246	0.10666667±0.10110501	0.7733334±0.092496246	0.031999998±0.017888544	0.33600003±0.08876936	0.53590345±0.0365317	0.10666667±0.10110501	50.0±0.0
9	8	150	0.85333335±0.08692269	0.53999996±0.07226494	0.14666668±0.0869227	0.45999998±0.07226494	0.148±0.08102126	0.504±0.12820123	0.64971054±0.04881381	0.14666668±0.0869227	150.0±0.0
9	8	500	0.6666667±0.08164965	0.9533334±0.029814234	0.33333334±0.08164965	0.04666667±0.029814241	0.36640003±0.07020541	0.57519996±0.071938865	0.935205±0.03854091	0.33333334±0.08164965	500.0±0.0
9	8	1500	0.48000002±0.07302969	1.0±0.0	0.52000004±0.07302969	0.0±0.0	0.49466667±0.04617119	0.5676001±0.04597729	1.0±0.0	0.52000004±0.07302969	1500.0±0.0
9	**8**	**5000**	**0.25333333±0.098882645**	**1.0±0.0**	**0.74666667±0.098882645**	**0.0±0.0**	**0.54592±0.009932363**	**0.56804±0.00993318**	**1.0±0.0**	**0.74666667±0.098882645**	**5000.0±0.0**
9	7	50	0.97333336±0.036514837	0.59333336±0.06411795	0.026666667±0.03651484	0.40666667±0.064117946	0.452±0.2243212	1.6240002±0.30835044	0.70601654±0.03938596	0.026666667±0.03651484	50.0±0.0
9	7	150	0.79999995±0.12472191	0.87333333±0.07958224	0.20000002±0.124721915	0.12666667±0.079582244	1.016±0.2754229	1.6706667±0.31239572	0.86268413±0.089544006	0.20000002±0.124721915	150.0±0.0
9	7	500	0.48000002±0.055777345	0.99333334±0.014907132	0.52±0.055777345	0.006666667±0.014907121	1.2992±0.07346562	1.5195999±0.0747984	0.9846153±0.034401044	0.52±0.055777345	500.0±0.0
9	7	1500	0.30666667±0.1738454	1.0±0.0	0.6933334±0.17384538	0.0±0.0	1.3277334±0.07047236	1.4026667±0.07057068	1.0±0.0	0.6933334±0.17384538	1500.0±0.0
9	**7**	**5000**	**0.2±0.047140457**	**1.0±0.0**	**0.79999995±0.04714045**	**0.0±0.0**	**1.47244±0.054617476**	**1.4952799±0.054817434**	**1.0±0.0**	**0.79999995±0.04714045**	**5000.0±0.0**

Table 8. *Iris Data Set under feature-detection Rule (Part 2)*

Detector Length	r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
9	9	50	0.8933333±0.13770607	0.053333335±0.044996575	0.10666667±0.13770609	0.9466666±0.04499658	0.0060±0.013498971	0.132±0.11360751	0.48263234±0.0421527	0.10666667±0.13770609	50.0±0.0
9	9	150	0.84000003±0.06440611	0.23000002±0.08812169	0.16000001±0.06440612	0.77±0.08812169	0.022000002±0.012976712	0.18466668±0.040006176	0.52255535±0.02262918	0.16000001±0.06440612	150.0±0.0
9	9	500	0.56666666±0.20427528	0.55±0.08050765	0.43333334±0.2042753	0.45000005±0.08050764	0.059600007±0.016701298	0.1884±0.029691752	0.54541814±0.088308245	0.43333334±0.2042753	500.0±0.0
9	9	1500	0.15333335±0.09962895	0.88±0.06126243	0.8466667±0.09962894	0.120000005±0.061262444	0.113199994±0.015519956	0.18059999±0.01628284	0.55523807±0.1574744	0.8466667±0.09962894	1500.0±0.0
9	**9**	**5000**	**0.04±0.05621827**	**1.0±0.0**	**0.96000004±0.056218266**	**0.0±0.0**	**0.15604±0.013125647**	**0.17812±0.013112911**	**0.4±0.5163978**	**0.96000004±0.056218266**	**5000.0±0.0**
14	11	50	0.97333336±0.036514834	0.060000002±0.054772258	0.026666667±0.03651484	0.93999994±0.05477227	0.0±0.0	0.088±0.06723095	0.50889957±0.010744805	0.026666667±0.03651484	50.0±0.0
14	11	150	0.88±0.07302967	0.14666668±0.018257417	0.120000005±0.073029675	0.85333335±0.018257434	0.017333332±0.015347819	0.11866667±0.031411253	0.507027±0.020881211	0.120000005±0.073029675	150.0±0.0
14	11	500	0.7866667±0.14452988	0.47999996±0.13038404	0.21333334±0.1445299	0.52000004±0.13038406	0.026800001±0.0046043466	0.1152±0.010639549	0.6014875±0.10327595	0.21333334±0.1445299	500.0±0.0
14	11	1500	0.56±0.13824295	0.74666667±0.060553007	0.44±0.13824295	0.25333333±0.06055301	0.053199995±0.009734246	0.10866666±0.010402992	0.6841536±0.08319307	0.44±0.13824295	1500.0±0.0
14	**11**	**5000**	**0.25333333±0.0869227**	**1.0±0.0**	**0.7466666±0.08692269**	**0.0±0.0**	**0.097279996±0.0053583556**	**0.11847999±0.005561656**	**1.0±0.0**	**0.7466666±0.08692269**	**5000.0±0.0**
14	14	50	0.99333334±0.02108185	0.0±0.0	0.006666667±0.02108185	1.0±0.0	0.0±0.0	0.0019999999±0.0063245553	0.49827585±0.005452205	0.006666667±0.02108185	50.0±0.0
14	14	150	0.99333334±0.021081852	0.006666667±0.0140545685	0.006666667±0.021081852	0.99333334±0.014054579	0.0±0.0	0.0026666666±0.003442652	0.49997076±0.00699916	0.006666667±0.021081852	150.0±0.0
14	14	500	0.98±0.03220306	0.023333335±0.027442422	0.020000001±0.032203063	0.9766666±0.027442422	0.0±0.0	0.0038000003±0.002394438	0.50081825±0.007498343	0.020000001±0.032203063	500.0±0.0
14	14	1500	0.9466666±0.061262432	0.080000006±0.06885304	0.053333335±0.061262447	0.91999996±0.06885304	4.0000002E-4±4.6613725E-4	0.0070±0.0030991836	0.50727665±0.022279445	0.053333335±0.061262447	1500.0±0.0
14	**14**	**5000**	**0.80666673±0.06629526**	**0.24333334±0.08613799**	**0.19333333±0.066295266**	**0.7566666±0.08613798**	**7.4E-4±5.7387573E-4**	**0.0060600005±0.001469845**	**0.5166026±0.036862854**	**0.19333333±0.066295266**	**5000.0±0.0**
21	8	50	0.5866667±0.19663842	0.52666676±0.14981471	0.41333333±0.19663842	0.47333336±0.1498147	0.796±0.1785497	2.1320002±0.25752673	0.5480799±0.12051104	0.41333333±0.19663842	50.0±0.0
21	8	150	0.14666668±0.055777334	0.9333333±0.04714046	0.85333335±0.055777334	0.06666667±0.047140453	1.7639999±0.29513088	2.4906666±0.3109341	0.7409523±0.16610792	0.85333335±0.055777334	150.0±0.0
21	8	500	0.026666667±0.03651484	1.0±0.0	0.97333336±0.036514837	0.0±0.0	1.9132±0.16959125	2.1467998±0.17089528	0.4±0.5477226	0.97333336±0.036514837	500.0±0.0
21	8	1500	0.026666667±0.036514837	1.0±0.0	0.97333336±0.036514834	0.0±0.0	2.142±0.06629067	2.2182667±0.06913857	0.4±0.5477226	0.97333336±0.036514834	1500.0±0.0
21	**8**	**5000**	**0.053333335±0.029814241**	**1.0±0.0**	**0.9466667±0.029814238**	**0.0±0.0**	**2.24212±0.023489006**	**2.26512±0.023556186**	**0.8±0.44721362**	**0.9466667±0.029814238**	**5000.0±0.0**
21	21	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	50.0±0.0
21	21	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	150.0±0.0
21	21	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	2.0000001E-4±6.324556E-4	0.5±0.0	0.0±0.0	500.0±0.0
21	21	1500	0.99333334±0.021081852	0.0±0.0	0.006666667±0.021081852	1.0±0.0	0.0±0.0	0.0±0.0	0.49827585±0.005452205	0.006666667±0.021081852	1500.0±0.0
21	**21**	**5000**	**0.99333334±0.021081852**	**0.0±0.0**	**0.006666667±0.021081852**	**1.0±0.0**	**0.0±0.0**	**2.0E-5±6.3245556E-5**	**0.49827585±0.0054522054**	**0.006666667±0.021081852**	**5000.0±0.0**

Table 9. *Iris Data Set under HD Rule*

r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
3	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
3	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
3	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
3	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
3	**5000**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.5±0.0**	**0.0±0.0**	**0.0±0.0**
4	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
4	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
4	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
4	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
4	**5000**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.5±0.0**	**0.0±0.0**	**0.0±0.0**
9	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
9	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
9	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
9	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
9	**5000**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.5±0.0**	**0.0±0.0**	**0.0±0.0**
12	50	0.62666667±0.20893615	0.37666667±0.15638667	0.3733333±0.20893617	0.62333333±0.15638667	4.2200003±4.3879447	15.955554±8.033242	0.49514943±0.08008387	0.3733333±0.20893617	4.5±1.95789
12	150	0.39333335±0.15218572	0.79333335±0.14470062	0.6066667±0.15218572	0.20666666±0.14470063	10.85366±3.7228777	18.413082±3.9236448	0.6752944±0.12613614	0.6066667±0.15218572	14.9±4.306326
12	500	0.16666667±0.12668617	0.92333335±0.06858354	0.8333334±0.12668616	0.07666667±0.068583556	13.644594±1.7730017	16.538666±1.9582787	0.6372527±0.24850994	0.8333334±0.12668616	47.1±14.145671
12	1500	0.09333334±0.078252524	0.9833333±0.023570232	0.90666664±0.07825252	0.016666668±0.023570227	14.895282±0.92522585	16.043577±0.9696866	0.81333333±0.31863907	0.90666664±0.07825252	125.5±40.10888
12	**5000**	**0.06666667±0.044444446**	**1.0±0.0**	**0.9333333±0.04444444**	**0.0±0.0**	**15.934755±0.9402727**	**16.207302±0.9451552**	**0.8±0.42163703**	**0.9333333±0.04444444**	**512.7±128.19261**
14	50	0.3333333±0.12957671	0.8466667±0.08344437	0.6666666±0.1295767	0.15333334±0.08344437	3.5579998±1.1240977	5.626±1.2182702	0.67449224±0.18974859	0.6666666±0.1295767	50.0±0.0
14	150	0.10666667±0.08432741	0.97333324±0.043885373	0.8933333±0.0843274	0.026666667±0.043885373	4.1686664±0.6082564	4.964667±0.6169783	0.69666666±0.40565446	0.8933333±0.0843274	150.0±0.0
14	500	0.013333334±0.028109135	1.0±0.0	0.9866667±0.028109133	0.0±0.0	5.057543±0.5611349	5.2941904±0.5634025	0.2±0.42163706	0.9866667±0.028109133	499.9±0.3162278
14	1500	0.033333335±0.035136424	1.0±0.0	0.9666667±0.035136417	0.0±0.0	4.7965164±0.21805303	4.8765945±0.2179148	0.5±0.52704626	0.9666667±0.035136417	1499.8±0.42163697
14	**5000**	**0.026666667±0.03442652**	**1.0±0.0**	**0.97333336±0.034426518**	**0.0±0.0**	**5.070233±0.2551826**	**5.093816±0.25516415**	**0.4±0.51639783**	**0.97333336±0.034426518**	**4999.5±0.70710677**
16	50	0.7466666±0.10327955	0.39333332±0.1654026	0.25333333±0.10327956	0.6066667±0.16540262	0.19399998±0.12580408	1.0200001±0.28142494	0.5564003±0.07604513	0.25333333±0.10327956	50.0±0.0
16	150	0.5333334±0.12957671	0.6733333±0.09787873	0.46666664±0.12957673	0.32666665±0.09787874	0.47199997±0.14833395	1.028±0.17992589	0.6191002±0.09352937	0.46666664±0.12957673	150.0±0.0
16	500	0.13333334±0.054433104	0.9833333±0.028327886	0.8666667±0.054433104	0.01666667±0.028327888	0.83220005±0.07840323	1.0574±0.08016676	0.92166674±0.1300641	0.8666667±0.054433104	500.0±0.0
16	1500	0.013333334±0.028109137	0.9966667±0.010540934	0.9866667±0.028109131	0.0033333334±0.010540926	0.9555333±0.09519314	1.0295999±0.09614763	0.2±0.421637	0.9866667±0.028109131	1500.0±0.0
16	**5000**	**0.020000001±0.032203063**	**1.0±0.0**	**0.98±0.03220306**	**0.0±0.0**	**0.98772±0.029338213**	**1.01028±0.02930942**	**0.3±0.4830459**	**0.98±0.03220306**	**5000.0±0.0**
18	50	0.9533334±0.06324555	0.05±0.0593171	0.04666667±0.063245565	0.95±0.059317093	0.0019999999±0.0063245553	0.077999994±0.053707022	0.50082076±0.024219608	0.04666667±0.063245565	50.0±0.0
18	150	0.9200001±0.075686164	0.09666668±0.086709395	0.08±0.075686164	0.9033333±0.08670938	0.0019999999±0.003220306	0.063999996±0.031301007	0.504859±0.037212886	0.08±0.075686164	150.0±0.0
18	500	0.72±0.13984117	0.31333333±0.09962894	0.28000003±0.13984117	0.6866666±0.099628925	0.0132±0.003910101	0.0828±0.016198765	0.5098301±0.063931726	0.28000003±0.13984117	500.0±0.0
18	1500	0.4666667±0.19372885	0.59±0.11336601	0.53333336±0.19372885	0.41000003±0.113366015	0.02373333±0.0065636486	0.0672±0.009932116	0.5205472±0.06836556	0.53333336±0.19372885	1500.0±0.0
18	**5000**	**0.06666667±0.054433104**	**0.9599999±0.030631224**	**0.9333333±0.054433104**	**0.04±0.030631222**	**0.050619997±0.0071606645**	**0.07156±0.007342903**	**0.53166664±0.32034415**	**0.9333333±0.054433104**	**5000.0±0.0**
21	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	50.0±0.0
21	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	150.0±0.0
21	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	500.0±0.0
21	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	1500.0±0.0
21	**5000**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.5±0.0**	**0.0±0.0**	**5000.0±0.0**

Table 10. *Iris Data Set under RCHK (no MHC) Rule*

r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
3	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
3	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
3	500	0.9866667±0.0421637	0.01±0.03162278	0.013333334±0.042163704	0.98999995±0.031622786	0.38627452±1.2215072	0.4±1.264911	0.49905664±0.0029832784	0.013333334±0.042163704	5.1±16.127617
3	1500	0.9866667±0.0421637	0.01±0.03162278	0.013333334±0.042163704	0.98999995±0.031622786	0.3955414±1.2508117	0.4±1.264911	0.49905664±0.0029832784	0.013333334±0.042163704	15.7±49.647762
3	**5000**	**0.9466667±0.09322744**	**0.09333334±0.22868381**	**0.053333335±0.09322746**	**0.9066666±0.22868383**	**3.2055562±7.520418**	**3.215905±7.5442333**	**0.5214466±0.07455268**	**0.053333335±0.09322746**	**207.6±364.83762**
4	50	0.9000001±0.09558138	0.7133333±0.089166224	0.1±0.0955814	0.28666666±0.08916624	5.153708±1.9535967	7.525148±2.2675576	0.7637254±0.05390075	0.1±0.0955814	35.1±4.1486278
4	150	0.8666667±0.09938079	0.74333334±0.084692605	0.13333334±0.099380806	0.25666666±0.08469262	6.56212±1.0670122	7.77992±1.0568224	0.7761036±0.04961956	0.13333334±0.099380806	108.2±5.7503624
4	500	0.80666673±0.08577892	0.71999997±0.07403701	0.19333334±0.08577893	0.28±0.07403704	7.3547335±1.0858318	7.774891±1.0723898	0.7459282±0.052229002	0.19333334±0.08577893	344.3±24.725382
4	1500	0.71999997±0.10795517	0.78±0.109092765	0.28000003±0.10795518	0.22±0.10909278	7.7531443±1.1783108	7.8930244±1.1752664	0.77924657±0.08524563	0.28000003±0.10795518	1087.7±78.36106
4	**5000**	**0.8066667±0.14555131**	**0.7333333±0.060858052**	**0.19333336±0.14555132**	**0.26666668±0.060858063**	**8.25968±0.75728536**	**8.300746±0.75519514**	**0.752445±0.03165363**	**0.19333336±0.14555132**	**3565.7±223.88739**
9	50	0.90666664±0.056218266	0.21999998±0.11987648	0.093333334±0.056218274	0.78000003±0.119876474	0.09199999±0.055136196	0.592±0.21212156	0.5396224±0.031750735	0.093333334±0.056218274	50.0±0.0
9	150	0.72±0.14673398	0.61±0.19119507	0.28000003±0.14673398	0.39000005±0.19119507	0.21733335±0.10653696	0.6946666±0.18642312	0.6578694±0.12569918	0.28000003±0.14673398	150.0±0.0
9	500	0.38±0.14072126	0.8966667±0.057628006	0.62±0.14072125	0.10333334±0.057628013	0.4276±0.08524371	0.65220004±0.09149353	0.7893232±0.08125502	0.62±0.14072125	500.0±0.0
9	1500	0.120000005±0.042163704	1.0±0.0	0.88±0.0421637	0.0±0.0	0.6321333±0.065427594	0.71646667±0.06610881	1.0±0.0	0.88±0.0421637	1500.0±0.0
9	**5000**	**0.12000002±0.061262444**	**1.0±0.0**	**0.88±0.06126244**	**0.0±0.0**	**0.68981993±0.038678672**	**0.71456003±0.03838424**	**0.9±0.31622776**	**0.88±0.06126244**	**5000.0±0.0**
14	50	0.99333334±0.021081852	0.0033333334±0.010540926	0.006666667±0.021081852	0.9966667±0.010540934	0.0±0.0	0.007999999±0.013984119	0.4991228±0.0027739268	0.006666667±0.021081852	50.0±0.0
14	150	0.98±0.03220306	0.026666667±0.021081852	0.020000001±0.032203063	0.97333324±0.021081857	0.0±0.0	0.024666665±0.020379124	0.50163543±0.010587996	0.020000001±0.032203063	150.0±0.0
14	500	0.9266666±0.049190983	0.08333334±0.036004115	0.07333334±0.049190987	0.9166666±0.036004107	2.0000001E-4±6.324556E-4	0.015599999±0.005796551	0.50256425±0.020780347	0.07333334±0.049190987	500.0±0.0
14	1500	0.82000005±0.054884836	0.21000001±0.07544109	0.18±0.054884847	0.78999996±0.07544109	0.0021333336±9.3227456E-4	0.015866665±0.0036079471	0.5097975±0.031099215	0.18±0.054884847	1500.0±0.0
14	**5000**	**0.5066667±0.10976969**	**0.5733334±0.09913204**	**0.49333334±0.109769695**	**0.42666668±0.09913204**	**0.00552±0.0018766401**	**0.017579999±0.0026439866**	**0.5429275±0.08199566**	**0.49333334±0.109769695**	**5000.0±0.0**
21	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	50.0±0.0
21	150	0.99333334±0.021081852	0.0±0.0	0.006666667±0.021081852	1.0±0.0	0.0±0.0	0.0±0.0	0.49827585±0.005452205	0.006666667±0.021081852	150.0±0.0
21	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	500.0±0.0
21	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	1.3333333E-4±4.21637E-4	0.5±0.0	0.0±0.0	1500.0±0.0
21	**5000**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**	**0.0±0.0**	**2.0E-5±6.3245556E-5**	**0.5±0.0**	**0.0±0.0**	**5000.0±0.0**

Table 11. *Iris Data Set under RCHK (Global MHC) Rule*

r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
3	50	0.93999994±0.18973665	0.06333333±0.20027761	0.060000002±0.18973666	0.93666667±0.20027761	3.148148±9.955318	3.6111112±11.419337	0.5021739±0.006874515	0.060000002±0.18973666	2.7±8.538149
3	150	1.0±0.0	0.026666667±0.08432741	0.0±0.0	0.97333336±0.0843274	1.9227272±6.080197	2.1±6.640783	0.5076923±0.02432521	0.0±0.0	2.2±6.9570107
3	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	11.6±24.495806
3	1500	0.9466667±0.16865481	0.0±0.0	0.053333335±0.16865481	1.0±0.0	0.0±0.0	0.0±0.0	0.4818182±0.057495967	0.053333335±0.16865481	49.0±110.13225
3	**5000**	**0.97333336±0.06440611**	**0.0±0.0**	**0.026666667±0.06440612**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.4927203±0.01780681**	**0.026666667±0.06440612**	**95.6±202.29582**
4	50	0.5066666±0.15137233	0.9066666±0.049190987	0.49333334±0.15137233	0.09333334±0.049190987	7.587088±1.2988973	10.880575±1.4479989	0.8416767±0.082930274	0.49333334±0.15137233	34.5±2.2236106
4	150	0.2±0.08888889	0.6933333±0.12049281	0.8000001±0.088888876	0.30666667±0.120492816	7.336225±0.9326625	8.415715±1.0693196	0.39178795±0.11240698	0.8000001±0.088888876	101.3±9.117139
4	500	0.04666667±0.044996575	0.95333326±0.04216371	0.9533334±0.044996563	0.046666667±0.042163704	7.443494±0.66882974	7.782298±0.7116106	0.41000003±0.4121758	0.9533334±0.044996563	354.6±15.334784
4	1500	0.026666667±0.03442652	1.0±0.0	0.97333336±0.034426514	0.0±0.0	10.78032±0.6496992	10.930087±0.6437873	0.4±0.51639783	0.97333336±0.034426514	1062.0±75.421776
4	**5000**	**0.1±0.0955814**	**1.0±0.0**	**0.9±0.09558138**	**0.0±0.0**	**12.048187±0.88876873**	**12.092896±0.8872982**	**0.7±0.48304588**	**0.9±0.09558138**	**3546.2±318.5749**
9	50	0.7±0.1968894	0.31333336±0.1020772	0.3±0.1968894	0.6866666±0.102077186	0.076000005±0.05399589	0.68±0.2686592	0.49780852±0.10162905	0.3±0.1968894	50.0±0.0
9	150	0.3266667±0.1790234	0.57666665±0.09434635	0.67333335±0.1790234	0.42333332±0.09434635	0.224±0.066161044	0.7133333±0.10217389	0.4123277±0.13530304	0.67333335±0.1790234	150.0±0.0
9	500	0.020000001±0.03220306	0.9599999±0.04388537	0.98±0.032203056	0.040000003±0.043885373	0.45240003±0.10740908	0.69439995±0.12082236	0.16666666±0.28327882	0.98±0.032203056	500.0±0.0
9	1500	0.006666667±0.021081852	1.0±0.0	0.99333334±0.02108185	0.0±0.0	0.6354667±0.055412095	0.71599996±0.055307	0.1±0.31622776	0.99333334±0.02108185	1500.0±0.0
9	**5000**	**0.0±0.0**	**1.0±0.0**	**1.0±0.0**	**0.0±0.0**	**0.62152004±0.03174554**	**0.646±0.031772673**	**0.0±0.0**	**1.0±0.0**	**5000.0±0.0**
14	50	02108185	±0.010540926	0.021081852	10540934	0.0±0.0	7942	027739266	0.021081852	50.0±0.0
14	150	0.97333336±0.046613723	0.033333335±0.015713485	0.026666667±0.04661373	0.9666666±0.015713483	0.0±0.0	0.016666666±0.010540926	0.50147855±0.010032615	0.026666667±0.04661373	150.0±0.0
14	500	0.88666666±0.07062332	0.089999996±0.041722186	0.11333333±0.07062333	0.91±0.041722186	6.0E-4±9.660918E-4	0.019599998±0.007351493	0.49301472±0.02323321	0.11333333±0.07062333	500.0±0.0
14	1500	0.80666673±0.14555131	0.19666669±0.097436264	0.19333336±0.14555132	0.8033334±0.097436264	0.0018666666±0.0018539248	0.016733333±0.0049035135	0.4986294±0.06586316	0.19333336±0.14555132	1500.0±0.0
14	**5000**	**0.42666668±0.24383765**	**0.62666667±0.10036968**	**0.5733334±0.24383764**	**0.37333333±0.10036969**	**0.00526±0.0018974837**	**0.01842±0.0030542503**	**0.49948317±0.14249705**	**0.5733334±0.24383764**	**5000.0±0.0**
21	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0019999999±0.0063245553	0.5±0.0	0.0±0.0	50.0±0.0
21	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	150.0±0.0
21	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	500.0±0.0
21	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	1500.0±0.0
21	**5000**	**1.0±0.0**	**0.0033333334±0.010540926**	**0.0±0.0**	**0.9966667±0.010540934**	**0.0±0.0**	**8.0E-5±1.3984118E-4**	**0.50084746±0.0026799003**	**0.0±0.0**	**5000.0±0.0**

Table 12. *Iris Data Set under RCHK (MHC) Rule*

r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
3	50	0.9333334±0.088888876	0.23333335±0.2888889	0.06666667±0.08888889	0.7666667±0.28888887	0.8486255±0.78088397	3.2239513±2.2262013	0.5672599±0.10510507	0.06666667±0.08888889	15.0±2.6246693
3	150	0.7933333±0.13128048	0.59000003±0.20967346	0.20666668±0.1312805	0.41±0.20967346	2.246284±1.5387622	4.4097605±2.153065	0.6742336±0.10524034	0.20666668±0.1312805	44.5±5.720334
3	500	0.5733334±0.1992579	0.82333326±0.07544109	0.42666668±0.19925788	0.17666666±0.075441085	3.1646423±0.9160805	4.0670805±0.95371896	0.7689044±0.094685756	0.42666668±0.19925788	148.3±12.129395
3	1500	0.34666663±0.10327956	0.89±0.060959406	0.6533333±0.10327955	0.110000014±0.060959414	4.0109186±0.56978536	4.336661±0.5761496	0.7575059±0.10070588	0.6533333±0.10327955	436.7±13.960658
3	**5000**	**0.24000001±0.08999315**	**0.94333327±0.031622775**	**0.75999993±0.08999313**	**0.056666665±0.03162278**	**4.4136825±0.45556828**	**4.513071±0.45379123**	**0.79752415±0.1357313**	**0.75999993±0.08999313**	**1479.8±36.39841**
4	50	0.68000007±0.11243653	0.43666667±0.16810489	0.32000002±0.11243653	0.56333333±0.16810489	1.5104654±0.73549527	3.8262355±1.2850778	0.5532085±0.08873324	0.32000002±0.11243653	29.8±3.224903
4	150	0.52666664±0.16465452	0.8366667±0.08081376	0.47333336±0.16465454	0.16333334±0.080813766	2.8375728±0.74969894	4.2260017±0.77115417	0.76966274±0.091742516	0.47333336±0.16465454	86.9±5.801341
4	500	0.20000002±0.10423146	0.95333326±0.039126266	0.8±0.104231454	0.046666667±0.03912626	4.0771327±0.6690778	4.516092±0.67755663	0.7872439±0.19589897	0.8±0.104231454	294.3±11.16592
4	1500	0.07333334±0.058373004	0.9966667±0.010540934	0.92666674±0.058372997	0.0033333334±0.010540926	4.412161±0.34530318	4.561801±0.3421707	0.68±0.47328636	0.92666674±0.058372997	892.6±28.33608
4	**5000**	**0.060000002±0.037843082**	**1.0±0.0**	**0.93999994±0.03784308**	**0.0±0.0**	**4.295509±0.36027455**	**4.341272±0.35835508**	**0.8±0.421637**	**0.93999994±0.03784308**	**2960.3±73.96103**
9	50	0.7866667±0.13984117	0.21333334±0.07403703	0.21333337±0.13984118	0.78666675±0.074037015	0.066±0.0730601	0.63199997±0.20595577	0.49707794±0.0436068	0.21333337±0.13984118	50.0±0.0
9	150	0.58000004±0.104468085	0.47666663±0.081725225	0.42000002±0.10446808	0.5233334±0.081725225	0.20933333±0.12866646	0.62±0.18520594	0.52502394±0.055108774	0.42000002±0.10446808	150.0±0.0
9	500	0.23333335±0.12272624	0.9166667±0.07412035	0.7666667±0.122726224	0.08333334±0.07412036	0.41500002±0.0828506	0.62659997±0.08716803	0.72602814±0.2007909	0.7666667±0.122726224	500.0±0.0
9	1500	0.04666667±0.044996575	0.9966667±0.010540934	0.9533334±0.044996563	0.0033333334±0.010540926	0.5434±0.045453493	0.6184667±0.046030637	0.58000004±0.5028806	0.9533334±0.044996563	1500.0±0.0
9	**5000**	**0.026666667±0.03442652**	**1.0±0.0**	**0.97333336±0.034426518**	**0.0±0.0**	**0.60067993±0.04480588**	**0.62332±0.045038365**	**0.4±0.5163978**	**0.97333336±0.034426518**	**5000.0±0.0**
14	50	0.98±0.032203056	0.0033333334±0.010540926	0.020000001±0.03220306	0.9966667±0.010540934	0.0±0.0	0.021999998±0.042635404	0.49567452±0.007336217	0.020000001±0.03220306	50.0±0.0
14	150	0.97333336±0.034426518	0.020000001±0.028109137	0.026666667±0.03442652	0.98±0.028109137	6.6666666E-4±0.002108185	0.018666666±0.015331722	0.4982461±0.01059245	0.026666667±0.03442652	150.0±0.0
14	500	0.9466667±0.05258737	0.06000001±0.05397759	0.053333335±0.05258738	0.93999994±0.053977598	2.0000001E-4±6.324556E-4	0.0148±0.0074356496	0.5017945±0.021424664	0.053333335±0.05258738	500.0±0.0
14	1500	0.88±0.07568616	0.19000001±0.09434635	0.12000002±0.075686164	0.81000006±0.09434635	0.0013333333±9.428091E-4	0.014466668±0.0022010095	0.52135813±0.041156746	0.12000002±0.075686164	1500.0±0.0
14	**5000**	**0.5±0.18121675**	**0.62±0.106805466**	**0.5±0.18121675**	**0.38±0.10680545**	**0.0052599995±0.0010156552**	**0.016980002±0.0016040227**	**0.55853224±0.0957422**	**0.5±0.18121675**	**5000.0±0.0**
21	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	50.0±0.0
21	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	150.0±0.0
21	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	500.0±0.0
21	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	1.3333333E-4±4.21637E-4	0.5±0.0	0.0±0.0	1500.0±0.0
21	**5000**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**	**0.0±0.0**	**4.0E-5±8.43274E-5**	**0.5±0.0**	**0.0±0.0**	**5000.0±0.0**

6.2.4 Iris Data Set Conclusion

The iris data set, re-formatted as a set of binary strings, is more complex than the car evaluation data set, because it comprises 21 attributes. The results of the best performing test group for each scenario are summarised in Figure 40. The feature-detection rule is the best performing rule, because its DR minus FR value is the greatest across all of the rules. The HD rule is the worst performing rule.

A comparison between GC and OC exhibited by the best performing test group for each scenario is summarised in Figure 41. The feature-detection rule has the highest OC and GC values followed the HD rule and RCHK (No MHC) rule. The HD rule is the best performing rule, with reference to generalisation and overfitting, because it has the greatest GC minus OC value whereas the RCHK (Global MHC) rule is the worst performing rule because it has the lowest GC minus OC value.

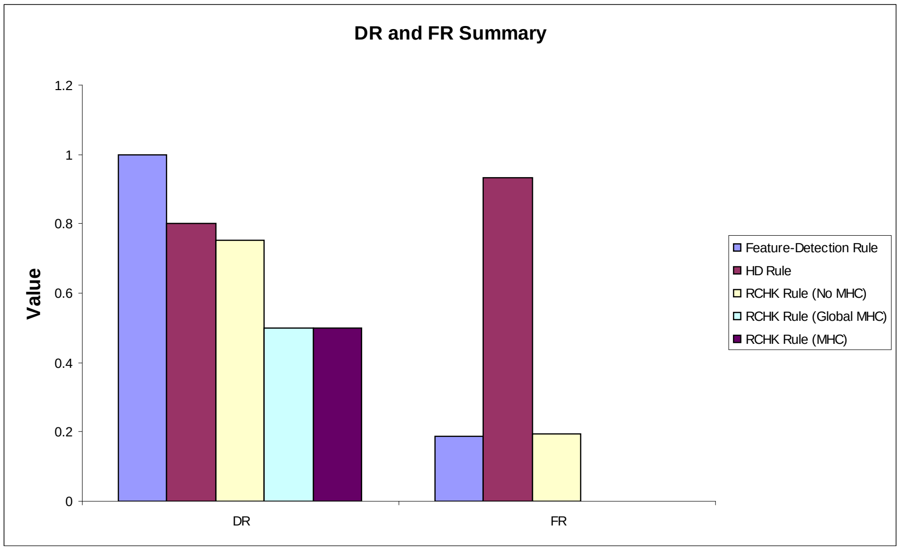


Figure 40. *Iris Data Set: DR and FR Summary*

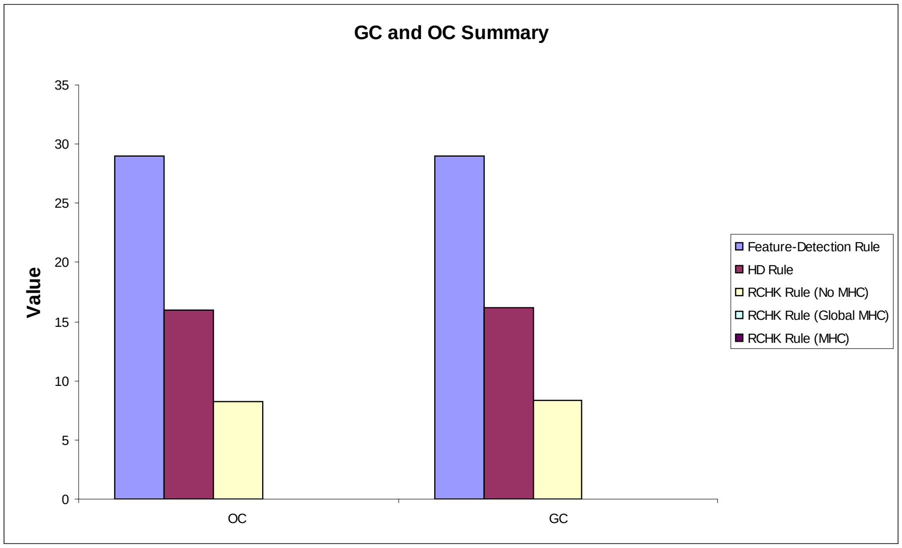


Figure 41. *Iris Data Set: GC and OC Summary*

6.2.5 Wisconsin Breast Cancer Experiment

The Wisconsin breast-cancer data set comprises 699 patterns distributed between two classes, namely, benign and malignant. Each pattern originally consisted of nine attributes with values in the range [0,10]. The tenth attribute indicates the target class of the pattern. The data set has 16 missing values for the "bare nuclei" attributes which were replaced by ones. Each pattern was converted into a bit string of length 37. The data sets were processed further to create a single self-set and non-self set as follows:

- Benign.self: Contains 458 patterns relating to the benign class.
- Benign.non-self: Contains all the patterns related to malignant class. The set contains 241 patterns in total.

The results of each scenario are reported as per the list below:

- The results of scenario 1, the Wisconsin breast-cancer data set, under the feature-detection rule are tabulated in Table 13 and Table 14.
- The results of scenario 2, the Wisconsin breast-cancer data set, under the HD rule are tabulated in Table 15 and Table 16.
- The results of scenario 3, the Wisconsin breast-cancer data set, under the RCHK rule with no permutation mark are tabulated in Table 17.
- The results of scenario 4, the Wisconsin breast-cancer data set, under the RCHK rule with a single global permutation mask are tabulated in Table 18.
- The results of scenario 5, the Wisconsin breast-cancer data set, under the RCHK rule with each detector having its own random permutation mask are tabulated in Table 19.

Table 13. *Wisconsin Breast-Cancer Data Set under Feature-Detection Rule (Part 1)*

Detector Length	r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
3	2	50	0.9708029±0.017879503	0.8694445±0.028800627	0.029197078±0.017879488	0.13055556±0.028800614	29.017227±7.7058196	34.561928±7.9616933	0.8820877±0.02186769	0.029197078±0.017879488	48.4±1.8165902
3	2	150	0.96934307±0.013057338	0.9388889±0.02324054	0.030656934±0.013057331	0.06111111±0.023240557	31.596691±5.342802	33.658703±5.3137665	0.94097614±0.021711493	0.030656934±0.013057331	141.2±4.1472883
3	2	500	0.96934307±0.0061070076	0.98611104±6.664002E-8	0.030656934±0.0061070067	0.01388889±1.0412503E-9	35.80622±4.2527304	36.497047±4.2408233	0.9858738±8.758481E-5	0.030656934±0.0061070067	466.6±19.03418
3	2	1500	0.960584±0.018321024	0.98611104±6.664002E-8	0.039416052±0.018321022	0.01388889±1.0412503E-9	37.084694±5.2265916	37.32355±5.222468	0.9857432±2.688026E-4	0.039416052±0.018321022	1405.6±59.9358
3	2	5000	0.9576642±0.014040426	0.98611104±6.664002E-8	0.042335767±0.014040426	0.01388889±1.0412503E-9	33.20325±1.563629	33.273182±1.5610502	0.98570204±2.0563458E-4	0.042335767±0.014040426	4639.8±131.02174
3	**2**	**50000**	**0.9518248±0.015136083**	**0.98611104±6.664002E-8**	**0.04817518±0.015136089**	**0.01388889±1.0412503E-9**	**35.2795±6.0915318**	**35.286324±6.0915985**	**0.98561513±2.2499077E-4**	**0.04817518±0.015136089**	**47404.2±1296.1891**
3	3	50	0.9715328±0.015936278	0.85±0.04283346	0.028467152±0.015936274	0.15±0.042833455	11.146±2.452954	15.644±2.8116708	0.8676165±0.032362953	0.028467152±0.015936274	50.0±0.0
3	3	150	0.9408759±0.023950806	0.96944445±0.015767956	0.05912409±0.023950802	0.030555556±0.01576795	14.352666±1.5337793	16.170666±1.5415505	0.96873647±0.015719466	0.05912409±0.023950802	150.0±0.0
3	3	500	0.9328467±0.01781314	0.9874999±0.0043920544	0.06715329±0.017813144	0.0125±0.0043920525	14.661±1.0272595	15.225801±1.0317414	0.9868363±0.0046309414	0.06715329±0.017813144	500.0±0.0
3	3	1500	0.90073±0.02338805	0.9944445±0.0071721952	0.099270076±0.02338806	0.0055555557±0.0071721915	15.8824005±1.3319075	16.076±1.3331056	0.9939146±0.007860418	0.099270076±0.02338806	1500.0±0.0
3	3	5000	0.8693431±0.034997612	1.0±0.0	0.13065693±0.034997612	0.0±0.0	15.555921±0.88677573	15.61454±0.8866998	1.0±0.0	0.13065693±0.034997612	5000.0±0.0
3	**3**	**50000**	**0.8540146±0.027093733**	**1.0±0.0**	**0.1459854±0.027093746**	**0.0±0.0**	**15.5381365±1.1239799**	**15.543915±1.1240369**	**1.0±0.0**	**0.1459854±0.027093746**	**50000.0±0.0**
5	4	50	0.96934307±0.0140404245	0.825±0.15945025	0.030656934±0.014040425	0.175±0.15945026	7.104±2.525209	11.268±3.1048222	0.8591341±0.10407931	0.030656934±0.014040425	50.0±0.0
5	4	150	0.94160587±0.030095652	0.95±0.02106353	0.058394156±0.030095663	0.05±0.021063544	10.204±1.1070038	11.944±1.1575319	0.9495287±0.021682972	0.058394156±0.030095663	150.0±0.0
5	4	500	0.9197081±0.031395353	0.9944445±0.007607261	0.08029197±0.03139535	0.0055555557±0.0076072575	11.538±0.3536325	12.092±0.36093745	0.9940616±0.00814396	0.08029197±0.03139535	500.0±0.0
5	4	1500	0.8919708±0.026012449	1.0±0.0	0.1080292±0.026012452	0.0±0.0	12.179067±0.5630346	12.368001±0.5620739	1.0±0.0	0.1080292±0.026012452	1500.0±0.0
5	4	5000	0.82919705±0.023425996	1.0±0.0	0.17080292±0.023425996	0.0±0.0	12.661481±0.45008466	12.717999±0.44971094	1.0±0.0	0.17080292±0.023425996	5000.0±0.0
5	**4**	**50000**	**0.7635037±0.023426006**	**1.0±0.0**	**0.23649636±0.023425996**	**0.0±0.0**	**12.973272±0.39807707**	**12.978897±0.39799342**	**1.0±0.0**	**0.23649636±0.023425996**	**50000.0±0.0**
5	5	50	0.9766423±0.012310574	0.58194447±0.12736951	0.023357663±0.012310569	0.4180556±0.12736951	1.6460001±0.4890853	4.512±0.9621365	0.7054861±0.06436257	0.023357663±0.012310569	50.0±0.0
5	5	150	0.959854±0.015861806	0.9319445±0.034303013	0.040145986±0.015861806	0.068055555±0.034303024	3.4959998±0.5301591	5.07±0.6065425	0.9346663±0.030990727	0.040145986±0.015861806	150.0±0.0
5	5	500	0.9145986±0.020944363	0.9916666±0.009711198	0.08540146±0.02094437	0.008333334±0.009711193	4.3952±0.3078487	4.915±0.3143971	0.9910919±0.010323828	0.08540146±0.02094437	500.0±0.0
5	5	1500	0.8664233±0.03706771	1.0±0.0	0.13357663±0.03706771	0.0±0.0	4.7353334±0.30499715	4.909734±0.30437544	1.0±0.0	0.13357663±0.03706771	1500.0±0.0
5	5	5000	0.8262774±0.026830265	1.0±0.0	0.17372264±0.026830263	0.0±0.0	4.84042±0.18421952	4.89308±0.18468791	1.0±0.0	0.17372264±0.026830263	5000.0±0.0
5	**5**	**50000**	**0.7510948±0.04303889**	**1.0±0.0**	**0.2489051±0.043038882**	**0.0±0.0**	**4.9307923±0.11548407**	**4.9361076±0.11546931**	**1.0±0.0**	**0.2489051±0.043038882**	**50000.0±0.0**
14	13	50	1.0±0.0	0.008333334±0.0076072575	0.0±0.0	0.9916667±0.0076072617	0.0±0.0	0.04±0.02828427	0.50209785±0.0019150946	0.0±0.0	50.0±0.0
14	13	150	1.0±0.0	0.016666668±0.0248452	0.0±0.0	0.9833333±0.024845213	0.0±0.0	0.025333334±0.009888264	0.5042656±0.006377675	0.0±0.0	150.0±0.0
14	13	500	0.98540145±0.008939761	0.05833333±0.038540103	0.014598539±0.008939744	0.9416667±0.038540095	0.0028000001±0.0026832814	0.0392±0.013682106	0.51150656±0.010719277	0.014598539±0.008939744	500.0±0.0
14	13	1500	0.9839417±0.01082657	0.22777776±0.03878956	0.016058395±0.010826567	0.7722222±0.038789555	0.005866667±9.888265E-4	0.0436±0.0063526034	0.5604912±0.010821766	0.016058395±0.010826567	1500.0±0.0
14	13	5000	0.94452554±0.015991908	0.54722226±0.06334307	0.05547445±0.0159919	0.4527778±0.06334309	0.01268±0.0014872794	0.04004±0.0033805324	0.6770841±0.027984152	0.05547445±0.0159919	5000.0±0.0
14	**13**	**50000**	**0.7649635±0.02701716**	**1.0±0.0**	**0.23503649±0.027017161**	**0.0±0.0**	**0.037436±3.4478938E-4**	**0.042320002±3.4380183E-4**	**1.0±0.0**	**0.23503649±0.027017161**	**50000.0±0.0**
14	14	50	1.0±0.0	0.0013888889±0.0043920525	0.0±0.0	0.9986111±0.0043920544	0.0±0.0	0.015999999±0.015776211	0.50034964±0.0011056802	0.0±0.0	50.0±0.0
14	14	150	1.0±0.0	0.011111111±0.012763008	0.0±0.0	0.98888886±0.012763013	0.0±0.0	0.013333334±0.007698004	0.502812±0.0032527386	0.0±0.0	150.0±0.0
14	14	500	0.9970803±0.0051036896	0.04027778±0.034303028	0.002919708±0.005103693	0.9597222±0.034303017	2.0000001E-4±6.324556E-4	0.015999999±0.008164966	0.50968516±0.009822381	0.002919708±0.005103693	500.0±0.0
14	14	1500	0.989781±0.010989383	0.07638889±0.045947764	0.010218977±0.010989382	0.92361104±0.045947764	4.666667E-4±4.4996574E-4	0.013266666±0.0023190042	0.51754797±0.013563929	0.010218977±0.010989382	1500.0±0.0
14	14	5000	0.97737217±0.01898747	0.2638889±0.048555966	0.022627737±0.018987458	0.7361111±0.048555974	0.00168±5.181163E-4	0.013079999±0.001503182	0.57076204±0.018970877	0.022627737±0.018987458	5000.0±0.0
14	**14**	**50000**	**0.8452555±0.023286594**	**0.94027776±0.032769136**	**0.15474454±0.023286588**	**0.059722222±0.032769147**	**0.009864001±7.923972E-4**	**0.014481999±8.5304404E-4**	**0.9347256±0.034048118**	**0.15474454±0.023286588**	**50000.0±0.0**

Table 14. *Wisconsin Breast-Cancer Data Set under Feature-Detection Rule (Part 2)*

Detector Length	r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
18	17	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.007999999±0.010954451	0.5±0.0	0.0±0.0	50.0±0.0
18	17	150	1.0±0.0	0.0055555557±0.0076072575	0.0±0.0	0.9944445±0.007607261	0.0±0.0	0.0026666666±0.0036514837	0.50139856±0.0019150943	0.0±0.0	150.0±0.0
18	17	500	1.0±0.0	0.008333334±0.0124226	0.0±0.0	0.9916667±0.012422605	0.0±0.0	0.0036±0.0032863356	0.50210774±0.003146643	0.0±0.0	500.0±0.0
18	17	1500	0.9956204±0.009792991	0.019444445±0.01583577	0.004379562±0.009792998	0.98055553±0.015835777	0.0±0.0	0.0028000001±8.6922705E-4	0.5038279±0.0025900814	0.004379562±0.009792998	1500.0±0.0
18	17	5000	0.98540145±0.013655684	0.05555556±0.025983732	0.014598539±0.013655684	0.9444445±0.025983721	1.2E-4±1.0954451E-4	0.0034399997±7.9246453E-4	0.5106641±0.00994992	0.014598539±0.013655684	5000.0±0.0
18	**17**	**50000**	**0.9372263±0.02611466**	**0.4333333±0.092858404**	**0.06277372±0.026114661**	**0.56666666±0.09285842**	**7.04E-4±8.6487E-5**	**0.002772±3.4164306E-4**	**0.6253177±0.03591798**	**0.06277372±0.026114661**	**50000.0±0.0**
18	18	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	50.0±0.0
18	18	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0013333333±0.00421637	0.5±0.0	0.0±0.0	150.0±0.0
18	18	500	0.9992701±0.002308224	0.0013888889±0.004392053	7.29927E-4±0.0023082318	0.9986111±0.004392054	0.0±0.0	0.0016000001±0.0029514595	0.50016654±5.2653695E-4	7.29927E-4±0.0023082318	500.0±0.0
18	18	1500	0.9992701±0.0023082239	0.0055555557±0.0071721915	7.29927E-4±0.002308232	0.9944445±0.0071721952	6.666667E-5±2.108185E-4	8.666667E-4±7.062333E-4	0.50121546±0.0016558042	7.29927E-4±0.002308232	1500.0±0.0
18	18	5000	0.99270076±0.010881106	0.020833336±0.011803287	0.007299269±0.010881109	0.9791666±0.011803291	0.0±0.0	7.6E-4±2.270585E-4	0.50343424±0.0039095245	0.007299269±0.010881109	5000.0±0.0
18	**18**	**50000**	**12989157**	**04629629**	**0.012989147**	**46296295**	**5±4.346135E-5**	**4±1.1548931E-**	**13362227**	**0.012989147**	**50000.0±0.0**
37	3	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
37	3	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
37	3	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
37	3	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
37	3	5000	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
37	**3**	**50000**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.5±0.1**	**0.0±0.0**	**0.0±0.0**
37	37	50	1.0:0.0	0.0:0.0	0.0:0.0	1.0:0.0	0.0:0.0	0.0:0.0	0.5:0.0	0.0:0.0	50.0:0.0
37	37	150	1.0:0.0	0.0:0.0	0.0:0.0	1.0:0.0	0.0:0.0	0.0:0.0	0.5:0.0	0.0:0.0	150.0:0.0
37	37	500	1.0:0.0	0.0:0.0	0.0:0.0	1.0:0.0	0.0:0.0	0.0:0.0	0.5:0.0	0.0:0.0	500.0:0.0
37	37	1500	1.0:0.0	0.0:0.0	0.0:0.0	1.0:0.0	0.0:0.0	0.0:0.0	0.5:0.0	0.0:0.0	1500.0:0.0
37	37	5000	1.0:0.0	0.0:0.0	0.0:0.0	1.0:0.0	0.0:0.0	0.0:0.0	0.5:0.0	0.0:0.0	5000.0:0.0
37	37	50000	1.0:0.0	0.0:0.0	0.0:0.0	1.0:0.0	0.0:0.0	0.0:0.0	0.5:0.0	0.0:0.0	50000.0:0.0

Table 15. *Wisconsin Breast-Cancer Data Set under HD Rule (Part 1)*

r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
3	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
3	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
3	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
3	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
3	**5000**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.5±0.0**	**0.0±0.0**	**0.0±0.0**
5	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
5	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
5	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
5	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
5	**5000**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.5±0.0**	**0.0±0.0**	**0.0±0.0**
14	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
14	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
14	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
14	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
14	**5000**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.5±0.0**	**0.0±0.0**	**0.0±0.0**
18	50	0.98905116±0.012043187	0.43055552±0.31454235	0.010948905±0.012043181	0.5694445±0.31454235	11.608334±14.164927	73.566666±49.093334	0.65910876±0.13214317	0.010948905±0.012043181	1.6±1.264911
18	150	0.9773723±0.01630353	0.71944445±0.23257151	0.022627737±0.016303519	0.28055555±0.23257151	36.001987±23.136963	92.34504±23.621826	0.80065584±0.13478164	0.022627737±0.016303519	5.1±2.9230883
18	500	0.95693433±0.020774081	0.8875001±0.10402767	0.04306569±0.020774087	0.1125±0.10402769	62.708324±12.714225	86.8261±8.580937	0.9024786±0.07783978	0.04306569±0.020774087	13.1±4.909175
18	1500	0.92189777±0.034245253	0.9833333±0.019422395	0.07810219±0.034245256	0.016666668±0.019422386	80.431±9.313218	88.387085±7.178845	0.98292387±0.019327555	0.07810219±0.034245256	49.7±25.051281
18	**5000**	**0.86569345±0.046573196**	**0.9930555±0.009820932**	**0.13430658±0.046573184**	**0.006944445±0.009820928**	**90.15043±7.4578876**	**92.872246±6.9625382**	**0.99256194±0.010355811**	**0.13430658±0.046573184**	**143.7±56.249542**
22	50	0.94817513±0.024440145	0.95694447±0.028874926	0.051824816±0.024440145	0.04305556±0.02887494	18.716±3.2323887	23.83±3.4463663	0.9577093±0.02672493	0.051824816±0.024440145	50.0±0.0
22	150	0.8722628±0.03286924	0.99027777±0.011434347	0.12773722±0.03286923	0.009722223±0.011434341	21.584469±1.0862687	23.42234±1.0812299	0.9892748±0.012751911	0.12773722±0.03286923	149.9±0.3162278
22	500	0.82627743±0.023030974	1.0±0.0	0.17372264±0.023030967	0.0±0.0	22.917145±0.9434092	23.476803±0.9440554	1.0±0.0	0.17372264±0.023030967	499.6±0.6992059
22	1500	0.7321168±0.047560535	1.0±0.0	0.2678832±0.047560524	0.0±0.0	23.6563±0.8697591	23.847868±0.86762226	1.0±0.0	0.2678832±0.047560524	1498.7±1.1595018
22	**5000**	**0.6576642±0.0383858**	**1.0±0.0**	**0.34233576±0.038385805**	**0.0±0.0**	**23.767342±0.89334124**	**23.822895±0.89365846**	**1.0±0.0**	**0.34233576±0.038385805**	**4993.2±5.672546**
26	50	0.6576642±0.0383858	1.0±0.0	0.34233576±0.038385805	0.0±0.0	23.767342±0.89334124	23.822895±0.89365846	1.0±0.0	0.34233576±0.038385805	4993.2±5.672546
26	150	0.96131384±0.015407434	0.69166666±0.060998768	0.03868613±0.015407436	0.30833334±0.060998775	0.7593334±0.16630925	1.8620001±0.24480881	0.7587365±0.032504108	0.03868613±0.015407436	150.0±0.0
26	500	0.9036497±0.019403791	0.9652778±0.020961538	0.09635037±0.0194038	0.03472222±0.02096154	1.4814001±0.15939064	1.9610001±0.16888197	0.96331483±0.02192539	0.09635037±0.0194038	500.0±0.0
26	1500	0.8233577±0.026608704	1.0±0.0	0.17664234±0.026608707	0.0±0.0	1.7411333±0.07687481	1.9098666±0.07585284	1.0±0.0	0.17664234±0.026608707	1500.0±0.0
26	**5000**	**0.729927±0.03282418**	**1.0±0.0**	**0.270073±0.032824177**	**0.0±0.0**	**1.8144401±0.05800046**	**1.8649±0.05805448**	**1.0±0.0**	**0.270073±0.032824177**	**5000.0±0.0**

__Table 16.__ Wisconsin Breast-Cancer Data Set under HD Rule (Part 2)

r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
28	50	0.9948905±0.0035258704	0.047222223±0.014930108	0.005109489±0.0035258823	0.95277774±0.014930094	0.007999999±0.013984118	0.282±0.08189424	0.51083684±0.0035931314	0.005109489±0.0035258823	50.0±0.0
28	150	0.99051094±0.007732479	0.15555558±0.048023343	0.00948905±0.007732481	0.84444445±0.048023347	0.019333333±0.01312805	0.27±0.04380089	0.54013336±0.015325987	0.00948905±0.007732481	150.0±0.0
28	500	0.9656935±0.019173624	0.43888888±0.048644166	0.034306567±0.019173613	0.5611111±0.048644166	0.058399998±0.017833801	0.262±0.036636356	0.63295716±0.022740938	0.034306567±0.019173613	500.0±0.0
28	1500	0.89854014±0.020487143	0.8347222±0.038428716	0.101459846±0.020487137	0.16527776±0.038428713	0.14926668±0.015655078	0.28353333±0.017919024	0.84540176±0.032295153	0.101459846±0.020487137	1500.0±0.0
28	**5000**	**0.8080292±0.031168256**	**0.99027777±0.011434347**	**0.1919708±0.031168254**	**0.009722223±0.011434341**	**0.23112002±0.008499516**	**0.28023997±0.008756865**	**0.9883336±0.013566038**	**0.1919708±0.031168254**	**5000.0±0.0**
30	50	1.0±0.0	0.0027777778±0.0058560697	0.0±0.0	0.9972221±0.0058560725	0.0±0.0	0.011999999±0.021499354	0.5006993±0.0014742404	0.0±0.0	50.0±0.0
30	150	0.9985401±0.0030776318	0.018055556±0.0067089703	0.001459854±0.0030776423	0.98194444±0.006708974	6.6666666E-4±0.002108185	0.022±0.011779875	0.50419396±0.0012286879	0.001459854±0.0030776423	150.0±0.0
30	500	0.99416053±0.0057577416	0.031944446±0.0185763	0.005839416±0.0057577416	0.9680556±0.018576294	8.0000004E-4±0.0010327956	0.025±0.0058309515	0.5066894±0.00465087	0.005839416±0.0057577416	500.0±0.0
30	1500	0.98686135±0.010207389	0.13611111±0.029860212	0.013138684±0.010207385	0.8638889±0.02986023	0.0014000001±0.0014211802	0.021466667±0.0038206715	0.5333325±0.00885171	0.013138684±0.010207385	1500.0±0.0
30	**5000**	**0.94817525±0.02562264**	**0.3611111±0.04293342**	**0.05182482±0.025622644**	**0.6388889±0.04293344**	**0.00468±0.0010921945**	**0.02336±0.0014261055**	**0.597753±0.016950887**	**0.05182482±0.025622644**	**5000.0±0.0**
37	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	50.0±0.0
37	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	150.0±0.0
37	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	500.0±0.0
37	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	1500.0±0.0
37	**5000**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.5±0.0**	**0.0±0.0**	**5000.0±0.0**

Table 17. *Wisconsin Breast-Cancer Data Set under RCHK (No MHC) Rule*

r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
3	50	0.9883211±0.0092329355	0.1138889±0.12893333	0.011678832±0.009232928	0.88611114±0.12893331	8.786622±3.9008367	12.724629±4.063304	0.5299871±0.04185495	0.011678832±0.009232928	19.0±7.9162283
3	150	0.9868612±0.008286815	0.1638889±0.15683447	0.013138686±0.008286806	0.83611107±0.15683445	10.367294±2.358778	11.872304±2.3252962	0.54583573±0.055955227	0.013138686±0.008286806	58.3±13.597794
3	500	0.9810219±0.012501452	0.32222217±0.20146072	0.0189781±0.012501444	0.6777778±0.20146072	12.415714±3.7671154	13.068098±3.6763296	0.599631±0.07211803	0.0189781±0.012501444	207.7±52.616116
3	1500	0.9883212±0.012501454	0.30833334±0.17133382	0.011678832±0.012501442	0.69166666±0.17133382	10.416087±2.389655	10.666994±2.3777726	0.5940817±0.06037655	0.011678832±0.012501442	501.2±170.53433
3	**5000**	**0.9832117±0.009762733**	**0.41944447±0.17007825**	**0.01678832±0.009762727**	**0.58055556±0.17007825**	**11.560615±2.8710377**	**11.6588955±2.8640175**	**0.63526887±0.06504298**	**0.01678832±0.009762727**	**1722.0±456.7197**
5	50	0.9846716±0.012141096	0.6458333±0.112310275	0.015328467±0.012141095	0.35416666±0.112310275	5.313056±2.2863677	9.190568±2.5173998	0.74042714±0.06249942	0.015328467±0.012141095	46.6±1.9550505
5	150	0.9744526±0.022889167	0.76805556±0.104766786	0.025547441±0.022889158	0.23194447±0.104766786	7.799165±1.689095	9.5139885±1.8663	0.8144081±0.071487345	0.025547441±0.022889158	137.2±4.3410187
5	500	0.95839417±0.03154583	0.8819445±0.047777243	0.041605838±0.031545836	0.11805556±0.047777243	8.9524±0.5686272	9.549214±0.5812519	0.8920549±0.040323894	0.041605838±0.031545836	459.6±7.862711
5	1500	0.9591241±0.023639817	0.925±0.04202521	0.04087591±0.023639819	0.075±0.042025205	8.781031±0.7983001	8.989993±0.7996678	0.92851937±0.038420115	0.04087591±0.023639819	1378.1±20.572096
5	**5000**	**0.9510949±0.012430195**	**0.9652778±0.021960251**	**0.048905108±0.012430209**	**0.034722224±0.021960262**	**9.557141±0.78123945**	**9.623947±0.7830472**	**0.9651927±0.021337688**	**0.048905108±0.012430209**	**4642.7±47.185333**
14	50	0.9978102±0.00352587	0.011111111±0.0109557025	0.002189781±0.0035258825	0.98888886±0.010955708	0.0019999999±0.0063245553	0.11±0.05754226	0.5022576±0.0030105037	0.002189781±0.0035258825	50.0±0.0
14	150	0.9963504±0.0038470398	0.061111115±0.030848762	0.0036496348±0.0038470533	0.93888885±0.030848747	0.0013333333±0.0028109136	0.086666666±0.033259176	0.5149621±0.00863406	0.0036496348±0.0038470533	150.0±0.0
14	500	0.9832117±0.0077324784	0.18333332±0.032605212	0.01678832±0.0077324808	0.8166667±0.03260521	0.0074±0.0044271885	0.09279999±0.018310895	0.5464176±0.010023312	0.01678832±0.0077324808	500.0±0.0
14	1500	0.95839405±0.014208085	0.43472224±0.081262864	0.041605838±0.014208079	0.5652777±0.081262864	0.0194±0.0057816226	0.08960001±0.014027135	0.6305272±0.034860805	0.041605838±0.014208079	1500.0±0.0
14	**5000**	**0.9±0.024824671**	**0.8666666±0.03286711**	**0.1±0.024824671**	**0.13333334±0.03286711**	**0.04914±0.0053300406**	**0.09176±0.006342661**	**0.87155664±0.028503396**	**0.1±0.024824671**	**5000.0±0.0**
18	50	1.0±0.0	0.0013888889±0.0043920525	0.0±0.0	0.9986111±0.0043920544	0.0±0.0	0.0060±0.01349897	0.50034964±0.0011056802	0.0±0.0	50.0±0.0
18	150	1.0±0.0	0.0027777778±0.0058560693	0.0±0.0	0.99722224±0.005856072	0.0±0.0	0.0060±0.0049190987	0.5006993±0.0014742404	0.0±0.0	150.0±0.0
18	500	1.0±0.0	0.006944445±0.009820928	0.0±0.0	0.9930555±0.009820932	0.0±0.0	0.0034000003±0.0025033313	0.50175315±0.0024840164	0.0±0.0	500.0±0.0
18	1500	0.9956204±0.0061552855	0.034722224±0.023832478	0.004379562±0.0061552846	0.9652778±0.02383248	6.666667E-5±2.108185E-4	0.004933334±0.0014124664	0.5077998±0.0062344843	0.004379562±0.0061552846	1500.0±0.0
18	**5000**	**0.99051094±0.009136252**	**0.10555556±0.042025205**	**0.00948905±0.009136246**	**0.89444447±0.04202521**	**1.3999999E-4±1.3498972E-4**	**0.0051999995±0.0012400716**	**0.5257064±0.01353999**	**0.00948905±0.009136246**	**5000.0±0.0**
37	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	50.0±0.0
37	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	150.0±0.0
37	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	500.0±0.0

Table 18. *Wisconsin Breast-Cancer Data Set under RCHK (Global MHC) Rule*

r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
3	50	0.34452555±0.14144605	0.7916666±0.10164119	0.6554745±0.14144604	0.20833333±0.10164119	37.508404±10.308839	51.853035±9.408149	0.6281225±0.099770464	0.6554745±0.14144604	21.2±3.3598943
3	150	0.013868613±0.036325634	0.9666667±0.019858908	0.98613137±0.036325637	0.033333335±0.019858899	62.203743±8.515221	68.65662±8.608078	0.1209961±0.21418992	0.98613137±0.036325637	60.1±9.374315
3	500	0.14525548±0.12747395	0.5902778±0.08391003	0.85474455±0.12747395	0.40972224±0.08391004	32.66379±7.835868	34.095337±7.7034526	0.21932395±0.16015738	0.85474455±0.12747395	174.6±39.732998
3	1500	0.002919708±0.005103693	0.85555553±0.081038356	0.9970803±0.00510369	0.14444444±0.081038356	40.80667±6.305399	41.436058±6.1615705	0.021689992±0.03930594	0.9970803±0.00510369	596.8±200.83315
3	**5000**	**0.013868612±0.007258605**	**0.9263889±0.02272765**	**0.9861315±0.007258618**	**0.07361111±0.022727663**	**53.688915±2.829769**	**53.923847±2.8149567**	**0.16305034±0.08212627**	**0.9861315±0.007258618**	**1647.8±297.90036**
5	50	0.66934305±0.14095342	0.9833334±0.026835881	0.33065695±0.14095342	0.016666668±0.026835881	22.304754±4.853672	28.921024±5.0419693	0.97988766±0.029927792	0.33065695±0.14095342	46.5±1.433721
5	150	0.004379562±0.011515484	0.9958333±0.009374292	0.9956204±0.011515479	0.004166667±0.009374287	26.84968±4.0724077	29.24116±4.1166124	0.17243461±0.3692846	0.9956204±0.011515479	138.2±2.8205595
5	500	0.0±0.0	1.0±0.0	1.0±0.0	0.0±0.0	25.405043±1.0491663	26.110447±1.0562929	0.0±0.0	1.0±0.0	465.9±9.58529
5	1500	0.0±0.0	1.0±0.0	1.0±0.0	0.0±0.0	28.801758±0.93504375	29.029074±0.9400669	0.0±0.0	1.0±0.0	1387.3±36.69408
5	**5000**	**0.0±0.0**	**1.0±0.0**	**1.0±0.0**	**0.0±0.0**	**30.356512±0.9174381**	**30.427536±0.91827047**	**0.0±0.0**	**1.0±0.0**	**4614.9±66.395195**
14	50	0.9963504±0.0051613594	0.029166669±0.015284791	0.0036496348±0.0051613636	0.9708334±0.015284798	0.0±0.0	0.12599999±0.049933285	0.5065101±0.004056134	0.0036496348±0.0051613636	50.0±0.0
14	150	0.96861315±0.030593148	0.04444444±0.012763008	0.03138686±0.030593144	0.95555556±0.012763014	0.0019999999±0.004499657	0.082±0.031119047	0.5032977±0.006792919	0.03138686±0.030593144	150.0±0.0
14	500	0.81094885±0.109971896	0.16944446±0.032605216	0.18905109±0.1099719	0.83055556±0.03260521	0.0088±0.0041311826	0.0924±0.013525286	0.49211898±0.037276816	0.18905109±0.1099719	500.0±0.0
14	1500	0.5613139±0.19758879	0.41805553±0.057619825	0.43868613±0.19758877	0.58194447±0.05761983	0.022466669±0.0061987657	0.09586666±0.012983561	0.47905588±0.10482505	0.43868613±0.19758877	1500.0±0.0
14	**5000**	**0.053284667±0.04927233**	**0.89027774±0.04604097**	**0.94671524±0.049272332**	**0.109722234±0.046040967**	**0.05664±0.0071608815**	**0.10054±0.008098314**	**0.29410192±0.12138563**	**0.94671524±0.049272332**	**5000.0±0.0**
18	50	0.97372264±0.083096355	0.0±0.0	0.026277373±0.08309635	1.0±0.0	0.0±0.0	0.0039999997±0.00843274	0.49243698±0.023916384	0.026277373±0.08309635	50.0±0.0
18	150	0.9992701±0.0023082239	0.0013888889±0.0043920525	7.29927E-4±0.0023082318	0.9986111±0.0043920544	0.0±0.0	0.0046666665±0.0063245557	0.50016654±0.001303946	7.29927E-4±0.0023082318	150.0±0.0
18	500	0.9978102±0.0035258704	0.019444447±0.013417942	0.002189781±0.0035258823	0.98055553±0.013417948	0.0±0.0	0.0058000004±0.0027406407	0.50438035±0.0031421261	0.002189781±0.0035258823	500.0±0.0
18	1500	0.98394156±0.023030967	0.04027778±0.023101805	0.016058395±0.023030967	0.9597222±0.023101805	6.666667E-5±2.108185E-4	0.004866667±0.0015088216	0.5062312±0.007721622	0.016058395±0.023030967	1500.0±0.0
18	**5000**	**0.9131387±0.10697022**	**0.083333336±0.020704333**	**0.08686131±0.10697021**	**0.9166666±0.020704335**	**4.0000002E-4±3.126944E-4**	**0.0054±9.285593E-4**	**0.49730983±0.031519603**	**0.08686131±0.10697021**	**5000.0±0.0**
37	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	50.0±0.0
37	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	150.0±0.0
37	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	500.0±0.0
37	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	1500.0±0.0
37	**5000**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.5±0.0**	**0.0±0.0**	**5000.0±0.0**

Table 19. *Wisconsin Breast-Cancer Data set under RCHK (MHC) Rule*

r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
3	50	0.9817518±0.013870756	0.6180555±0.12426913	0.018248174±0.013870747	0.38194442±0.12426912	9.412839±4.1763096	21.992273±5.1671844	0.7247017±0.057579037	0.018248174±0.013870747	12.8±4.4671645
3	150	0.97883224±0.0116433045	0.84305555±0.08930505	0.021167884±0.011643294	0.15694445±0.089305066	15.51458±4.607574	21.387207±5.1116986	0.8666981±0.06407955	0.021167884±0.011643294	37.8±3.9665267
3	500	0.9423358±0.016303515	0.9708333±0.012161048	0.057664234±0.016303519	0.029166669±0.012161043	21.543339±2.6757593	23.62818±2.64606	0.9701873±0.011966607	0.057664234±0.016303519	137.6±16.187788
3	1500	0.92116785±0.024037156	0.99027765±0.0067089736	0.07883212±0.024037156	0.009722223±0.0067089708	24.10754±1.6637292	24.82481±1.6101668	0.9895439±0.0072238613	0.07883212±0.024037156	411.1±34.31051
3	**5000**	**0.90583944±0.036488242**	**0.9944445±0.0071721952**	**0.09416058±0.036488235**	**0.0055555557±0.0071721915**	**22.730402±0.9328272**	**22.959623±0.9189715**	**0.9939731±0.0077868267**	**0.09416058±0.036488235**	**1303.5±90.12738**
5	50	0.9547445±0.021978766	0.8291667±0.097343385	0.045255475±0.021978762	0.17083333±0.0973434	7.6729364±2.956192	12.561213±3.3966663	0.8540083±0.07135807	0.045255475±0.021978762	43.3±3.3349996
5	150	0.9277371±0.020774087	0.9763888±0.011434347	0.07226278±0.020774087	0.023611112±0.011434343	12.133601±0.77661574	14.263132±0.7626694	0.97538203±0.011343002	0.07226278±0.020774087	125.1±6.2972655
5	500	0.8795621±0.025109213	0.99722224±0.005856072	0.12043794±0.025109205	0.0027777778±0.0058560697	13.785373±0.9514382	14.455777±0.94006026	0.99698555±0.006360116	0.12043794±0.025109205	410.9±14.471811
5	1500	0.84306574±0.042177465	1.0±0.0	0.15693429±0.04217746	0.0±0.0	13.981442±0.7319865	14.204378±0.72726196	1.0±0.0	0.15693429±0.04217746	1231.8±23.117094
5	**5000**	**0.79708034±0.036706623**	**1.0±0.0**	**0.20291972±0.036706615**	**0.0±0.0**	**14.006597±0.88469905**	**14.074193±0.88474834**	**1.0±0.0**	**0.20291972±0.036706615**	**4095.3±71.3007**
14	50	0.9963503±0.0051613594	0.011111111±0.0127630085	0.0036496348±0.005161363	0.9888889±0.012763014	0.0±0.0	0.07400001±0.0525357	0.50189507±0.0032880658	0.0036496348±0.005161363	50.0±0.0
14	150	0.9956204±0.005103688	0.041666668±0.029280351	0.004379562±0.005103693	0.9583334±0.02928034	0.0026666666±0.003442652	0.07733333±0.02355975	0.5096443±0.007088245	0.004379562±0.005103693	150.0±0.0
14	500	0.9781022±0.01141221	0.16944444±0.06984453	0.02189781±0.011412203	0.8305556±0.06984452	0.0092±0.008651269	0.0912±0.023346664	0.5415222±0.022097196	0.02189781±0.011412203	500.0±0.0
14	1500	0.94014597±0.02523854	0.41111112±0.04908281	0.059854012±0.025238546	0.58888894±0.049082812	0.0196±0.0059146187	0.08933333±0.011983528	0.61529434±0.022316841	0.059854012±0.025238546	1500.0±0.0
14	**5000**	**0.8948905±0.023888936**	**0.8458334±0.033029735**	**0.10510949±0.02388893**	**0.15416667±0.03302974**	**0.0514±0.004398989**	**0.0923±0.0054145902**	**0.8538004±0.026330443**	**0.10510949±0.02388893**	**5000.0±0.0**
18	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0060±0.009660917	0.5±0.0	0.0±0.0	50.0±0.0
18	150	1.0±0.0	0.006944445±0.0073200874	0.0±0.0	0.9930555±0.0073200907	0.0±0.0	0.0046666665±0.004499657	0.50174826±0.0018428005	0.0±0.0	150.0±0.0
18	500	0.9978102±0.00352587	0.011111111±0.010955703	0.002189781±0.0035258823	0.98888886±0.010955709	2.0000001E-4±6.324556E-4	0.0066±0.004526465	0.50225765±0.0030072092	0.002189781±0.0035258823	500.0±0.0
18	1500	0.9963504±0.0051613594	0.031944446±0.017384244	0.0036496348±0.0051613636	0.96805555±0.017384239	1.9999998E-4±3.2203057E-4	0.0057333335±0.0019925789	0.50723475±0.004785661	0.0036496348±0.0051613636	1500.0±0.0
18	**5000**	**0.9912408±0.011819917**	**0.08333333±0.04088778**	**0.008759124±0.01181991**	**0.9166666±0.040887777**	**2.0000001E-4±2.1081851E-4**	**0.00454±0.0010751744**	**0.519743±0.012755031**	**0.008759124±0.01181991**	**5000.0±0.0**
37	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	50.0±0.0
37	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	150.0±0.0
37	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	500.0±0.0
37	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	1500.0±0.0

6.2.6 Wisconsin Breast-cancer Data Set Conclusion

The results of the best performing test group for each scenario are summarised in Figure 42. The feature-detection rule is the best performing rule because its DR minus FR value is the greatest across all of the rules. The RCHK (MHC) rule has a slightly higher DR than the feature-detection rule but a much greater FR than the feature-detection rule. The RCHK (Global MHC) rule is the worst performing rule.

A comparison between GC and OC exhibited by the best performing test group for each scenario is summarised in Figure 43. The HD rule has the greatest GC and OC values whereas the RCHK (Global MHC) rule has the lowest GC and OC values across all of the rules. The HD rule is the best performing rule, with reference to generalisation and overfitting, because it has the greatest GC minus OC value. The RCK (Global MHC) rule, followed by the feature-detection rule, is the worst performing rule because it has the lowest GC minus OC value.

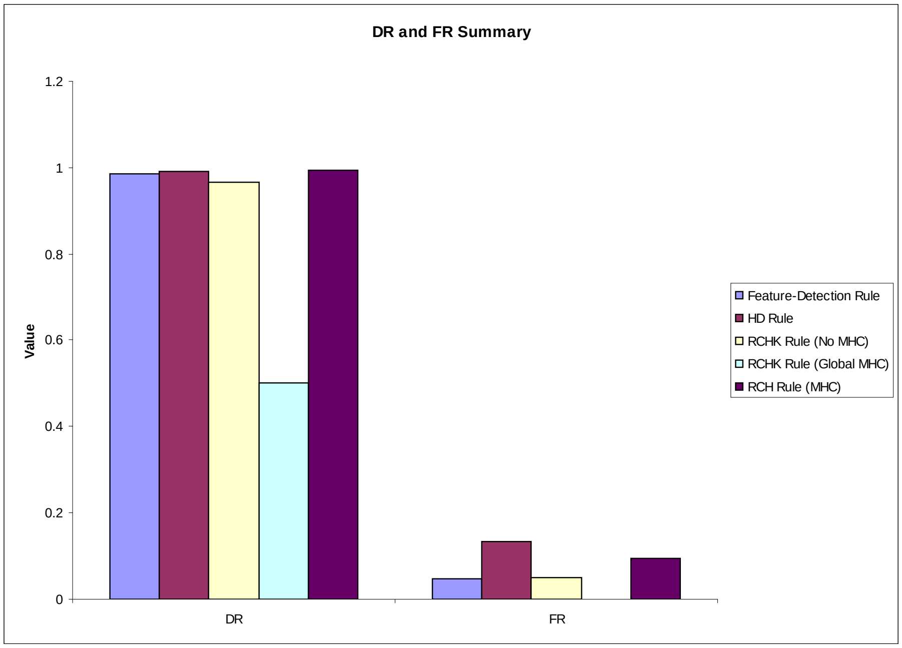


Figure 42. *Wisconsin Breast-Cancer Data Set - DR and FR Summary*

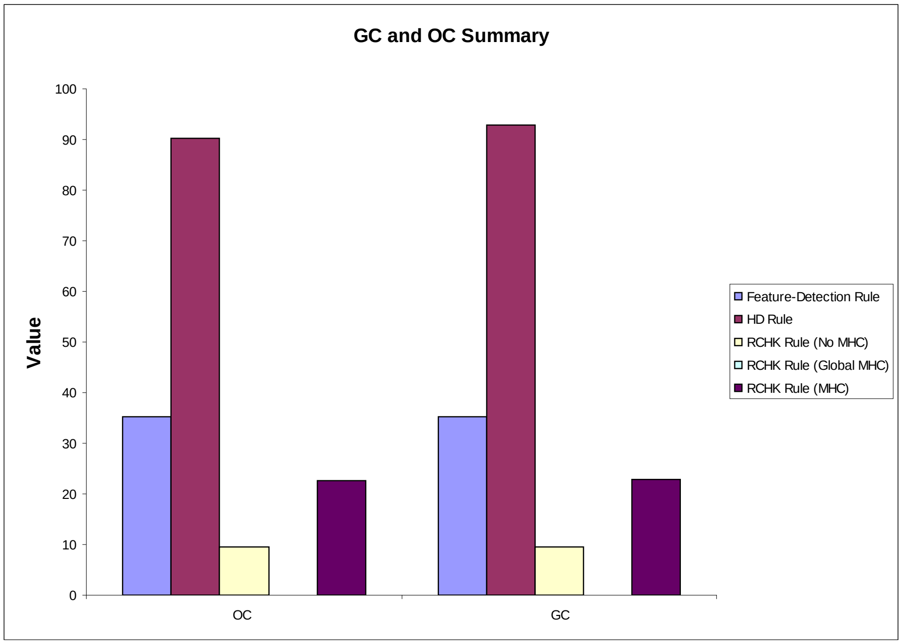


Figure 43. *Wisconsin Breast-Cancer Data Set - GC and OC Summary*

6.2.7 Glass Experiment

The glass data set comprises 214 patterns, distributed between 7 classes, each class being a specific glass type. Each pattern, originally consisted of nine continuous-valued attributes, was converted into a binary string of length 46 [41]. No patterns were recorded in the vehicle_windows_non_float class subset and, subsequently, this subset was excluded from the other data sets. The data sets were processed further to create a single self-set and non-self set as follows:

- Building_window_float.self: Contains 70 patterns relating to the building_window_float class.
- Building_window_float.non-self: Contains all the patterns related to the building_window_non-float, containers, headlamps, tableware and vehicle_window_float classes. The set contains 144 patterns in total.

The results of each scenario are reported as per the list below:

- The results of scenario 1, the glass data set, under the feature-detection rule are tabulated in Table 20 and Table 21.

- The results of scenario 2, the glass data set, under the HD rule are tabulated in Table 22 and Table 23.
- The results of scenario 3, the glass data set, under the RCHK rule with no permutation mark are tabulated in Table 24.
- The results of scenario 4, the glass data set, under the RCHK rule with a single global permutation mask are tabulated in Table 25.
- The results of scenario 5, the glass data set, under the RCHK rule with each detector having its own random permutation mask are tabulated in Table 26.

Table 20. *Glass Data Set under Feature-Detection Rule (Part 1)*

Detector Length	r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
3	2	50	0.9333333±0.054294065	0.5116279±0.04651164	0.06666667±0.054294072	0.4883721±0.046511635	3.712±0.9571415	5.084±0.97554094	0.65668607±0.0228702	0.06666667±0.054294072	50.0±0.0
3	2	150	0.9047619±0.058321178	0.6465117±0.05303142	0.0952381±0.058321185	0.35348836±0.053031407	3.836±0.35666987	4.508±0.34239027	0.71984905±0.028838875	0.0952381±0.058321185	150.0±0.0
3	2	500	0.82857144±0.07221786	0.7069768±0.060643747	0.17142858±0.07221787	0.2930233±0.060643744	4.158±0.33748484	4.3996±0.3476216	0.7388736±0.048520356	0.17142858±0.07221787	500.0±0.0
3	2	1500	0.75238097±0.10858813	0.73023254±0.02651572	0.24761906±0.10858814	0.26976746±0.026515706	4.5105333±0.10322294	4.605467±0.109984346	0.73434436±0.027510002	0.24761906±0.10858814	1500.0±0.0
3	2	5000	0.71428573±0.22080044	0.7767442±0.0265157	0.2857143±0.22080046	0.22325583±0.026515711	4.4102855±0.22387145	4.440127±0.22473171	0.75313336±0.04369405	0.2857143±0.22080046	4999.8±0.44721365
3	**2**	**50000**	**0.64761907±0.14521858**	**0.7627907±0.03032186**	**0.35238096±0.14521858**	**0.23720929±0.030321872**	**4.431445±0.17084321**	**4.4343486±0.17080075**	**0.72819775±0.03434478**	**0.35238096±0.14521858**	**49998.2±1.3038408**
3	3	50	0.90952384±0.09642122	0.4534884±0.09759441	0.0904762±0.09642123	0.5465116±0.0975944	0.94200003±0.37015915	2.244±0.6451735	0.6259264±0.050273042	0.0904762±0.09642123	50.0±0.0
3	3	150	0.7571429±0.14966157	0.6976744±0.08701528	0.24285717±0.14966159	0.30232558±0.08701529	1.7653332±0.2960447	2.4806666±0.33452305	0.71339524±0.082226984	0.24285717±0.14966159	150.0±0.0
3	3	500	0.62857145±0.08922837	0.7930232±0.035439156	0.37142858±0.08922838	0.20697674±0.03543916	2.0036001±0.17639302	2.2603998±0.17629975	0.7518362±0.03440971	0.37142858±0.08922838	500.0±0.0
3	3	1500	0.5047619±0.13127667	0.84651166±0.02499926	0.49523813±0.13127667	0.15348837±0.024999248	2.2756667±0.27686203	2.3768666±0.28077653	0.76043564±0.050043695	0.49523813±0.13127667	1500.0±0.0
3	3	5000	0.40476188±0.0958972	0.9325582±0.031867906	0.5952381±0.09589719	0.067441866±0.031867914	2.38098±0.12721117	2.41428±0.1281865	0.86258±0.06032462	0.5952381±0.09589719	5000.0±0.0
3	**3**	**50000**	**0.24285717±0.119733475**	**0.96744186±0.027297385**	**0.7571429±0.11973347**	**0.03255814±0.027297392**	**2.4022138±0.17931214**	**2.405682±0.1793297**	**0.88656634±0.10345073**	**0.7571429±0.11973347**	**50000.0±0.0**
4	3	50	0.9714286±0.06388765	0.5069767±0.10707781	0.02857143±0.06388766	0.49302325±0.107077815	1.076±0.2747362	2.216±0.3876597	0.66563004±0.04371143	0.02857143±0.06388766	50.0±0.0
4	3	150	0.82857144±0.07221786	0.6465117±0.07242981	0.17142859±0.07221787	0.3534884±0.07242981	2.1266665±0.4236613	2.7893333±0.49465138	0.70199794±0.049137842	0.17142859±0.07221787	150.0±0.0
4	3	500	0.6952381±0.0796819	0.75348836±0.02651572	0.30476195±0.07968191	0.24651162±0.026515713	2.8256±0.42153382	3.0808±0.4260155	0.73721564±0.030173426	0.30476195±0.07968191	500.0±0.0
4	3	1500	0.59047616±0.20919889	0.84651166±0.048224285	0.40952381±0.20919889	0.15348837±0.04822428	3.0088±0.28610775	3.1146667±0.29324123	0.7887158±0.045706663	0.40952381±0.20919889	1500.0±0.0
4	3	5000	0.43809524±0.052164063	0.9116279±0.025475455	0.5619048±0.052164063	0.0883721±0.025475468	3.0554802±0.122387156	3.08924±0.12225201	0.83405626±0.038905576	0.5619048±0.052164063	5000.0±0.0
4	**3**	**50000**	**0.35238096±0.07221787**	**0.96279067±0.020800643**	**0.64761907±0.07221786**	**0.037209302±0.020800631**	**3.1839957±0.22540231**	**3.1874518±0.22550656**	**0.91240656±0.04973411**	**0.64761907±0.07221786**	**50000.0±0.0**
4	4	50	0.9238095±0.060233854	0.40697676±0.14636596	0.07619048±0.06023386	0.59302324±0.14636596	0.53200006±0.22670588	1.5680001±0.44002017	0.6137885±0.046982046	0.07619048±0.06023386	50.0±0.0
4	4	150	0.71428573±0.128953	0.6767441±0.054208167	0.28571433±0.128953	0.32325578±0.05420816	0.91133326±0.2303631	1.5400001±0.28541502	0.6856849±0.052027408	0.28571433±0.128953	150.0±0.0
4	4	500	0.59047616±0.11920625	0.81162786±0.056381695	0.4095238±0.11920625	0.1883721±0.056381695	1.299±0.20172203	1.5583999±0.2137076	0.7579904±0.04694468	0.4095238±0.11920625	500.0±0.0
4	4	1500	0.47619048±0.10528968	0.90232563±0.034319296	0.52380955±0.10528968	0.097674415±0.034319296	1.4613334±0.12205118	1.5626667±0.12782474	0.8279954±0.050257098	0.52380955±0.10528968	1500.0±0.0
4	4	5000	0.3809524±0.10997148	0.9488373±0.018344415	0.6190476±0.10997148	0.05116279±0.018344436	1.54814±0.11158635	1.5810201±0.111567646	0.87776935±0.04593318	0.6190476±0.10997148	5000.0±0.0
4	**4**	**50000**	**0.20476191±0.09269081**	**0.9930233±0.011233631**	**0.7952381±0.0926908**	**0.0069767446±0.011233626**	**1.6100502±0.06679349**	**1.6134479±0.06683407**	**0.9693658±0.05308783**	**0.7952381±0.0926908**	**50000.0±0.0**
5	4	50	0.9238095±0.07221786	0.46046513±0.21716496	0.07619049±0.07221787	0.53953487±0.21716496	0.84±0.74551994	2.2519999±1.4527628	0.64384294±0.09613654	0.07619049±0.07221787	50.0±0.0
5	4	150	0.63809526±0.20370713	0.74418604±0.049333032	0.36190477±0.20370714	0.25581396±0.049333036	1.7426666±0.24606232	2.524±0.3358769	0.69885695±0.09920049	0.36190477±0.20370714	150.0±0.0
5	4	500	0.49523813±0.098744966	0.81860465±0.050417397	0.5047619±0.098744966	0.18139535±0.0504174	1.9104±0.3681451	2.2008±0.39550626	0.72994554±0.076897345	0.5047619±0.098744966	500.0±0.0
5	4	1500	0.49523813±0.0865043	0.8930233±0.02651572	0.50476193±0.0865043	0.10697675±0.02651571	1.9210666±0.120027445	2.0204±0.120296516	0.81863135±0.053706292	0.50476193±0.0865043	1500.0±0.0
5	4	5000	0.34285718±0.021295885	0.96279067±0.020800618	0.6571429±0.021295883	0.037209302±0.020800633	1.9915597±0.10410085	2.02508±0.10379436	0.90330106±0.050031748	0.6571429±0.021295883	5000.0±0.0
5	**4**	**50000**	**0.2±0.10858814**	**0.9953488±0.01040032**	**0.8±0.10858813**	**0.004651163±0.010400316**	**2.0999238±0.13082394**	**2.103408±0.13086733**	**0.972±0.062609896**	**0.8±0.10858813**	**50000.0±0.0**
5	5	50	0.9±0.06525334	0.34651163±0.15089422	0.1±0.065253355	0.6534883±0.15089421	0.226±0.12294534	1.0439999±0.40560517	0.58369255±0.048040938	0.1±0.065253355	50.0±0.0
5	5	150	0.80952376±0.0839921	0.6348837±0.105180964	0.19047621±0.0839921	0.36511627±0.10518097	0.49599996±0.1086028	1.0526667±0.1850312	0.6938271±0.048111763	0.19047621±0.0839921	150.0±0.0
5	5	500	0.5619048±0.109512314	0.8255814±0.033342335	0.43809524±0.109512314	0.1744186±0.033342343	0.81560004±0.09962061	1.074±0.10809872	0.76014435±0.042628147	0.43809524±0.109512314	500.0±0.0
5	5	1500	0.4095238±0.07512482	0.9186047±0.033342335	0.59047616±0.07512482	0.08139535±0.033342347	0.9352668±0.082310155	1.0353999±0.0838461	0.8338618±0.061916098	0.59047616±0.07512482	1500.0±0.0
5	5	5000	0.2761905±0.12458043	0.96279067±0.027297381	0.7238096±0.12458042	0.037209302±0.02729739	1.03748±0.08908133	1.0694001±0.08932238	0.8504181±0.14241457	0.7238096±0.12458042	5000.0±0.0
5	**5**	**50000**	**0.22857146±0.083390005**	**0.99767435±0.0073541375**	**0.7714286±0.08339**	**0.0023255814±0.007354134**	**1.0216999±0.040236343**	**1.024978±0.040245812**	**0.9924731±0.023802103**	**0.7714286±0.08339**	**50000.0±0.0**

Table 21. *Glass Data Set under Feature-Detection Rule (Part 2)*

Detector Length	r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
10	9	50	0.9809524±0.026082026	0.06511628±0.044733457	0.01904762±0.026082026	0.9348837±0.044733446	0.012±0.010954451	0.19999999±0.10198039	0.512184±0.010043031	0.01904762±0.026082026	50.0±0.0
10	9	150	0.9333334±0.054294065	0.2±0.048224285	0.06666667±0.05429407	0.8±0.048224278	0.016±0.018012341	0.17333333±0.027888669	0.5384579±0.019271424	0.06666667±0.05429407	150.0±0.0
10	9	500	0.7714286±0.10323563	0.56279075±0.034493938	0.22857144±0.10323564	0.43720928±0.03449394	0.059599996±0.022645088	0.206±0.035972208	0.63621765±0.046369754	0.22857144±0.10323564	500.0±0.0
10	9	1500	0.61904764±0.18747638	0.7813953±0.07817552	0.3809524±0.1874764	0.21860465±0.07817552	0.10946667±0.010722767	0.18786666±0.010722771	0.73581046±0.08632018	0.3809524±0.1874764	1500.0±0.0
10	9	5000	0.2952381±0.07824608	0.93953484±0.020800613	0.7047619±0.07824607	0.060465116±0.020800633	0.16504±0.014394027	0.19483998±0.014661105	0.82546616±0.060772207	0.7047619±0.07824607	5000.0±0.0
10	**9**	**50000**	**0.15238096±0.062087644**	**0.9953488±0.01040032**	**0.84761906±0.06208764**	**0.004651163±0.010400316**	**0.18932±0.005536138**	**0.192452±0.0055039707**	**0.9607477±0.0877709**	**0.84761906±0.06208764**	**50000.0±0.0**
10	10	50	0.98571426±0.023002183	0.034883723±0.057227995	0.014285715±0.023002185	0.9651163±0.057228003	0.0±0.0	0.07599999±0.07705697	0.5056297±0.014639738	0.014285715±0.023002185	50.0±0.0
10	10	150	0.9666666±0.04517539	0.06511629±0.061821397	0.033333335±0.045175396	0.9348837±0.06182139	0.0039999997±0.0056218267	0.06600001±0.032727834	0.50859964±0.023596324	0.033333335±0.045175396	150.0±0.0
10	10	500	0.88095236±0.08473871	0.2488372±0.12259341	0.11904763±0.08473872	0.7511627±0.12259341	0.0116±0.009559173	0.081±0.031513665	0.5414856±0.055244934	0.11904763±0.08473872	500.0±0.0
10	10	1500	0.75714284±0.11099767	0.5348837±0.041019402	0.24285714±0.11099768	0.46511632±0.0410194	0.024266666±0.0064574587	0.0746±0.008952275	0.61675787±0.05156084	0.24285714±0.11099768	1500.0±0.0
10	10	5000	0.46190482±0.12305427	0.8790698±0.034319308	0.53809524±0.12305427	0.12093024±0.034319296	0.051779997±0.005320359	0.07834±0.0066653676	0.7908218±0.04244471	0.53809524±0.12305427	5000.0±0.0
10	**10**	**50000**	**0.21904762±0.09309764**	**0.9953488±0.009805516**	**0.78095233±0.093097635**	**0.004651163±0.009805513**	**0.073916±0.0034220442**	**0.076968±0.0034422753**	**0.98357487±0.03477789**	**0.78095233±0.093097635**	**50000.0±0.0**
23	22	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	50.0±0.0
23	22	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	150.0±0.0
23	22	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	500.0±0.0
23	22	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	1.3333333E-4±2.981424E-4	0.5±0.0	0.0±0.0	1500.0±0.0
23	22	5000	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	8.0E-5±1.7888544E-4	0.5±0.0	0.0±0.0	5000.0±0.0
23	**22**	**50000**	**0.9714285±0.04259177**	**0.013953489±0.012737734**	**0.02857143±0.04259177**	**0.9860465±0.01273774**	**0.0±0.0**	**6.8E-5±5.215362E-5**	**0.4960878±0.009537603**	**0.02857143±0.04259177**	**50000.0±0.0**
23	23	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	50.0±0.0
23	23	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	150.0±0.0
23	23	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	500.0±0.0
23	23	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	1500.0±0.0
23	23	5000	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	5000.0±0.0
23	**23**	**50000**	**0.9952381±0.015058463**	**0.004651163±0.014708268**	**0.004761905±0.015058466**	**0.9953488±0.014708276**	**0.0±0.0**	**1.19999995E-5±2.1499354E-5**	**0.49997097±0.005680728**	**0.004761905±0.015058466**	**50000.0±0.0**
30	29	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	50.0±0.0
30	29	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	150.0±0.0
30	29	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	500.0±0.0
30	29	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	1500.0±0.0
30	29	5000	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	5000.0±0.0
30	**29**	**50000**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.5±0.0**	**0.0±0.0**	**50000.0±0.0**
30	30	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	50.0±0.0
30	30	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	150.0±0.0
30	30	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	500.0±0.0
30	30	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	1500.0±0.0
30	30	5000	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	5000.0±0.0
30	**30**	**50000**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.5±0.0**	**0.0±0.0**	**50000.0±0.0**

Table 22. *Glass Data Set under HD Rule (Part 1)*

r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
3	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
3	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
3	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
3	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
3	5000	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
3	**50000**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.5±0.0**	**0.0±0.0**	**0.0±0.0**
4	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
4	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
4	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
4	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
4	5000	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
4	**50000**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.5±0.0**	**0.0±0.0**	**0.0±0.0**
5	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
5	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
5	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
5	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
5	5000	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
5	**50000**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.5±0.0**	**0.0±0.0**	**0.0±0.0**
10	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
10	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
10	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
10	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
10	5000	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
10	**50000**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.5±0.0**	**0.0±0.0**	**0.0±0.0**
20	50	0.9952381±0.015058463	0.048837207±0.10822216	0.004761905±0.015058466	0.9511628±0.108222164	0.0±0.0	6.5±10.102255	0.5129782±0.029135816	0.004761905±0.015058466	0.4±0.51639783
20	150	0.97619045±0.06044264	0.1±0.12502028	0.023809524±0.060442645	0.9±0.12502028	0.6±1.3498971	10.925±12.682036	0.5220973±0.02855374	0.023809524±0.060442645	0.9±1.286684
20	500	0.91904765±0.081092305	0.3372093±0.2157352	0.08095238±0.08109232	0.66279066±0.2157352	2.77±3.481962	20.369999±7.998679	0.5918059±0.06953432	0.08095238±0.08109232	2.4±1.4298407
20	1500	0.7761905±0.12708329	0.60697675±0.15129194	0.22380953±0.1270833	0.39302325±0.15129194	8.548217±3.9590602	19.17448±2.871987	0.67253184±0.07721886	0.22380953±0.1270833	9.5±4.5030856
20	5000	0.5952381±0.12944052	0.82790697±0.106402226	0.40476194±0.12944053	0.17209303±0.10640223	15.531412±3.8455102	20.749231±4.114126	0.7819675±0.08519047	0.40476194±0.12944053	28.4±8.983936
20	**50000**	**0.252381±0.10777298**	**0.988372±0.022600584**	**0.7476191±0.107772976**	**0.011627907±0.022600587**	**19.770397±1.6137989**	**20.286346±1.5261385**	**0.9589304±0.076007016**	**0.7476191±0.107772976**	**350.3±76.49408**
23	50	0.7047619±0.1706626	0.6372093±0.14919208	0.2952381±0.17066261	0.3627907±0.14919208	5.570545±1.9236504	10.514498±2.4599261	0.6643947±0.09595997	0.2952381±0.17066261	20.4±3.6575644
23	150	0.50476193±0.08751778	0.855814±0.060841605	0.49523807±0.08751778	0.14418605±0.060841605	8.33427±0.9674668	10.612213±1.0347575	0.7803383±0.073570535	0.49523807±0.08751778	60.1±8.5693245
23	500	0.32380956±0.10718694	0.9604651±0.022062395	0.6761905±0.10718694	0.039534885±0.022062402	9.829529±0.7881025	10.598936±0.78515047	0.8881976±0.06961326	0.6761905±0.10718694	207.2±19.112534
23	1500	0.23809524±0.0925548	0.988372±0.016444352	0.76190484±0.092554785	0.011627907±0.016444344	10.589329±0.48597506	10.855404±0.47665945	0.96619797±0.048766017	0.76190484±0.092554785	621.6±60.939312
23	5000	0.1809524±0.049180746	1.0±0.0	0.81904775±0.049180735	0.0±0.0	10.65942±0.5930913	10.74222±0.58803684	1.0±0.0	0.81904775±0.049180735	2029.0±157.25563
23	**50000**	**0.1809524±0.06269339**	**1.0±0.0**	**0.81904763±0.06269339**	**0.0±0.0**	**10.810839±0.5617536**	**10.819035±0.56157833**	**1.0±0.0**	**0.81904763±0.06269339**	**20548.5±1128.9988**
26	50	0.6142857±0.091045275	0.7069767±0.0728847	0.3857143±0.09104527	0.29302323±0.07288471	2.918±0.80141264	5.034±1.0422219	0.67835623±0.053125173	0.3857143±0.09104527	50.0±0.0
26	150	0.40952381±0.100890465	0.89767444±0.03501269	0.5904762±0.100890465	0.10232558±0.03501268	4.3953156±0.5020622	5.3593154±0.52150756	0.7974443±0.065636456	0.5904762±0.100890465	149.9±0.31622776
26	500	0.21904762±0.09309764	0.972093±0.021370629	0.78095233±0.093097635	0.027906975±0.021370618	4.7962947±0.33945233	5.1128225±0.34092423	0.86870784±0.14645883	0.78095233±0.093097635	499.8±0.421637
26	1500	0.21428573±0.07186813	0.9953488±0.009805517	0.78571427±0.07186813	0.004651163±0.009805513	4.8649993±0.20390247	4.9706416±0.20581971	0.9859514±0.029711718	0.78571427±0.07186813	1499.4±1.0749677
26	5000	0.15238097±0.03756241	1.0±0.0	0.84761906±0.03756241	0.0±0.0	5.155404±0.43017474	5.1876907±0.43062395	1.0±0.0	0.84761906±0.03756241	4999.0±1.0540925
26	**50000**	**0.17142858±0.03329552**	**1.0±0.0**	**0.82857144±0.033295516**	**0.0±0.0**	**5.0095086±0.22666542**	**5.012742±0.22665998**	**1.0±0.0**	**0.82857144±0.033295516**	**49983.2±6.47731**

Table 23. *Glass Data Set under HD Rule (Part 2)*

r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
30	50	0.8333333±0.09848947	0.33023256±0.110611215	0.16666667±0.09848947	0.6697674±0.11061121	0.25800002±0.057696525	1.266±0.22921123	0.5553608±0.053901047	0.16666667±0.09848947	50.0±0.0
30	150	0.6142857±0.12982924	0.66511625±0.09250501	0.38571432±0.12982926	0.33488372±0.092505015	0.508±0.13766304	1.1506667±0.20476304	0.6491089±0.052665647	0.38571432±0.12982926	150.0±0.0
30	500	0.352381±0.09309765	0.9348837±0.030617688	0.6476191±0.09309764	0.06511629±0.030617703	0.8539999±0.13116571	1.1345999±0.13727847	0.84826726±0.05647938	0.6476191±0.09309764	500.0±0.0
30	1500	0.22380956±0.08414196	0.98139536±0.021370629	0.7761905±0.08414195	0.018604651±0.02137062	1.0020001±0.074924156	1.1024666±0.07616618	0.915734±0.101191044	0.7761905±0.08414195	1500.0±0.0
30	5000	0.24285717±0.06127067	1.0±0.0	0.7571429±0.06127066	0.0±0.0	1.05916±0.037076846	1.08988±0.037422307	1.0±0.0	0.7571429±0.06127066	5000.0±0.0
30	**50000**	**0.15714286±0.06753031**	**1.0±0.0**	**0.8428572±0.067530304**	**0.0±0.0**	**1.094888±0.032369517**	**1.0979521±0.032376822**	**1.0±0.0**	**0.8428572±0.067530304**	**50000.0±0.0**
34	50	0.99047625±0.020077953	0.030232554±0.031104501	0.00952381±0.020077955	0.9697674±0.031104503	0.0019999999±0.0063245553	0.08±0.06733003	0.5053501±0.01009524	0.00952381±0.020077955	50.0±0.0
34	150	0.9666668±0.023002183	0.04883721±0.025593137	0.033333335±0.023002187	0.9511628±0.025593122	0.0026666666±0.004661373	0.06933333±0.029847348	0.5040595±0.02936059	0.033333335±0.023002187	150.0±0.0
34	500	0.88571435±0.08751776	0.21627907±0.07519779	0.11428572±0.08751778	0.78372085±0.0751978	0.010000001±0.0065996633	0.073±0.016391056	0.5303013±0.031133622	0.11428572±0.08751778	500.0±0.0
34	1500	0.74285716±0.10812308	0.47674417±0.07125883	0.25714287±0.108123094	0.5232558±0.07125882	0.018666666±0.0039999997	0.065066665±0.008398707	0.58611083±0.027190857	0.25714287±0.108123094	1500.0±0.0
34	5000	0.37142855±0.06659105	0.87674415±0.042529825	0.6285714±0.06659104	0.12325583±0.042529818	0.04182±0.0039185598	0.067660004±0.0043767556	0.7500234±0.08258855	0.6285714±0.06659104	5000.0±0.0
34	**50000**	**0.18571429±0.065253355**	**1.0±0.0**	**0.81428564±0.06525335**	**0.0±0.0**	**0.065859996±0.002002288**	**0.068849996±0.0020236871**	**1.0±0.0**	**0.81428564±0.06525335**	**50000.0±0.0**
37	50	1.0±0.0	0.0023255814±0.007354134	0.0±0.0	0.99767435±0.0073541375	0.0±0.0	0.0019999999±0.0063245553	0.50058824±0.0018601726	0.0±0.0	50.0±0.0
37	150	0.9952381±0.015058463	0.0023255814±0.007354134	0.004761905±0.015058466	0.99767435±0.0073541375	0.0±0.0	0.0026666666±0.004661373	0.49936873±0.004463906	0.004761905±0.015058466	150.0±0.0
37	500	0.9857143±0.045175392	0.004651163±0.009805513	0.014285715±0.045175407	0.9953488±0.009805517	0.0±0.0	0.0028000001±0.0035527768	0.49733034±0.012811552	0.014285715±0.045175407	500.0±0.0
37	1500	0.99047625±0.020077951	0.018604651±0.018344432	0.00952381±0.020077953	0.98139536±0.018344441	0.0±0.0	0.0015333334±7.0623326E-4	0.5022945±0.0064957417	0.00952381±0.020077953	1500.0±0.0
37	5000	0.9476191±0.0725659	0.09534884±0.06245003	0.052380957±0.072565906	0.90465117±0.06245003	9.999999E-5±1.0540926E-4	0.00264±7.988881E-4	0.51140445±0.03340081	0.052380957±0.072565906	5000.0±0.0
37	**50000**	**0.6142857±0.106361054**	**0.6232558±0.13054596**	**0.3857143±0.10636105**	**0.3767442±0.13054596**	**7.7000004E-4±1.617268E-4**	**0.0024100002±2.6637485E-4**	**0.6243481±0.1018129**	**0.3857143±0.10636105**	**50000.0±0.0**
40	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	50.0±0.0
40	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	150.0±0.0
40	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	500.0±0.0
40	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	1500.0±0.0
40	5000	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	5000.0±0.0
40	**50000**	**0.99047625±0.030116927**	**0.004651163±0.009805513**	**0.00952381±0.030116932**	**0.99534875±0.009805516**	**0.0±0.0**	**1.19999995E-5±1.0327954E-5**	**0.49867645±0.008671075**	**0.00952381±0.030116932**	**50000.0±0.0**
43	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	50.0±0.0
43	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	150.0±0.0
43	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	500.0±0.0
43	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	1500.0±0.0
43	5000	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	5000.0±0.0
43	**50000**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.5±0.0**	**0.0±0.0**	**50000.0±0.0**

Table 24. *Glass Data Set under RCHK Rule (No MHC)*

r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
3	50	0.9952381±0.015058463	0.0±0.0	0.004761905±0.015058466	1.0±0.0	0.3333333±1.0540924	0.7±2.213595	0.4987805±0.0038564324	0.004761905±0.015058466	0.3±0.9486834
3	150	0.9809524±0.033295516	0.025581395±0.080895476	0.01904762±0.03329552	0.97441864±0.08089548	1.5966666±2.5871692	2.2117648±3.5742471	0.50243±0.017026598	0.01904762±0.03329552	2.8±5.49343
3	500	0.94285715±0.083390005	0.11627908±0.19239806	0.05714286±0.083390005	0.883721±0.19239803	2.7477965±3.569662	3.0850585±4.0049863	0.52147686±0.03984839	0.05714286±0.083390005	22.9±32.439518
3	1500	0.95238096±0.077761576	0.06511628±0.13727717	0.04761905±0.07776158	0.9348837±0.13727716	3.0727916±4.0080724	3.1690917±4.137878	0.5066654±0.021075856	0.04761905±0.07776158	62.5±100.80702
3	5000	0.9619048±0.058536787	0.04418605±0.13972855	0.03809524±0.05853679	0.955814±0.13972853	3.780357±4.0869803	3.8119469±4.1243753	0.5043122±0.031367112	0.03809524±0.05853679	158.9±214.72952
3	**50000**	**0.9476191±0.069006555**	**0.037209302±0.11766615**	**0.052380957±0.069006555**	**0.96279067±0.11766614**	**4.8788047±3.3671396**	**4.8828077±3.3698754**	**0.49750367±0.018719977**	**0.052380957±0.069006555**	**2177.0±2437.418**
4	50	0.8952381±0.08339	0.37906975±0.07358219	0.10476191±0.083390005	0.6209302±0.07358219	3.1855748±0.835195	5.220943±1.1344805	0.5909282±0.018137148	0.10476191±0.083390005	32.9±5.021066
4	150	0.81428564±0.06127066	0.46046513±0.06740175	0.18571429±0.06127067	0.53953487±0.06740175	4.2986846±0.794539	5.171873±0.80291915	0.60217774±0.034330837	0.18571429±0.06127067	100.2±11.970333
4	500	0.80952376±0.14197256	0.49534884±0.06582303	0.1904762±0.14197257	0.5046512±0.06582303	4.783385±0.57271	5.08765±0.55205864	0.61399364±0.029802337	0.1904762±0.14197257	321.6±47.44869
4	1500	0.82857144±0.09309763	0.5488372±0.059240222	0.17142859±0.09309765	0.45116282±0.059240233	4.7709265±0.40157795	4.8807225±0.39577875	0.64760953±0.02502077	0.17142859±0.09309765	949.5±132.75311
4	5000	0.68095237±0.13846874	0.6953488±0.0945927	0.31904763±0.13846876	0.30465117±0.0945927	4.983709±0.49624225	5.0197806±0.4993514	0.69559115±0.02971134	0.31904763±0.13846876	3306.9±347.2164
4	**50000**	**0.64285713±0.081711344**	**0.760465±0.05029807**	**0.35714287±0.081711344**	**0.23953488±0.05029807**	**4.9721837±0.22040553**	**4.976135±0.22020245**	**0.7290802±0.044787027**	**0.35714287±0.081711344**	**34518.7±3375.251**
5	50	0.8761905±0.07169263	0.46279067±0.053088035	0.12380953±0.07169263	0.53720933±0.053088047	1.7452924±0.46817815	3.1875374±0.59823483	0.61975664±0.0334265	0.12380953±0.07169263	49.3±0.82327265
5	150	0.8333334±0.115010925	0.6139535±0.082913116	0.16666669±0.11501093	0.3860465±0.08291313	2.8135474±0.3166245	3.474409±0.32508323	0.6843349±0.048169978	0.16666669±0.11501093	146.9±1.4491377
5	500	0.74761903±0.08708487	0.6837209±0.0528043	0.25238097±0.08708487	0.31627905±0.052804295	3.2594776±0.45599228	3.4782548±0.4539657	0.70356756±0.031376023	0.25238097±0.08708487	492.8±2.2509258
5	1500	0.77619046±0.114571944	0.71860456±0.05194376	0.22380953±0.114571944	0.28139538±0.051943764	3.2410684±0.24100436	3.3171177±0.24491523	0.7340803±0.03201579	0.22380953±0.114571944	1475.1±8.900063
5	5000	0.63809526±0.12738033	0.7651162±0.067088984	0.36190477±0.12738034	0.23488374±0.067088984	3.4977927±0.27101275	3.524032±0.27346817	0.734777±0.034182213	0.36190477±0.12738034	4927.1±13.101739
5	**50000**	**0.6095238±0.12458042**	**0.7767441±0.046769306**	**0.3904762±0.124580435**	**0.22325583±0.046769314**	**3.4485002±0.14926931**	**3.4511216±0.14924602**	**0.72927964±0.049658358**	**0.3904762±0.124580435**	**49165.4±78.287506**
10	50	0.9476191±0.04735376	0.15581395±0.10110263	0.052380957±0.04735376	0.844186±0.10110263	0.053999998±0.048120223	0.43600002±0.15629032	0.53013474±0.03783357	0.052380957±0.04735376	50.0±0.0
10	150	0.8761905±0.095765725	0.3651163±0.071930304	0.12380953±0.09576573	0.6348837±0.07193028	0.13133332±0.048488256	0.49800006±0.11452748	0.57956666±0.029593945	0.12380953±0.09576573	150.0±0.0
10	500	0.6952381±0.100890465	0.6837209±0.0728847	0.30476195±0.100890465	0.31627908±0.072884716	0.2928±0.04360632	0.489±0.050184116	0.68830246±0.048873566	0.30476195±0.100890465	500.0±0.0
10	1500	0.5714286±0.11878277	0.80930233±0.042175096	0.42857146±0.11878277	0.19069766±0.042175103	0.3857333±0.036745638	0.4692667±0.039345823	0.74555105±0.05858532	0.42857146±0.11878277	1500.0±0.0
10	5000	0.4666667±0.07027285	0.8860465±0.041673433	0.53333336±0.07027285	0.11395349±0.04167343	0.44923997±0.020304529	0.47823995±0.02025654	0.80627376±0.06508751	0.53333336±0.07027285	5000.0±0.0
10	**50000**	**0.3952381±0.08708487**	**0.9046513±0.013201081**	**0.6047619±0.08708486**	**0.09534883±0.013201076**	**0.46049±0.015004503**	**0.46347603±0.015084453**	**0.80131805±0.03580803**	**0.6047619±0.08708486**	**50000.0±0.0**
23	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	50.0±0.0
23	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	150.0±0.0
23	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	500.0±0.0

Table 25. *Glass Data Set under RCHK (Global MHC) Rule*

r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
3	50	0.99047625±0.030116927	0.076744184±0.1626245	0.00952381±0.030116932	0.92325574±0.16262451	3.2416663±9.010835	6.166667±14.990738	0.5213848±0.046649527	0.00952381±0.030116932	1.1±2.1317704
3	150	0.8952381±0.25868633	0.13255814±0.3112478	0.10476191±0.25868633	0.86744183±0.3112478	3.4716668±7.350401	5.220834±11.1586485	0.5351945±0.09556517	0.10476191±0.25868633	2.9±7.5784783
3	500	0.60952383±0.50485176	0.38372093±0.49079624	0.39047617±0.50485176	0.61627907±0.49079624	15.611067±22.310247	17.02638±23.97929	0.35117644±0.24234612	0.39047617±0.50485176	27.8±32.604702
3	1500	0.9±0.31622773	0.12325583±0.31182644	0.1±0.31622773	0.87674415±0.3118265	6.48541±14.171725	6.652715±14.523705	0.45617285±0.1607956	0.1±0.31622773	62.1±70.2115
3	5000	0.81428564±0.39168093	0.18372092±0.38766667	0.18571429±0.39168093	0.81627905±0.38766667	6.5678735±14.0471115	6.6312346±14.180851	0.49561438±0.016042164	0.18571429±0.39168093	243.9±294.74637
3	**50000**	**0.9666666±0.10540926**	**0.041860465±0.1323744**	**0.033333335±0.105409265**	**0.95813954±0.13237442**	**4.0964155±12.954003**	**4.1±12.96534**	**0.5034162±0.0108028175**	**0.033333335±0.105409265**	**442.1±932.0337**
4	50	0.05714286±0.109512314	0.86976737±0.0871533	0.94285715±0.1095123	0.13023256±0.0871533	8.958323±3.3654995	14.456305±3.9609573	0.13361117±0.22482152	0.94285715±0.1095123	33.0±3.1269438
4	150	0.0±0.0	0.9953488±0.014708276	1.0±0.0	0.004651163±0.014708268	12.484487±2.6378107	14.769422±2.6221848	0.0±0.0	1.0±0.0	97.1±11.589746
4	500	0.0±0.0	0.9906977±0.0120092565	1.0±0.0	0.009302326±0.012009251	19.976822±3.8474326	20.6483±3.7935975	0.0±0.0	1.0±0.0	323.6±41.769737
4	1500	0.0±0.0	0.99767435±0.007354138	1.0±0.0	0.0023255814±0.007354134	20.332546±2.506348	20.556635±2.49479	0.0±0.0	1.0±0.0	984.4±83.32826
4	5000	0.0±0.0	1.0±0.0	1.0±0.0	0.0±0.0	21.162405±3.7877228	21.22525±3.788769	0.0±0.0	1.0±0.0	3490.4±181.23172
4	**50000**	**0.0±0.0**	**1.0±0.0**	**1.0±0.0**	**0.0±0.0**	**21.876986±1.8535818**	**21.88363±1.8536389**	**0.0±0.0**	**1.0±0.0**	**32134.9±3253.5427**
5	50	0.0952381±0.070986286	0.8±0.12934358	0.9047619±0.07098628	0.2±0.12934358	4.9436507±2.6606731	8.139898±3.0746574	0.2993428±0.18799506	0.9047619±0.07098628	48.7±0.94868326
5	150	0.0±0.0	1.0±0.0	1.0±0.0	0.0±0.0	17.391571±2.1354585	18.702265±2.1306646	0.0±0.0	1.0±0.0	148.1±0.994429
5	500	0.0±0.0	1.0±0.0	1.0±0.0	0.0±0.0	16.236181±1.2401501	16.650606±1.2431763	0.0±0.0	1.0±0.0	491.3±3.4009807
5	1500	0.0±0.0	1.0±0.0	1.0±0.0	0.0±0.0	16.421528±0.25572318	16.559887±0.25540945	0.0±0.0	1.0±0.0	1474.5±8.553751
5	5000	0.0±0.0	1.0±0.0	1.0±0.0	0.0±0.0	20.868145±0.5005871	20.91037±0.5021556	0.0±0.0	1.0±0.0	4923.1±24.401505
5	**50000**	**0.0±0.0**	**1.0±0.0**	**1.0±0.0**	**0.0±0.0**	**13.174988±0.8899951**	**13.178961±0.89003295**	**0.0±0.0**	**1.0±0.0**	**49154.0±155.41986**
10	50	0.6857143±0.21201025	0.26976746±0.14634542	0.31428576±0.21201026	0.73023254±0.14634542	0.148±0.10674997	1.136±0.31662107	0.479885±0.10193973	0.31428576±0.21201026	50.0±0.0
10	150	0.23333332±0.13365419	0.7395348±0.0991521	0.7666667±0.13365419	0.26046515±0.0991521	0.4533333±0.16979289	1.1973333±0.23128624	0.45225447±0.19140065	0.7666667±0.13365419	150.0±0.0
10	500	0.01904762±0.03329552	0.9837209±0.019145884	0.98095244±0.033295516	0.016279068±0.019145874	0.768±0.10297789	1.0849999±0.111296415	0.25058824±0.4253067	0.98095244±0.033295516	500.0±0.0
10	1500	0.0±0.0	1.0±0.0	1.0±0.0	0.0±0.0	1.1633999±0.047341637	1.2708±0.047463864	0.0±0.0	1.0±0.0	1500.0±0.0
10	5000	0.0±0.0	1.0±0.0	1.0±0.0	0.0±0.0	1.2457999±0.062283743	1.2801799±0.062932506	0.0±0.0	1.0±0.0	5000.0±0.0
10	**50000**	**0.0±0.0**	**1.0±0.0**	**1.0±0.0**	**0.0±0.0**	**1.147278±0.024349857**	**1.150646±0.024379004**	**0.0±0.0**	**1.0±0.0**	**50000.0±0.0**
23	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	50.0±0.0
23	150	0.9952381±0.015058463	0.0±0.0	0.004761905±0.015058466	1.0±0.0	0.0±0.0	0.0±0.0	0.4987805±0.0038564324	0.004761905±0.015058466	150.0±0.0
23	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	500.0±0.0

Table 26. *Glass Data Set under RCHK (MHC) Rule*

r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
3	50	0.8714286±0.097976506	0.43720928±0.11999238	0.12857142±0.097976506	0.5627907±0.119992375	1.5296618±0.68811005	4.1056714±1.2495579	0.61094445±0.05110929	0.12857142±0.097976506	23.3±2.9078436
3	150	0.8428572±0.110085964	0.67906976±0.07255415	0.15714286±0.11008598	0.3209302±0.07255416	2.8481133±0.87744546	4.5180883±1.1029426	0.7246135±0.059179217	0.15714286±0.11008598	62.1±5.9151406
3	500	0.71428573±0.05019488	0.760465±0.041092582	0.2857143±0.050194886	0.2395349±0.041092582	3.5731494±0.30506957	4.1507907±0.3344633	0.7500489±0.030431	0.2857143±0.050194886	213.0±10.893423
3	1500	0.59047616±0.12130142	0.8255814±0.042811476	0.40952381±0.12130143	0.1744186±0.042811476	3.9422047±0.29657614	4.1672206±0.3055191	0.7694938±0.059313163	0.40952381±0.12130143	634.6±8.846846
3	5000	0.4857143±0.0626934	0.88139534±0.053088054	0.5142857±0.0626934	0.118604645±0.053088043	4.2391233±0.32158607	4.315446±0.3224119	0.8100405±0.06505675	0.5142857±0.0626934	2108.9±56.28193
3	**50000**	**0.2666667±0.12130142**	**0.96279067±0.031392947**	**0.73333335±0.121301405**	**0.037209302±0.031392958**	**4.351283±0.2649779**	**4.3596377±0.26506206**	**0.868186±0.08936007**	**0.73333335±0.121301405**	**21251.1±431.01288**
4	50	0.81428576±0.117610365	0.5116279±0.109079435	0.18571432±0.117610365	0.4883721±0.10907943	1.6879463±0.7762898	3.925763±0.9310747	0.62511814±0.070688814	0.18571432±0.117610365	33.9±3.5103023
4	150	0.6952382±0.13127665	0.7581395±0.045466296	0.30476192±0.13127665	0.24186048±0.045466285	2.748637±0.7025969	3.8315856±0.81384385	0.73948216±0.039385345	0.30476192±0.13127665	102.9±6.903301
4	500	0.5619048±0.12046772	0.83720934±0.052576207	0.43809527±0.12046772	0.1627907±0.0525762	3.6979206±0.4968366	4.106407±0.5325562	0.7782221±0.046235673	0.43809527±0.12046772	348.4±9.512565
4	1500	0.38571432±0.11099768	0.91627914±0.024999252	0.61428577±0.11099768	0.08372094±0.02499925	3.890334±0.402285	4.0452223±0.4051408	0.81850207±0.039274015	0.61428577±0.11099768	1065.2±15.619078
4	5000	0.2904762±0.12387056	0.9581394±0.0306177	0.70952386±0.12387055	0.041860465±0.030617703	3.8408477±0.15966877	3.8886075±0.15942991	0.8649079±0.07622035	0.70952386±0.12387055	3499.6±43.77772
4	**50000**	**0.25238097±0.09537027**	**0.9906977±0.0120092565**	**0.74761903±0.09537027**	**0.009302326±0.012009251**	**3.8780797±0.30551845**	**3.8829963±0.30545288**	**0.9669405±0.04447665**	**0.74761903±0.09537027**	**34979.0±724.2717**
5	50	0.8238095±0.08414195	0.6046511±0.11602039	0.17619048±0.08414196	0.39534885±0.116020374	1.6741127±0.8311934	3.5559907±1.1541469	0.6801237±0.06917711	0.17619048±0.08414196	46.8±1.8135295
5	150	0.5714286±0.10997148	0.79999995±0.03983017	0.42857146±0.109971486	0.2±0.039830178	2.837321±0.2930957	3.7692275±0.31316403	0.73986036±0.030842584	0.42857146±0.109971486	137.2±4.685675
5	500	0.36666667±0.118888795	0.92093027±0.036688864	0.6333334±0.118888795	0.07906978±0.03668887	3.3038394±0.3229541	3.6289482±0.3281322	0.8216284±0.07052505	0.6333334±0.118888795	461.9±5.915141
5	1500	0.27142856±0.09537028	0.9651163±0.01976364	0.7285715±0.09537027	0.034883723±0.019763643	3.580412±0.24787942	3.6990154±0.24814913	0.8835987±0.06994	0.7285715±0.09537027	1378.6±13.71293
5	5000	0.1809524±0.05853679	0.9906977±0.0120092565	0.8190476±0.058536787	0.009302326±0.012009251	3.53556±0.28326243	3.57175±0.2833429	0.9516964±0.06901005	0.8190476±0.058536787	4598.3±54.83926
5	**50000**	**0.22380956±0.084141955**	**1.0±0.0**	**0.77619046±0.08414195**	**0.0±0.0**	**3.7274082±0.29098976**	**3.7310824±0.29099393**	**1.0±0.0**	**0.77619046±0.08414195**	**45919.5±385.32706**
10	50	0.9285714±0.06447649	0.19069767±0.10841633	0.071428575±0.0644765	0.80930233±0.10841634	0.072±0.061246317	0.562±0.25147563	0.5358424±0.036045093	0.071428575±0.0644765	50.0±0.0
10	150	0.8047619±0.11544823	0.45116282±0.09315237	0.19523811±0.11544824	0.5488373±0.09315237	0.14533333±0.059694696	0.56933326±0.10551228	0.5935995±0.0727671	0.19523811±0.11544824	150.0±0.0
10	500	0.55714285±0.11675033	0.79767436±0.08207542	0.44285718±0.11675033	0.20232558±0.08207542	0.3672±0.08648288	0.607±0.10781981	0.7396345±0.061302025	0.44285718±0.11675033	500.0±0.0
10	1500	0.31904763±0.06753031	0.95348835±0.036359802	0.6809524±0.067530304	0.046511628±0.036359817	0.48613334±0.030569566	0.5854±0.030907985	0.87379456±0.097133994	0.6809524±0.067530304	1500.0±0.0
10	5000	0.17619048±0.07792341	0.9953488±0.009805517	0.8238095±0.07792341	0.004651163±0.009805513	0.56424±0.022898773	0.5951201±0.022716453	0.95828915±0.104448855	0.8238095±0.07792341	5000.0±0.0
10	**50000**	**0.17142859±0.055894658**	**1.0±0.0**	**0.82857144±0.055894654**	**0.0±0.0**	**0.588532±0.016320607**	**0.591588±0.016336117**	**1.0±0.0**	**0.82857144±0.055894654**	**50000.0±0.0**
23	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	50.0±0.0
23	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	150.0±0.0

6.2.8 Glass Data Set Conclusion

The results of the best performing test group for each scenario are summarised in Figure 44. None of the rules performed particularly well for the glass data set since each rule was only able to achieve an average DR value of 0.5. The HD rule followed by the feature-detection rules were the worst performing rules for the glass data set experiment.

A comparison between GC and OC exhibited by the best performing test group for each scenario is summarised in Figure 45. The OC value was equal to 0 across all of the rules within the glass data set. The feature-detection rule and the HD rule had the highest GC values. The RCHK (MHC) rule is the best performing rule, with reference to generalisation and overfitting, because it has the greatest GC minus OC value. The HD rule and feature-detection rule have identical GC minus OC values and are the worst performing rules.

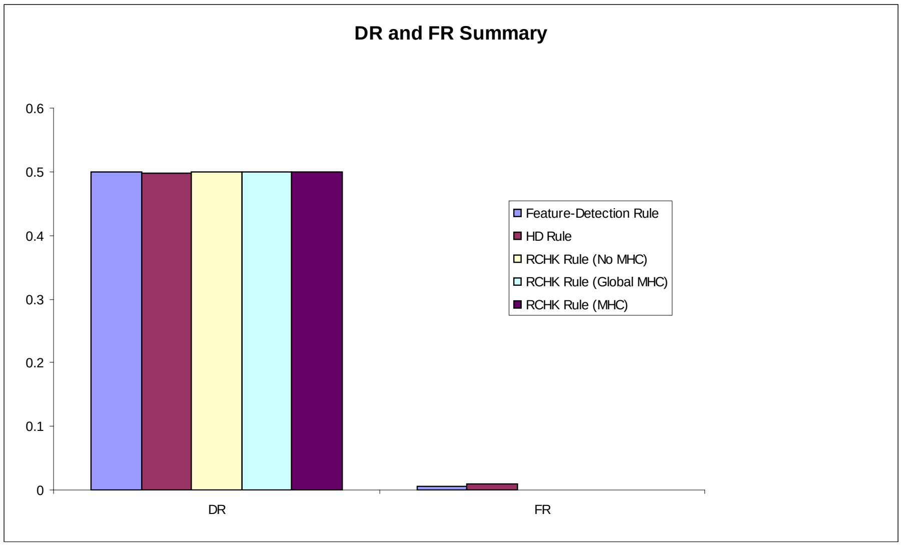


Figure 44. *Glass Data Set: DR and FR Summary*

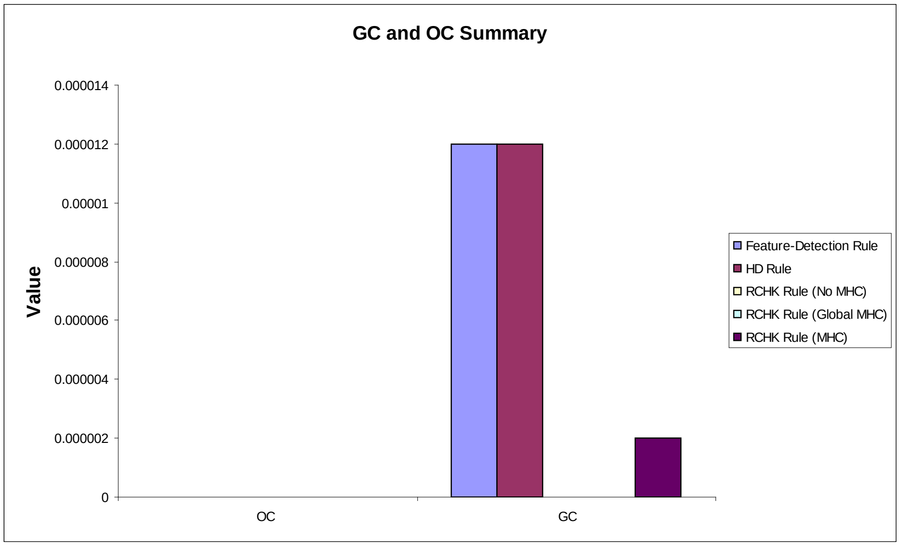


Figure 45. *Glass Data Set: GC and OC Summary*

6.2.9 Mushroom Experiment

The mushroom data set comprises hypothetical samples, corresponding to 23 species of mushroom described by 22 nominal-valued attributes. Each pattern is classified as being edible, non-edible or unknown. The non-edible and unknown classes were combined into the same data set [41] and each pattern was converted into a binary string of length 58. The 2480 missing attributes were replaced by ones. The data sets were further processed to create a single self-set and non-self set as follows:

- Edible.self: Contains 4 208 patterns relating to the edible class.
- Edible.non-self: Contains all the patterns related to poisonous class. The set contains 3 916 patterns in total.

The results of each scenario are reported as per the list below:

- The results of scenario 1, the mushroom data set, under the feature-detection rule are tabulated in Table 27 and Table 28.
- The results of scenario 2, the mushroom data set, under the HD rule are tabulated in Table 29 and Table 30.
- The results of scenario 3, the mushroom data set, under the RCHK rule with no permutation mark are tabulated in Table 31.
- The results of scenario 4, the mushroom data set, under the RCHK rule with a single global permutation mask are tabulated in Table 32.
- The results of scenario 5, the mushroom data set, under the RCHK rule with each detector having its own random permutation mask are tabulated in Table 33.

Table 27. *Mushroom Data Set under Feature-Detection Rule (Part 1)*

Detector Length	r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
4	3	50	1.0±0.0	0.69702125±0.07462234	0.0±0.0	0.3029787±0.07462233	43.463997±23.56166	110.159996±39.4344	0.76958054±0.046078883	0.0±0.0	50.0±0.0
4	3	150	1.0±0.0	0.91217023±0.08229703	0.0±0.0	0.08782979±0.08229704	61.444±20.600685	92.968±24.27559	0.9234908±0.06994341	0.0±0.0	150.0±0.0
4	3	500	1.0±0.0	0.99761707±0.0053285095	0.0±0.0	0.0023829788±0.0053285025	86.8408±6.821222	99.0276±7.2009373	0.9976451±0.005265734	0.0±0.0	500.0±0.0
4	3	1500	0.999683±7.0872996E-4	1.0±0.0	3.1695722E-4±7.087379E-4	0.0±0.0	91.61014±4.3730774	95.86293±4.5231586	1.0±0.0	3.1695722E-4±7.087379E-4	1500.0±0.0
4	3	5000	0.9993661±8.6801336E-4	1.0±0.0	6.3391443E-4±8.680231E-4	0.0±0.0	95.7916±4.2446437	97.073044±4.226847	1.0±0.0	6.3391443E-4±8.680231E-4	5000.0±0.0
4	**3**	**50000**	**0.9982567±0.0011753137**	**1.0±0.0**	**0.0017432647±0.0011753088**	**0.0±0.0**	**96.84412±1.3487492**	**96.97956±1.3469739**	**1.0±0.0**	**0.0017432647±0.0011753088**	**48755.8±33.72981**
4	4	50	1.0±0.0	0.43378726±0.3263127	0.0±0.0	0.5662128±0.3263127	11.955999±7.9828753	53.908±28.419512	0.66635454±0.1519384	0.0±0.0	50.0±0.0
4	4	150	1.0±0.0	0.8775319±0.11462148	0.0±0.0	0.12246808±0.114621475	31.888666±20.311428	59.444664±23.345339	0.89890033±0.0874324	0.0±0.0	150.0±0.0
4	4	500	0.9987322±0.0030621032	0.9940426±0.008697709	0.0012678287±0.0030620976	0.005957446±0.008697708	46.0436±7.661872	57.0828±7.860389	0.9941436±0.008514198	0.0012678287±0.0030620976	500.0±0.0
4	4	1500	0.99928683±0.0013180211	1.0±0.0	7.1315374E-4±0.0013180111	0.0±0.0	50.2868±4.4355907	54.288532±4.5174923	1.0±0.0	7.1315374E-4±0.0013180111	1500.0±0.0
4	**4**	**5000**	**0.99786055±0.0026160937**	**1.0±0.0**	**0.002139461±0.0026160975**	**0.0±0.0**	**53.414345±2.5325062**	**54.6144±2.540683**	**1.0±0.0**	**0.002139461±0.0026160975**	**5000.0±0.0**
4	**4**	**50000**	**0.9869255±5.6034233E-4**	**1.0±0.0**	**0.013074486±5.603061E-4**	**0.0±0.0**	**53.769913±1.1727937**	**53.88747±1.1754507**	**1.0±0.0**	**0.013074486±5.603061E-4**	**50000.0±0.0**
5	4	50	1.0±0.0	0.691234±0.14816506	0.0±0.0	0.30876598±0.14816505	22.359999±9.485874	75.228±20.97463	0.7713316±0.080420494	0.0±0.0	50.0±0.0
5	4	150	1.0±0.0	0.9341276±0.06688463	0.0±0.0	0.06587234±0.06688463	43.749336±26.232157	73.95333±28.537117	0.94110173±0.057888854	0.0±0.0	150.0±0.0
5	4	500	0.9993661±0.0014174865	1.0±0.0	6.3391443E-4±0.0014174758	0.0±0.0	67.489204±5.1937623	79.11±5.0827436	1.0±0.0	6.3391443E-4±0.0014174758	500.0±0.0
5	4	1500	0.9993661±0.0014174865	1.0±0.0	6.3391443E-4±0.0014174758	0.0±0.0	62.81613±4.168236	66.76387±4.150565	1.0±0.0	6.3391443E-4±0.0014174758	1500.0±0.0
5	4	5000	0.9988907±0.001544661	1.0±0.0	0.0011093502±0.0015446584	0.0±0.0	67.33044±2.9782577	68.53693±2.9925284	1.0±0.0	0.0011093502±0.0015446584	5000.0±0.0
5	**4**	**50000**	**0.9964342±5.603002E-4**	**1.0±0.0**	**0.0035657687±5.603063E-4**	**0.0±0.0**	**69.22625±0.20408516**	**69.34732±0.20210527**	**1.0±0.0**	**0.0035657687±5.603063E-4**	**49964.5±0.70710677**
5	5	50	0.9999207±2.5057385E-4	0.42451063±0.27462012	7.9239304E-5±2.5057668E-4	0.5754894±0.27462015	5.4519997±9.797279	35.374±29.373774	0.6519661±0.11061471	7.9239304E-5±2.5057668E-4	50.0±0.0
5	5	150	1.0±0.0	0.7666384±0.12535655	0.0±0.0	0.23336169±0.12535657	11.634001±5.9630756	32.006004±10.77589	0.81799144±0.07914066	0.0±0.0	150.0±0.0
5	5	500	1.0±0.0	0.9687659±0.035439543	0.0±0.0	0.031234046±0.03543954	25.8866±4.0379605	36.0858±4.347611	0.9707239±0.032729674	0.0±0.0	500.0±0.0
5	5	1500	0.9992076±0.0013468176	1.0±0.0	7.923931E-4±0.0013468091	0.0±0.0	30.776134±2.3899307	34.54513±2.4453573	1.0±0.0	7.923931E-4±0.0013468091	1500.0±0.0
5	5	5000	0.9967511±0.0029987942	1.0±0.0	0.0032488115±0.0029987884	0.0±0.0	33.63446±1.8926196	34.8164±1.8993171	1.0±0.0	0.0032488115±0.0029987884	5000.0±0.0
5	**5**	**50000**	**0.9896989±0.007844287**	**1.0±0.0**	**0.010301109±0.007844291**	**0.0±0.0**	**36.15195±0.4049604**	**36.270348±0.4101637**	**1.0±0.0**	**0.010301109±0.007844291**	**50000.0±0.0**
8	7	50	1.0±0.0	0.31642553±0.34844252	0.0±0.0	0.68357444±0.3484425	2.816±3.972188	20.048±16.562859	0.61975133±0.1576186	0.0±0.0	50.0±0.0
8	7	150	1.0±0.0	0.50723404±0.21386015	0.0±0.0	0.49276596±0.21386015	2.568±2.143929	15.149335±5.02384	0.6815764±0.101985164	0.0±0.0	150.0±0.0
8	7	500	0.99809825±0.0034357219	0.82485104±0.13839199	0.0019017432±0.0034357342	0.17514893±0.13839199	6.6131997±2.8061378	14.165601±4.007726	0.85924256±0.08937	0.0019017432±0.0034357342	500.0±0.0
8	7	1500	0.9993661±0.0014174865	1.0±0.0	6.3391443E-4±0.0014174758	0.0±0.0	13.696533±1.3123132	17.344799±1.5937269	1.0±0.0	6.3391443E-4±0.0014174758	1500.0±0.0
8	7	5000	0.999683±4.3400665E-4	0.92459583±0.035268344	3.1695722E-4±4.3401154E-4	0.07540426±0.03526836	8.004±3.4756649	15.944±4.4535556	0.93067235±0.030967826	3.1695722E-4±4.3401154E-4	500.0±0.0
8	**7**	**50000**	**0.9793978±0.0**	**1.0±0.0**	**0.020602219±0.0**	**0.0±0.0**	**17.42082±0.86450803**	**17.53071±0.86565447**	**1.0±0.0**	**0.020602219±0.0**	**50000.0±0.0**
8	8	50	1.0±0.0	0.18476596±0.17278177	0.0±0.0	0.81523407±0.17278178	0.51199996±1.0748934	11.82±9.449247	0.5555171±0.054074783	0.0±0.0	50.0±0.0
8	8	150	0.9998415±5.011477E-4	0.25804254±0.18796925	1.5847861E-4±5.0115335E-4	0.7419575±0.18796924	1.0433333±0.85599846	9.266±3.2747889	0.58058226±0.0678448	1.5847861E-4±5.0115335E-4	150.0±0.0
8	8	500	0.9995246±7.655158E-4	0.5678298±0.18919745	4.7543584E-4±7.655244E-4	0.43217024±0.18919745	1.8024±1.1812133	6.2383995±2.743206	0.7104604±0.10432369	4.7543584E-4±7.655244E-4	500.0±0.0
8	8	1500	0.99960375±6.7339704E-4	0.95472336±0.038047623	3.9619656E-4±6.7340455E-4	0.045276597±0.03804762	5.0198±1.1386182	8.166733±1.298796	0.9577851±0.03393721	3.9619656E-4±6.7340455E-4	1500.0±0.0
8	8	5000	0.9971474±0.003625432	0.99982977±5.3826073E-4	0.002852615±0.0036254325	1.7021276E-4±5.3826E-4	6.6673393±0.84673965	7.7646194±0.8714975	0.99983007±5.373183E-4	0.002852615±0.0036254325	5000.0±0.0
8	**8**	**50000**	**0.98098254±0.002241243**	**1.0±0.0**	**0.019017432±0.002241226**	**0.0±0.0**	**7.6297398±0.517772**	**7.74076±0.513501**	**1.0±0.0**	**0.019017432±0.002241226**	**50000.0±0.0**

Table 28. *Mushroom Data Set under feature-detection Rule (Part 2)*

Detector Length	**r**	**Nc**	**TP**	**TN**	**FP**	**FN**	**OC**	**GC**	**DR**	**FR**	**Actual Population Size**
18	17	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.384±0.8586501	0.5±0.0	0.0±0.0	50.0±0.0
18	17	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	150.0±0.0
18	17	500	1.0±0.0	0.0073191486±0.016366113	0.0±0.0	0.99268085±0.016366122	0.0±0.0	0.0384±0.06258434	0.5018639±0.0041678	0.0±0.0	500.0±0.0
18	17	1500	1.0±0.0	0.0076595745±0.013266772	0.0±0.0	0.99234045±0.013266764	0.0±0.0	0.029066667±0.04750111	0.5019402±0.0033682927	0.0±0.0	1500.0±0.0
18	17	5000	0.999683±7.0872996E-4	0.096510634±0.103747085	3.1695722E-4±7.087379E-4	0.9034894±0.10374709	4.0000002E-4±8.9442724E-4	0.06412±0.04387245	0.5265921±0.03035928	3.1695722E-4±7.087379E-4	5000.0±0.0
18	**17**	**50000**	**0.99524564±0.0022412008**	**0.49617022±0.098694034**	**0.0047543584±0.0022412261**	**0.5038298±0.09869407**	**0.00798±0.0040446506**	**0.04564±0.010606602**	**0.6653642±0.043305043**	**0.0047543584±0.0022412261**	**50000.0±0.0**
18	12	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	1.5439999±1.3739287	0.5±0.0	0.0±0.0	50.0±0.0
18	12	150	1.0±0.0	0.10229788±0.08717154	0.0±0.0	0.8977021±0.08717153	0.376±0.7670811	3.4146667±3.9557674	0.52785975±0.024682764	0.0±0.0	150.0±0.0
18	12	500	0.9995246±7.0872996E-4	0.10791489±0.08620021	4.7543584E-4±7.087379E-4	0.8920851±0.08620021	0.2408±0.25846702	1.9883999±1.0925515	0.52927196±0.023799784	4.7543584E-4±7.087379E-4	500.0±0.0
18	12	1500	0.999683±7.0872996E-4	0.6076596±0.08967726	3.1695722E-4±7.087379E-4	0.39234042±0.08967725	1.0058666±0.5498419	2.9256±0.7191176	0.72039807±0.043517333	3.1695722E-4±7.087379E-4	1500.0±0.0
18	12	5000	0.99587953±0.003238168	0.97021276±0.017243242	0.004120444±0.0032381641	0.029787233±0.017243234	1.55596±0.37439027	2.4749603±0.39238054	0.9711938±0.016166512	0.004120444±0.0032381641	5000.0±0.0
18	**12**	**50000**	**0.98098254±0.0033618433**	**1.0±0.0**	**0.019017432±0.0033618393**	**0.0±0.0**	**2.3811698±0.14437707**	**2.4843202±0.14563574**	**1.0±0.0**	**0.019017432±0.0033618393**	**50000.0±0.0**
18	18	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	50.0±0.0
18	18	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	150.0±0.0
18	18	500	1.0±0.0	0.005531915±0.017493451	0.0±0.0	0.9944681±0.017493445	0.0±0.0	0.0288±0.06479506	0.5014223±0.004497757	0.0±0.0	500.0±0.0
18	18	1500	1.0±0.0	0.0073191486±0.013927892	0.0±0.0	0.9926809±0.0139279	0.0±0.0	0.011066667±0.01660641	0.50185883±0.00355401	0.0±0.0	1500.0±0.0
18	18	5000	0.999683±0.0010023143	0.034042552±0.048406836	3.1695722E-4±0.0010023067	0.96595746±0.048406847	1.00000005E-4±3.162278E-4	0.017900001±0.017142864	0.50886905±0.013171588	3.1695722E-4±0.0010023067	5000.0±0.0
18	18	50000	**0.9988114±5.603002E-4**	**0.10978723±0.03851475**	**0.0011885896±5.6030654E-4**	**0.8902128±0.03851476**	**8.3000003E-4±7.071068E-5**	**0.01262±9.05097E-4**	**0.5288547±0.0106429225**	**0.0011885896±5.6030654E-4**	**50000.0±0.0**
29	28	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	50.0±0.0
29	28	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	150.0±0.0
29	28	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	500.0±0.0
29	28	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	1500.0±0.0
29	28	5000	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	5000.0±0.0
29	28	50000	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	50000.0±0.0
29	16	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.152±0.27770486	0.5±0.0	0.0±0.0	50.0±0.0
29	16	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.20533332±0.35796958	0.5±0.0	0.0±0.0	150.0±0.0
29	16	500	1.0±0.0	0.037446808±0.04547023	0.0±0.0	0.9625532±0.04547023	0.0192±0.042932507	0.39040002±0.46515888	0.50976527±0.012138217	0.0±0.0	500.0±0.0
29	16	1500	0.9993661±0.0014174865	0.12595744±0.037663784	6.3391443E-4±0.0014174758	0.87404263±0.0376638	0.023200002±0.027949955	0.43319997±0.25827652	0.5336205±0.010742749	6.3391443E-4±0.0014174758	1500.0±0.0
29	16	5000	0.9992076±0.0011206005	0.35012767±0.12653348	7.92393E-4±0.001120613	0.6498724±0.1265335	0.06016±0.037043866	0.34743997±0.112009585	0.60869753±0.045450617	7.92393E-4±0.001120613	5000.0±0.0
29	**16**	**50000**	**0.9916799±0.0016809427**	**0.99106383±0.007823298**	**0.008320127±0.001680919**	**0.0089361705±0.007823309**	**0.26129±0.035114918**	**0.35898±0.03586444**	**0.99110603±0.007734031**	**0.008320127±0.001680919**	**50000.0±0.0**
29	29	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	50.0±0.0
29	29	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	150.0±0.0
29	29	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	500.0±0.0
29	29	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	1500.0±0.0
29	29	5000	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	5000.0±0.0
29	**29**	**50000**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.5±0.0**	**0.0±0.0**	**50000.0±0.0**

Table 29. *Mushroom Data Set under HD Rule (Part 1)*

r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Population Size
4	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
4	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
4	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
4	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
4	5000	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
4	**50000**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.5±0.0**	**0.0±0.0**	**0.0±0.0**
5	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
5	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
5	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
5	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
5	5000	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
5	**50000**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.5±0.0**	**0.0±0.0**	**0.0±0.0**
8	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
8	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
8	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
8	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
8	5000	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
8	**50000**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.5±0.0**	**0.0±0.0**	**0.0±0.0**
18	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
18	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
18	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
18	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
18	5000	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	0.0±0.0
18	**50000**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.5±0.0**	**0.0±0.0**	**0.0±0.0**
29	50	1.0±0.0	0.03812766±0.082679145	0.0±0.0	0.9618724±0.08267914	0.0±0.0	62.9±136.7995	0.51060414±0.023395646	0.0±0.0	0.5±0.52704626
29	150	0.9998415±3.340985E-4	0.06187234±0.10612967	1.5847861E-4±3.3410222E-4	0.9381277±0.106129676	6.1±19.289892	156.94±286.65704	0.5174333±0.030566	1.5847861E-4±3.3410222E-4	1.2±1.4757297
29	500	0.9984943±0.0013699299	0.10765958±0.12313119	0.0015055469±0.001369921	0.8923405±0.123131186	9.52698±15.847981	100.691444±105.53751	0.53032815±0.038384773	0.0015055469±0.001369921	5.7±2.626785
29	1500	0.995087±0.0026884351	0.448±0.22318155	0.004912837±0.0026884335	0.552±0.22318156	48.574554±33.24388	139.63503±54.149807	0.6582142±0.11844484	0.004912837±0.0026884335	20.4±5.3374977
29	5000	0.98684627±0.0031739732	0.8220426±0.07870021	0.013153724±0.0031739713	0.17795745±0.07870022	94.59834±19.222433	152.4141±29.202215	0.85075986±0.058920704	0.013153724±0.0031739713	62.1±9.338689
29	**50000**	**0.9061014±0.0073791416**	**0.99234045±0.0045743412**	**0.09389858±0.0073791407**	**0.007659574±0.0045743305**	**142.17032±6.71335**	**149.35657±6.6462173**	**0.99165154±0.004916182**	**0.09389858±0.0073791407**	**661.0±23.48522**
32	50	0.99667186±0.0017036012	0.34221274±0.17187156	0.003328051±0.0017035936	0.6577872±0.17187156	20.21659±21.699276	84.29561±42.81381	0.6084197±0.06464365	0.003328051±0.0017035936	24.7±2.0575066
32	150	0.9871632±0.0065427804	0.67668086±0.1772373	0.012836767±0.0065427762	0.32331914±0.1772373	40.806343±14.218592	83.87942±19.621346	0.76639307±0.10823254	0.012836767±0.0065427762	71.1±6.7568893
32	500	0.9643424±0.0076095327	0.9222128±0.06626674	0.03565769±0.007609541	0.077787235±0.06626674	64.56476±10.582784	81.951584±11.776536	0.9287437±0.056296945	0.03565769±0.007609541	241.9±10.556199
32	1500	0.91743267±0.011469495	0.99021274±0.012144038	0.08256735±0.01146949	0.009787234±0.01214403	74.80389±9.445896	81.15986±9.476468	0.98961174±0.012688861	0.08256735±0.01146949	719.3±29.132074
32	5000	0.807607±0.009746409	0.99982977±3.588279E-4	0.19239302±0.009746399	1.7021276E-4±3.5884004E-4	75.5018±3.1938293	77.555046±3.2100976	0.9997899±4.4287893E-4	0.19239302±0.009746399	2426.2±38.159897
32	**50000**	**0.459271±0.012978482**	**1.0±0.0**	**0.540729±0.01297849**	**0.0±0.0**	**77.23062±1.1889819**	**77.44145±1.1872092**	**1.0±0.0**	**0.540729±0.01297849**	**24023.1±157.65958**
36	50	0.996672±0.0022965726	0.24195746±0.18061547	0.0033280507±0.002296573	0.7580425±0.18061547	4.752±5.7030144	26.434002±15.581465	0.57397234±0.06478694	0.0033280507±0.002296573	50.0±0.0
36	150	0.98763865±0.0035672334	0.6181277±0.19238394	0.012361331±0.0035672355	0.38187233±0.19238396	8.786665±5.551827	24.369333±9.803609	0.73455215±0.106470436	0.012361331±0.0035672355	150.0±0.0
36	500	0.96600646±0.005840219	0.9067234±0.06771078	0.033993658±0.005840223	0.0932766±0.06771079	16.004599±4.2424874	23.7896±4.8530383	0.915244±0.05638885	0.033993658±0.005840223	500.0±0.0
36	1500	0.91188586±0.0064892387	0.99387234±0.007941247	0.0881141±0.006489242	0.006127659±0.007941236	20.366266±1.1653075	23.433468±1.1561518	0.9933953±0.008495512	0.0881141±0.006489242	1500.0±0.0
36	5000	0.7867671±0.011570328	1.0±0.0	0.21323295±0.011570324	0.0±0.0	22.591898±1.3866035	23.56524±1.3868679	1.0±0.0	0.21323295±0.011570324	5000.0±0.0
36	**50000**	**0.36053884±0.002241222**	**1.0±0.0**	**0.63946116±0.002241243**	**0.0±0.0**	**23.72004±0.3590979**	**23.81806±0.35725963**	**1.0±0.0**	**0.63946116±0.002241243**	**50000.0±0.0**

Table 30. *Mushroom Data Set under HD Rule (Part 2)*

r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
39	50	0.99865294±0.0013494034	0.080170214±0.10722414	0.0013470681±0.0013493968	0.9198297±0.107224144	0.036±0.040879223	3.72±3.9738476	0.5221044±0.031085536	0.0013470681±0.0013493968	50.0±0.0
39	150	0.9956419±0.002340214	0.18893619±0.16899946	0.004358162±0.0023402069	0.8110639±0.16899945	0.35199997±0.43634617	4.2559996±2.7453957	0.5558071±0.05650185	0.004358162±0.0023402069	150.0±0.0
39	500	0.98518217±0.0042924304	0.5817873±0.10770672	0.01481775±0.0042924336	0.4182128±0.107706726	1.8504±0.76395166	6.401±1.4602013	0.70567435±0.05322618	0.01481775±0.0042924336	500.0±0.0
39	1500	0.95641834±0.009610892	0.89259577±0.06489674	0.043581616±0.009610886	0.107404254±0.06489674	3.6785998±0.5235904	6.2306004±0.65332294	0.9020715±0.053771183	0.043581616±0.009610886	1500.0±0.0
39	5000	0.8778923±0.009614155	0.9982127±0.002487737	0.12210777±0.00961415	0.0017872341±0.0024877347	5.59548±0.6783551	6.5180006±0.6743856	0.99797046±0.002821673	0.12210777±0.00961415	5000.0±0.0
39	**50000**	**0.45404118±0.0011206215**	**1.0±0.0**	**0.54595876±0.0011206425**	**0.0±0.0**	**6.3485603±0.059368726**	**6.44538±0.059453692**	**1.0±0.0**	**0.54595876±0.0011206425**	**50000.0±0.0**
43	50	1.0±0.0	0.005531915±0.01462163	0.0±0.0	0.9944681±0.014621621	0.0±0.0	0.232±0.5224898	0.5014115±0.0037433838	0.0±0.0	50.0±0.0
43	150	0.999683±5.540397E-4	0.009361701±0.014691641	3.1695722E-4±5.540459E-4	0.99063826±0.014691642	0.0053333333±0.010795519	0.47466666±0.33513293	0.5022971±0.0037986836	3.1695722E-4±5.540459E-4	150.0±0.0
43	500	0.9993661±8.1837084E-4	0.09029786±0.06996052	6.3391443E-4±8.1838E-4	0.9097021±0.06996052	0.010999999±0.021791942	0.46260005±0.29809624	0.52412146±0.019324906	6.3391443E-4±8.1838E-4	500.0±0.0
43	1500	0.9953248±0.0024337263	0.16706383±0.07704836	0.004675119±0.0024337347	0.83293617±0.077048354	0.0176±0.006936752	0.3772±0.14222875	0.5452982±0.023049496	0.004675119±0.0024337347	1500.0±0.0
43	5000	0.98391455±0.0026426292	0.36834043±0.12156868	0.016085576±0.0026426306	0.6316596±0.12156868	0.06786±0.02905031	0.36868±0.08531157	0.6122273±0.04750673	0.016085576±0.0026426306	5000.0±0.0
43	**50000**	**0.844691±0.005603086**	**0.99276596±0.0054161227**	**0.15530904±0.0056030652**	**0.0072340425±0.005416137**	**0.28617±0.00504875**	**0.3772±0.0059396923**	**0.991549±0.006248186**	**0.15530904±0.0056030652**	**50000.0±0.0**
46	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.01±0.016996732	0.5±0.0	0.0±0.0	50.0±0.0
46	150	0.9999207±2.5057385E-4	0.0±0.0	7.9239304E-5±2.5057668E-4	1.0±0.0	0.0±0.0	0.014666666±0.03794733	0.49998015±6.2671745E-5	7.9239304E-5±2.5057668E-4	150.0±0.0
46	500	0.9995245±6.68197E-4	0.0010212766±0.002942707	4.7543584E-4±6.6820445E-4	0.99897873±0.0029427016	0.0±0.0	0.0070±0.011085526	0.5001375±8.0129463E-4	4.7543584E-4±6.6820445E-4	500.0±0.0
46	1500	0.999683±5.540397E-4	0.0010212766±0.0026551737	3.1695722E-4±5.540459E-4	0.99897873±0.0026551776	0.0±0.0	0.0064666667±0.009232525	0.500177±6.439526E-4	3.1695722E-4±5.540459E-4	1500.0±0.0
46	5000	0.9989699±0.001059822	0.010382978±0.013680703	0.001030111±0.0010598205	0.989617±0.013680698	0.0±0.0	0.00772±0.0057489704	0.5023732±0.003432526	0.001030111±0.0010598205	5000.0±0.0
46	**50000**	**0.98454833±0.003922144**	**0.16255319±0.03370041**	**0.015451664±0.003922146**	**0.8374468±0.033700366**	**5.4000004E-4±5.6568522E-5**	**0.01164±3.1112673E-4**	**0.5404613±0.009007329**	**0.015451664±0.003922146**	**50000.0±0.0**
50	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	50.0±0.0
50	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	150.0±0.0
50	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	500.0±0.0
50	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	1500.0±0.0
50	5000	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	2.9999999E-4±8.8065624E-4	0.5±0.0	0.0±0.0	5000.0±0.0
50	**50000**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**	**0.0±0.0**	**3.0E-5±4.2426407E-5**	**0.5±0.0**	**0.0±0.0**	**50000.0±0.0**
54	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	50.0±0.0
54	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	150.0±0.0
54	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	500.0±0.0
54	1500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	1500.0±0.0
54	5000	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	5000.0±0.0
54	**50000**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**0.5±0.0**	**0.0±0.0**	**50000.0±0.0**

Table 31. *Mushroom Data Set under RCHK Rule (No MHC)*

r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
4	50	1.0±0.0	0.005957447±0.012559401	0.0±0.0	0.9940426±0.012559404	11.820044±8.701273	27.16063±12.593609	0.5015119±0.003187331	0.0±0.0	18.7±4.083843
4	150	1.0±0.0	0.010978723±0.016353078	0.0±0.0	0.9890213±0.016353084	24.111988±7.4141755	32.74413±8.527584	0.50279063±0.0041618785	0.0±0.0	55.9±3.6040103
4	500	1.0±0.0	0.03182979±0.0032888218	0.0±0.0	0.9681703±0.0032888271	26.686655±1.8892406	32.200867±3.226187	0.5080874±8.501924E-4	0.0±0.0	187.0±16.586473
4	1500	1.0±0.0	0.13021275±0.3056308	0.0±0.0	0.8697872±0.3056308	29.546057±2.0157392	33.3054±3.2385743	0.55768347±0.15541664	0.0±0.0	566.1±22.417503
4	5000	1.0±0.0	0.22927657±0.4062069	0.0±0.0	0.7707234±0.40620688	31.859491±1.68345	33.603176±1.9359661	0.6074556±0.20688906	0.0±0.0	1889.0±28.067379
4	**50000**	**1.0±0.0**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**33.215576±0.34709042**	**33.4862±0.32766646**	**1.0±0.0**	**0.0±0.0**	**18957.0±281.4285**
5	50	1.0±0.0	0.14936168±0.3084011	0.0±0.0	0.8506383±0.3084011	23.21824±7.961108	66.52985±26.206745	0.56346846±0.15497707	0.0±0.0	47.7±0.94868326
5	150	1.0±0.0	0.15668085±0.30556405	0.0±0.0	0.8433191±0.30556405	39.34468±9.48368	59.35299±14.514336	0.56539446±0.15428439	0.0±0.0	143.0±2.309401
5	500	1.0±0.0	0.16468084±0.30999324	0.0±0.0	0.83531916±0.30999324	48.141735±6.4522495	55.3899±7.1286464	0.5682105±0.15482424	0.0±0.0	468.7±6.600505
5	1500	1.0±0.0	0.6450213±0.37493983	0.0±0.0	0.35497874±0.37493983	53.98475±5.5728965	57.49181±5.95905	0.7925104±0.2188744	0.0±0.0	1409.8±11.487193
5	5000	1.0±0.0	0.90161705±0.23492794	0.0±0.0	0.09838298±0.23492791	56.21656±2.4059608	57.53488±2.476286	0.9374382±0.14134353	0.0±0.0	4709.2±17.637081
5	**50000**	**1.0±0.0**	**1.0±0.0**	**0.0±0.0**	**0.0±0.0**	**56.23073±0.10749402**	**56.37847±0.10102296**	**1.0±0.0**	**0.0±0.0**	**47085.5±119.501045**
8	50	1.0±0.0	0.40068084±0.42409757	0.0±0.0	0.5993191±0.42409757	9.33±9.75669	48.344±20.36707	0.6759914±0.21414967	0.0±0.0	50.0±0.0
8	150	1.0±0.0	0.81387234±0.21457614	0.0±0.0	0.18612766±0.21457614	37.117336±17.594069	70.51867±23.256191	0.86722696±0.14873143	0.0±0.0	150.0±0.0
8	500	1.0±0.0	0.992±0.025298215	0.0±0.0	0.007999999±0.025298221	49.806±7.7022357	61.708202±7.8795094	0.99259263±0.023424285	0.0±0.0	500.0±0.0
8	1500	1.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	52.57973±6.066954	56.741596±6.0767136	1.0±0.0	0.0±0.0	1500.0±0.0
8	5000	0.9993661±0.00200461	1.0±0.0	6.3391443E-4±0.0020046134	0.0±0.0	54.250843±3.1516454	55.532063±3.1428046	1.0±0.0	6.3391443E-4±0.0020046134	5000.0±0.0
8	**50000**	**0.99841523±0.0**	**1.0±0.0**	**0.0015847861±0.0**	**0.0±0.0**	**56.68237±2.086346**	**56.814873±2.089429**	**1.0±0.0**	**0.0015847861±0.0**	**50000.0±0.0**
18	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	50.0±0.0
18	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.112±0.3021699	0.5±0.0	0.0±0.0	150.0±0.0
18	500	1.0±0.0	0.0029787235±0.00941955	0.0±0.0	0.9970213±0.009419553	0.0±0.0	0.0936±0.15197894	0.5007559±0.0023904983	0.0±0.0	500.0±0.0
18	1500	1.0±0.0	0.108±0.14745958	0.0±0.0	0.8920001±0.1474596	0.011066666±0.020277578	0.20533332±0.17981058	0.5317087±0.04524287	0.0±0.0	1500.0±0.0
18	5000	1.0±0.0	0.24723406±0.15515591	0.0±0.0	0.75276595±0.15515591	0.0077799996±0.008627321	0.15652±0.07536956	0.57475±0.05312338	0.0±0.0	5000.0±0.0
18	**50000**	**1.0±0.0**	**0.8506383±0.14382853**	**0.0±0.0**	**0.1493617±0.14382853**	**0.0673±0.008796406**	**0.14633±0.017267544**	**0.87691414±0.10973501**	**0.0±0.0**	**50000.0±0.0**
29	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	50.0±0.0
29	150	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	150.0±0.0
29	500	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	500.0±0.0

Table 32. *Mushroom Data Set under RCHK (Global MHC) Rule*

r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
4	50	0.70990497±0.27624604	0.5445958±0.36513105	0.29009512±0.27624607	0.45540422±0.36513105	16.92912±11.002508	101.452286±52.658417	0.68541604±0.22108233	0.29009512±0.27624607	19.9±3.2472212
4	150	0.14690967±0.094811775	0.97744673±0.015611872	0.8530903±0.094811775	0.022553192±0.01561186	1111.3855±210.99782	1231.9104±215.58998	0.7898246±0.29386997	0.8530903±0.094811775	54.8±7.3756356
4	500	0.027812997±0.0432134	0.988766±0.005023767	0.97218704±0.043213397	0.011234042±0.0050237603	982.76044±62.743404	1018.00684±63.51552	0.42160597±0.3468847	0.97218704±0.043213397	190.0±9.637888
4	1500	0.058399368±0.055331573	0.9297021±0.11658104	0.9416007±0.055331573	0.07029787±0.11658104	96.980995±20.588842	107.40198±21.629555	0.53981036±0.31740895	0.9416007±0.055331573	569.9±20.957365
4	5000	0.0±0.0	1.0±0.0	1.0±0.0	0.0±0.0	105.1542±5.777478	108.62724±5.832725	0.0±0.0	1.0±0.0	1880.6±42.172134
4	**50000**	**0.0±0.0**	**1.0±0.0**	**1.0±0.0**	**0.0±0.0**	**154.08511±0.6530402**	**154.4632±0.64528245**	**0.0±0.0**	**1.0±0.0**	**18785.5±164.75587**
5	50	0.0381141±0.04025282	0.9706383±0.074736826	0.9618858±0.040252827	0.0293617±0.07473683	226.81345±84.86678	341.96255±94.05652	0.71001804±0.36689875	0.9618858±0.040252827	47.5±1.7795131
5	150	0.001030111±0.0023639358	0.99991494±2.69121E-4	0.99896985±0.002363932	8.510638E-5±2.6913002E-4	441.04184±65.60221	485.5606±67.219345	0.2±0.42163703	0.99896985±0.002363932	141.4±1.9550506
5	500	0.0±0.0	1.0±0.0	1.0±0.0	0.0±0.0	440.28973±21.83272	454.9833±21.75868	0.0±0.0	1.0±0.0	471.9±4.0400767
5	1500	0.0±0.0	1.0±0.0	1.0±0.0	0.0±0.0	656.0951±19.86889	660.99884±19.881073	0.0±0.0	1.0±0.0	1414.4±7.7201037
5	5000	0.0±0.0	1.0±0.0	1.0±0.0	0.0±0.0	275.06488±9.025789	276.41937±9.001372	0.0±0.0	1.0±0.0	4701.5±14.683702
5	**50000**	**0.0±0.0**	**1.0±0.0**	**1.0±0.0**	**0.0±0.0**	**233.15485±2.7620957**	**233.29616±2.7669837**	**0.0±0.0**	**1.0±0.0**	**47089.332±36.82843**
8	50	0.11259905±0.070906706	0.8765957±0.14062856	0.8874009±0.0709067	0.12340425±0.14062856	66.708±30.134165	150.20999±42.4026	0.615578±0.30520397	0.8874009±0.0709067	50.0±0.0
8	150	0.0019017432±0.002152303	0.99965954±0.0010765215	0.99809825±0.0021523053	3.4042553E-4±0.0010765201	150.59465±22.837961	187.80933±23.585642	0.5537856±0.49693552	0.99809825±0.0021523053	150.0±0.0
8	500	0.0±0.0	1.0±0.0	1.0±0.0	0.0±0.0	143.458±13.758787	154.9164±13.665003	0.0±0.0	1.0±0.0	500.0±0.0
8	1500	0.0±0.0	1.0±0.0	1.0±0.0	0.0±0.0	164.68521±8.0457	168.52234±8.007914	0.0±0.0	1.0±0.0	1500.0±0.0
8	5000	0.0±0.0	1.0±0.0	1.0±0.0	0.0±0.0	113.90718±3.6696684	115.11122±3.6520426	0.0±0.0	1.0±0.0	5000.0±0.0
8	**50000**	**0.0±0.0**	**1.0±0.0**	**1.0±0.0**	**0.0±0.0**	**149.77882±0.59936**	**149.89317±0.5968308**	**0.0±0.0**	**1.0±0.0**	**50000.0±0.0**
18	50	0.99896985±0.002746207	0.0±0.0	0.001030111±0.0027462004	1.0±0.0	0.0±0.0	0.0±0.0	0.49974146±6.89525E-4	0.001030111±0.0027462004	50.0±0.0
18	150	0.9987322±0.0017991982	1.7021276E-4±5.3826004E-4	0.0012678287±0.0017991955	0.99982977±5.3826073E-4	0.0±0.0	0.06533333±0.20196074	0.49972504±4.8232905E-4	0.0012678287±0.0017991955	150.0±0.0
18	500	0.9749603±0.024093037	0.01412766±0.02100134	0.025039619±0.024093036	0.9858724±0.021001346	4.0000002E-4±0.0012649111	0.13159999±0.13975789	0.4972009±0.0049879565	0.025039619±0.024093036	500.0±0.0
18	1500	0.9534073±0.030417066	0.08417021±0.05339886	0.046592712±0.03041707	0.9158298±0.05339886	0.0123333335±0.020618223	0.21626666±0.17282408	0.51030326±0.017482562	0.046592712±0.03041707	1500.0±0.0
18	5000	0.8317749±0.03677032	0.19497873±0.109347224	0.16822504±0.036770318	0.8050213±0.109347224	0.016320001±0.011950343	0.17977999±0.058583535	0.5100225±0.03709205	0.16822504±0.036770318	5000.0±0.0
18	**50000**	**0.123085044±0.018075129**	**0.89191484±0.02181446**	**0.876915±0.018075144**	**0.1080851±0.02181448**	**0.09216666±0.03613454**	**0.17748±0.044095937**	**0.53332764±0.075637914**	**0.876915±0.018075144**	**50000.0±0.0**
29	50	1.0±0.0	0.0±0.0	0.0±0.0	1.0±0.0	0.0±0.0	0.0±0.0	0.5±0.0	0.0±0.0	50.0±0.0

Table 33. *Mushroom Data Set under RCHK (MHC) Rule*

r	Nc	TP	TN	FP	FN	OC	GC	DR	FR	Actual Population Size
4	50	1.0±0.0	0.6174468±0.28208262	0.0±0.0	0.3825532±0.28208265	24.327267±19.269402	89.47553±38.949303	0.7492238±0.1429538	0.0±0.0	39.6±1.8973665
4	150	1.0±0.0	0.9073192±0.11863885	0.0±0.0	0.09268086±0.118638866	65.4457±28.484499	103.8782±30.392515	0.92369175±0.0875852	0.0±0.0	122.1±6.0635524
4	500	1.0±0.0	0.99914896±0.0021605056	0.0±0.0	8.510638E-4±0.002160503	97.72725±19.30461	112.11742±19.27626	0.9991538±0.002146279	0.0±0.0	400.7±7.4840574
4	1500	0.9981775±0.0023041544	1.0±0.0	0.001822504±0.0023041554	0.0±0.0	93.36151±8.41909	98.305786±8.486203	1.0±0.0	0.0±0.1	1215.0±17.416468
4	5000	0.9982567±0.0018973408	1.0±0.0	0.0017432647±0.0018973359	0.0±0.0	101.046585±5.909478	102.55873±5.902949	1.0±0.0	0.0017432647±0.0018973359	4017.5±33.76471
4	**50000**	**0.9949815±0.002420806**	**1.0±0.0**	**0.005018489±0.002420801**	**0.0±0.0**	**100.44702±0.7921632**	**100.59853±0.78638285**	**1.0±0.0**	**0.005018489±0.002420801**	**40346.332±201.51758**
5	50	1.0±0.0	0.61489356±0.12761371	0.0±0.0	0.38510638±0.12761372	21.741444±14.393579	78.69987±27.537529	0.72776026±0.07002552	0.0±0.0	48.1±1.3703203
5	150	1.0±0.0	0.94399995±0.04487617	0.0±0.0	0.056±0.04487617	43.850758±10.957314	75.48036±12.248281	0.948476±0.039397586	0.0±0.0	145.2±2.1499352
5	500	0.99976236±7.517216E-4	0.9995745±0.0013456425	2.377179E-4±7.517299E-4	4.255319E-4±0.0013456498	77.02253±13.156308	88.879395±13.207469	0.9995763±0.0013399313	2.377179E-4±7.517299E-4	481.3±3.9735236
5	1500	0.99659264±0.0040239925	1.0±0.0	0.0034072902±0.00402399	0.0±0.0	86.380936±6.708734	90.486496±6.7010803	1.0±0.0	0.0034072902±0.00402399	1447.3±6.650814
5	5000	0.9964341±0.0031750817	1.0±0.0	0.0035657685±0.0031750698	0.0±0.0	86.541336±2.693077	87.76869±2.7087092	1.0±0.0	0.0035657685±0.0031750698	4827.4±14.017449
5	**50000**	**0.9849445±0.0039619803**	**1.0±0.0**	**0.015055466±0.003961965**	**0.0±0.0**	**88.83602±0.33894545**	**88.95706±0.33889985**	**1.0±0.0**	**0.015055466±0.003961965**	**48295.668±56.074356**
8	50	0.9992076±0.0020115573	0.48561698±0.2925818	7.923931E-4±0.0020115618	0.51438296±0.2925818	16.466±16.637562	57.286±31.248003	0.6840696±0.13930975	7.923931E-4±0.0020115618	50.0±0.0
8	150	0.9988907±0.0018375698	0.84434044±0.07708011	0.0011093502±0.0018375622	0.15565959±0.07708011	21.06±10.53324	46.239998±12.71232	0.86878765±0.06007408	0.0011093502±0.0018375622	150.0±0.0
8	500	0.997385±0.0027714975	0.9994043±0.0012718553	0.0026148972±0.0027714882	5.957447E-4±0.0012718589	39.848797±6.3938384	50.4896±6.712111	0.99940443±0.0012718834	0.0026148972±0.0027714882	500.0±0.0
8	1500	0.9969889±0.0022965749	1.0±0.0	0.0030110935±0.002296573	0.0±0.0	45.622936±3.5570722	49.294266±3.551394	1.0±0.0	0.0030110935±0.002296573	1500.0±0.0
8	5000	0.99072903±0.005296477	1.0±0.0	0.009270998±0.0052964687	0.0±0.0	47.01552±3.0634825	48.116478±3.0616374	1.0±0.0	0.009270998±0.0052964687	5000.0±0.0
8	**50000**	**0.95853144±0.004364171**	**1.0±0.0**	**0.04146857±0.0043641604**	**0.0±0.0**	**49.2777±0.42073905**	**49.389484±0.4177469**	**1.0±0.0**	**0.04146857±0.0043641604**	**50000.0±0.0**
18	50	1.0±0.0	0.0013617021±0.00430608	0.0±0.0	0.99863833±0.004306086	0.0±0.0	0.392±0.8039459	0.5003427±0.0010839101	0.0±0.0	50.0±0.0
18	150	1.0±0.0	0.002212766±0.0069973804	0.0±0.0	0.9977872±0.0069973893	0.0±0.0	0.076±0.13627134	0.5005594±0.0017689075	0.0±0.0	150.0±0.0
18	500	0.9998415±5.0114776E-4	0.005957447±0.008309684	1.5847861E-4±5.0115335E-4	0.9940426±0.008309684	0.0±0.0	0.1576±0.14110924	0.50146204±0.0021334544	1.5847861E-4±5.0115335E-4	500.0±0.0

6.2.10 Mushroom Data Set Conclusion

The mushroom data set, re-formatted as a set of binary strings, is by far the most complex data set, in that it comprises 58 attributes and contains a large number of both self (4208) and non-self data (3916). The results of the best performing test group for each scenario are summarised in Figure 46. Although the feature-detection rule, RCHK (No MHC) rule and RCHK (MHC) rule perform equally well under the mushroom data set, the RCHK

(MHC) rule and feature-detection rule are the best performing rules because they both reach their peak performance at an average population size of 5000 detectors. That is the performance (DR and FR values) exhibited by the feature-detection rule and RCHK (MHC) rule do not improve or degrade after an average population size of 5000 is reached. The RCHK rule (Global MHC) has the worst performance followed by the HD rule.

A comparison between GC and OC exhibited by the best performing test group for each scenario is summarised in Figure 47. The HD rule has the greatest OC and GC values and the RCHK (Global MHC) rule has the lowest OC and GC values. The HD rule is the best performing rule, with reference to generalisation and overfitting, because it has the greatest GC minus OC value. The feature-detection rule is consequently the worst performing rule because it has the lowest GC minus OC value.

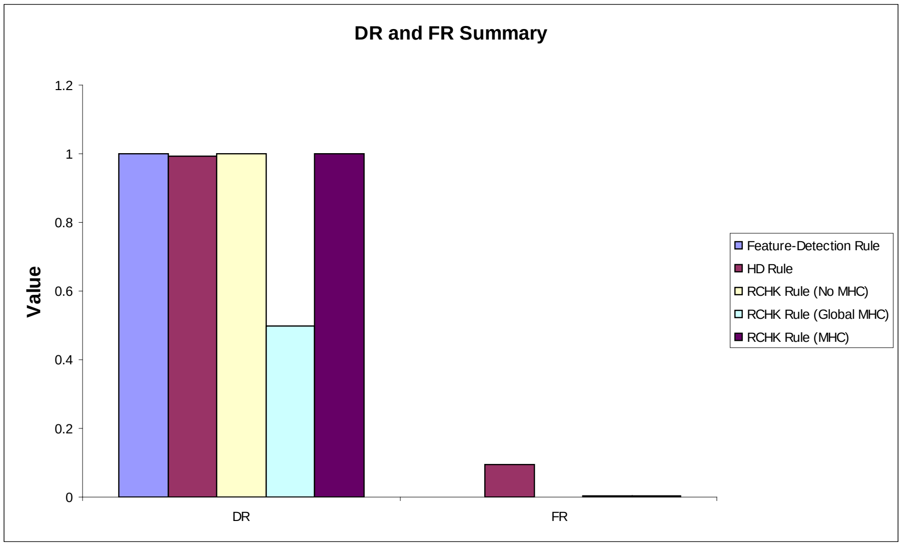


Figure 46. *Mushroom Data Set: DR and FR Summary*

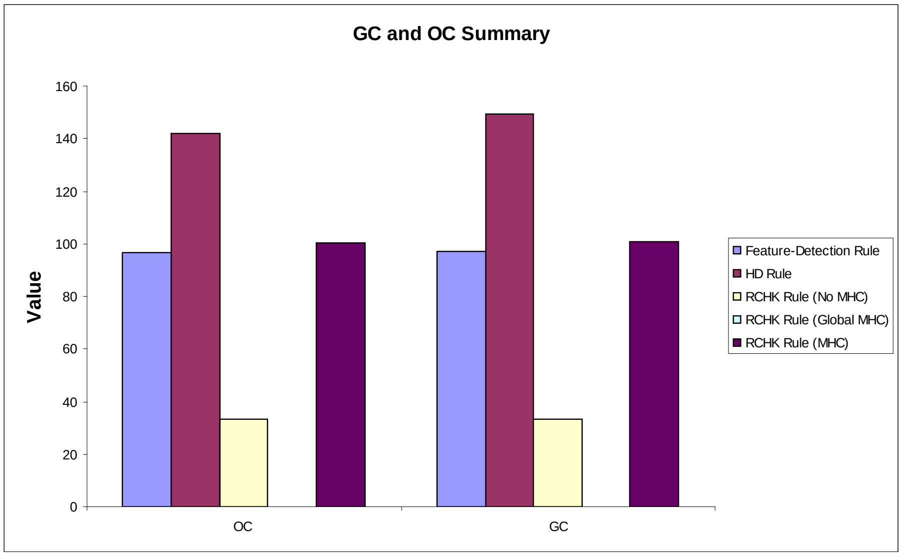


Figure 47. *Mushroom Data Set: GC and OC Summary*

6.2.11 Benchmarking the number of detectors generated for each experiment

The actual number of detectors generated under each experiment, in order to exhibit satisfactory DR and FR values, can be benchmarked against a theoretical number of detectors by utilising Forret *et al.*'s [33] original equations to calculate the number of candidate detectors, n_r (equation 4.6), needed to generate, n_c (equation 4.7) detectors given a matching probability, P_M and a failure probability, P_f.

By fixing P_f at 0.05, P_M, n_c and n_r were calculated for the feature-detection rule, HD rule and RCHK rule respectively for each experiment and are presented below. The rows which correlate to the best performing test groups are highlighted using the same conventions followed in this chapter (apart from the worst performing test groups, which are not shown in the tables below). In order to collate the rows pertaining to best performing test groups for the RCHK (No MHC), RHCK (Global MHC) and RHCK (MHC) into a single table, an addition column termed "Best Under" was added to the tables relating to the RCHK rule for each experiment. Equation (5.1), equation (5.3) and equation (4.2) were used to calculate P_M for the feature-detection, RCHK and HD rule respectively. By comparing the theoretical number of detectors to the actual number of detectors generated under each experiment the following can be observed:

- The theoretical n_r calculated for the Car Evaluation Experiment (Table 34, Table 35 and Table 36) is large for all of the rules, whereas the theoretical n_c is much smaller than the experimental/actual n_c, 5000, needed by the best performing target test groups to exhibit satisfactory DR and FR values . The theoretical n_r for all of the rules is much greater, than the stopping condition employed by the NSA algorithm, implemented in this chapter i.e. the algorithm will only attempt to generate a candidate detector at most $1000 \times n_c$ times, after which, it will terminate. The stopping condition (indicated when the target n_c is not equal to the actual population size) was true for the HD, RCHK (No MHC), RCHK (Global MHC) and RCHK (MHC) rules for the Car Evaluation Experiment (see Table 3, Table 4, Table 5 and Table 6 respectively).

- The theoretical n_r calculated for the Iris Experiment (Table 37, Table 38 and Table 39) is large for all of the rules, whereas the theoretical n_c is much smaller than the actual n_c, 5000, needed by the best performing test groups with the exception of the RCHK (MHC) and RCHK (Global MHC) which have a theoretical n_c equal to 6282655.717. The stopping condition was true for the HD and RCHK (No MHC) rules (see Table 9 and Table 10 respectively).

- The theoretical n_r calculated for the Wisconsin Breast Cancer Experiment (Table 40, Table 41 and Table 42) is large (much larger than the stopping condition) for all of the rules, whereas the theoretical n_c is much smaller than the actual n_c needed by the best performing test groups with the exception of the RCHK (Global MHC). The stopping condition was true for the feature-detection rule, HD rule, RCHK (No MHC) and RCHK (MHC) rules (see Table 13, Table 14, Table 15, Table 16, Table 17 and Table 19 respectively).

- The theoretical n_r calculated for the Glass Experiment (Table 43, Table 44 and Table 45) and the theoretical n_c is large (much larger than the stopping condition) for all of the rules. Interestingly the theoretical n_c is significantly larger than the actual n_c, needed by the best performing test groups. The stopping condition was false for any of the rules in the Glass Experiment.

- The theoretical n_r calculated for the Mushroom Experiment (Table 46, Table 47 and Table 48) is once again large (with some of the results being reported as “Infinity” meaning that they could not be computed) for all of the rules, whereas the theoretical n_c is much smaller than the actual n_c, needed by the best performing test groups with the exception of the RCHK (Global MHC) rule. The stopping condition was true for the feature-detection, HD, RCHK (No MHC) and RCHK (MHC) rules (see Table 27, Table 28, Table 29, Table 30, Table 31 and Table 33 respectively).

Table 34. *Car Evaluation Experiment – Feature-Detection Rule Benchmark*

n'	*r*	n_r	n_c	P_m	P_f	S
3	2	1.93E+79	7.988619	0.375	0.05	384
3	3	4.45E+23	23.96586	0.125	0.05	384
5	3	1.14E+49	11.98293	0.25	0.05	384
5	4	8.34E+17	31.95448	0.09375	0.05	384
5	5	1.89E+07	95.86343	0.03125	0.05	384
6	4	4.45E+23	23.96586	0.125	0.05	384
6	**5**	**6.49E+09**	**63.908955**	**0.046875**	**0.05**	**384**
6	6	81100.21243	191.7269	0.015625	0.05	384
10	3	3.90E+138	5.325746	0.5625	0.05	384
10	10	4464.192968	3067.63	9.77E-04	0.05	384
13	5	4.14E+29	19.17269	0.15625	0.05	384
13	13	25718.86141	24541.039	0.000122	0.05	384

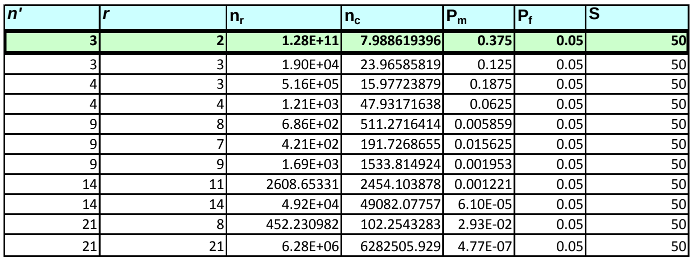
Table 35. *Car Evaluation Experiment – HD Rule Benchmark*

r	n_r	n_c	P_m	P_f	S
3	Infinity	3.029757875	0.98877	0.05	384
5	Infinity	3.456971233	0.866577	0.05	384
6	5.78E+206	4.222477423	0.709473	0.05	384
9	**1.71E+25**	**22.45291746**	**0.1334229**	**0.05**	**384**
10	4.91E+09	64.92338303	0.046143	0.05	384
13	25718.8614	24541.03878	0.0001221	0.05	384

Table 36. *Car Evaluation Experiment – RCHK Rule Benchmark*

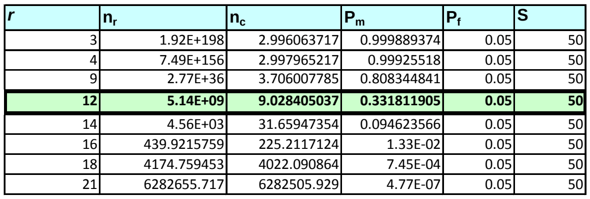
r	n_r	n_c	P_m	P_f	S	Best Under
3	**4.45E+23**	**23.96585819**	**0.125**	**0.05**	**384**	**RCHK (No MHC) and RCHK (Global MHC)**
5	**1.89E+07**	**95.86343275**	**0.03125**	**0.05**	**384**	**RCHK (MHC)**
6	81100.212	191.7268655	0.015625	0.05	384	N/A
10	4464.193	3067.629848	9.77E-04	0.05	384	N/A
13	25718.861	24541.03878	1.22E-04	0.05	384	N/A

Table 37. Iris Experiment – Feature-Detection Rule Benchmark



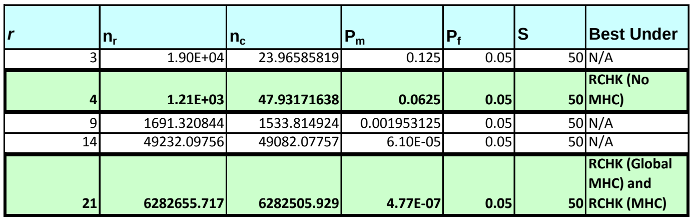
n'	*r*	n_r	n_c	P_m	P_f	S
3	**2**	**1.28E+11**	**7.988619396**	**0.375**	**0.05**	**50**
3	3	1.90E+04	23.96585819	0.125	0.05	50
4	3	5.16E+05	15.97723879	0.1875	0.05	50
4	4	1.21E+03	47.93171638	0.0625	0.05	50
9	8	6.86E+02	511.2716414	0.005859	0.05	50
9	7	4.21E+02	191.7268655	0.015625	0.05	50
9	9	1.69E+03	1533.814924	0.001953	0.05	50
14	11	2608.65331	2454.103878	0.001221	0.05	50
14	14	4.92E+04	49082.07757	6.10E-05	0.05	50
21	8	452.230982	102.2543283	2.93E-02	0.05	50
21	21	6.28E+06	6282505.929	4.77E-07	0.05	50

Table 38. *Iris Experiment – HD Rule Benchmark*



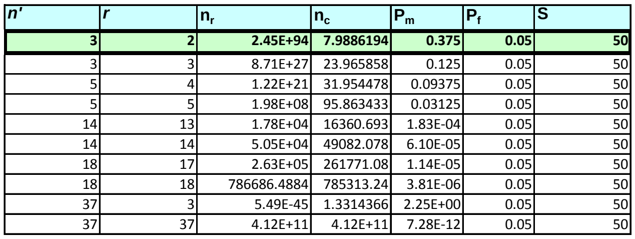
r	n_r	n_c	P_m	P_f	S
3	1.92E+198	2.996063717	0.999889374	0.05	50
4	7.49E+156	2.997965217	0.99925518	0.05	50
9	2.77E+36	3.706007785	0.808344841	0.05	50
12	**5.14E+09**	**9.028405037**	**0.331811905**	**0.05**	**50**
14	4.56E+03	31.65947354	0.094623566	0.05	50
16	439.9215759	225.2117124	1.33E-02	0.05	50
18	4174.759453	4022.090864	7.45E-04	0.05	50
21	6282655.717	6282505.929	4.77E-07	0.05	50

Table 39. *Iris – RCHK Rule Benchmark*



r	n_r	n_c	P_m	P_f	S	Best Under
3	1.90E+04	23.96585819	0.125	0.05	50	N/A
4	**1.21E+03**	**47.93171638**	**0.0625**	**0.05**	**50**	**RCHK (No MHC)**
9	1691.320844	1533.814924	0.001953125	0.05	50	N/A
14	49232.09756	49082.07757	6.10E-05	0.05	50	N/A
21	**6282655.717**	**6282505.929**	**4.77E-07**	**0.05**	**50**	**RCHK (Global MHC) and RCHK (MHC)**

Table 40. *Wisconsin Breast Cancer Experiment – Feature-Detection Rule Benchmark*



n'	*r*	n_r	n_c	P_m	P_f	S
3	**2**	**2.45E+94**	**7.9886194**	**0.375**	**0.05**	**50**
3	3	8.71E+27	23.965858	0.125	0.05	50
5	4	1.22E+21	31.954478	0.09375	0.05	50
5	5	1.98E+08	95.863433	0.03125	0.05	50
14	13	1.78E+04	16360.693	1.83E-04	0.05	50
14	14	5.05E+04	49082.078	6.10E-05	0.05	50
18	17	2.63E+05	261771.08	1.14E-05	0.05	50
18	18	786686.4884	785313.24	3.81E-06	0.05	50
37	3	5.49E-45	1.3314366	2.25E+00	0.05	50
37	37	4.12E+11	4.12E+11	7.28E-12	0.05	50

Table 41. *Wisconsin Breast Cancer Experiment – HD Rule Benchmark*

r	n_r	n_c	P_m	P_f	S
3	Infinity	2.995732289	1	0.05	50
5	Infinity	2.995733898	0.9999995	0.05	50
14	Infinity	3.151530964	0.9505641	0.05	50
18	**4.82E+197**	**4.765832378**	**0.6285853**	**0.05**	**50**
22	2.64E+36	18.49168369	0.1620043	0.05	50
26	30297.18702	298.4710942	1.00E-02	0.05	50
28	4205.643742	2337.486683	1.28E-03	0.05	50
30	32758.95201	31356.37754	9.55E-05	0.05	50
37	4.12E+11	4.12E+11	7.28E-12	0.05	50

Table 42. *Wisconsin Breast Cancer Experiment – RCHK Rule Benchmark*

r	n_r	n_c	P_m	P_f	S	Best Under
3	**8.71E+27**	**23.96585819**	**0.125**	**0.05**	**50**	**RCHK (MHC)**
5	**1.98E+08**	**95.86343275**	**0.03125**	**0.05**	**50**	**RCHK (No MHC)**
14	50473.52311	49082.07757	6.10E-05	0.05	50	N/A
18	786686.4884	785313.2411	3.81E-06	0.05	50	N/A
37	**4.12E+11**	**4.12E+11**	**7.28E-12**	**0.05**	**50**	**RCHK (Global MHC)**

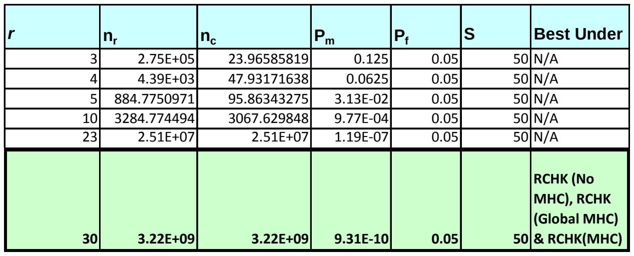
Table 43. *Glass Experiment – Feature-Detection Rule Benchmark*

n'	*r*	n_r	n_c	P_m	P_f	S
3	2	1.55E+15	7.9886194	0.375	0.05	50
3	3	2.75E+05	23.965858	0.125	0.05	50
4	3	3.28E+07	15.977239	0.1875	0.05	50
4	4	4.39E+03	47.931716	0.0625	0.05	50
5	4	3.14E+04	31.954478	9.38E-02	0.05	50
5	5	8.85E+02	95.863433	3.13E-02	0.05	50
10	9	1.26E+03	1022.5433	2.93E-03	0.05	50
10	10	3284.774494	3067.6298	9.77E-04	0.05	50
23	22	8.38E+06	8376674.6	3.58E-07	0.05	50
23	**23**	**2.51E+07**	**2.51E+07**	**1.19E-07**	**0.05**	**50**
30	29	1.07E+09	1.07E+09	2.79E-09	0.05	50
30	30	3.22E+09	3.22E+09	9.31E-10	0.05	50

Table 44. *Glass Experiment – HD Rule Benchmark*

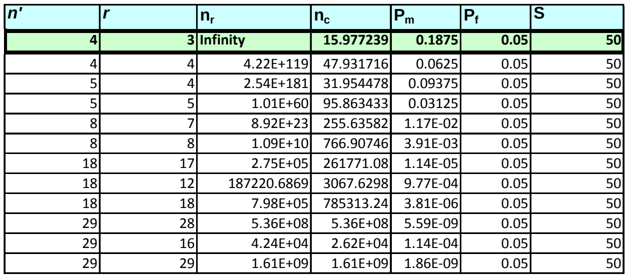
r	n_r	n_c	P_m	P_f	S
3	Infinity	2.995732274	1	0.05	50
4	Infinity	2.995732274	1	0.05	50
5	Infinity	2.995732281	1	0.05	50
10	Infinity	2.995793029	0.9999797	0.05	50
20	1.05E+58	3.52853302	0.8490022	0.05	50
23	3.84E+25	5.363869848	5.59E-01	0.05	50
26	1.22E+09	12.98565031	2.31E-01	0.05	50
30	754.7743983	110.7970452	2.70E-02	0.05	50
34	3.87E+03	3.65E+03	8.21E-04	0.05	50
37	1.48E+05	1.48E+05	2.03E-05	0.05	50
40	**1.93E+07**	**1.93E+07**	**1.55E-07**	**0.05**	**50**
43	1.30E+10	1.30E+10	2.31E-10	0.05	50

Table 45. *Glass Experiment – RCHK Rule Benchmark*



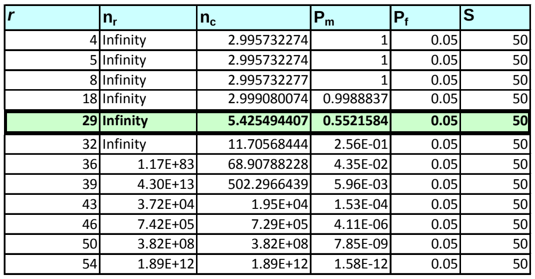
r	n_r	n_c	P_m	P_f	S	Best Under
3	2.75E+05	23.96585819	0.125	0.05	50	N/A
4	4.39E+03	47.93171638	0.0625	0.05	50	N/A
5	884.7750971	95.86343275	3.13E-02	0.05	50	N/A
10	3284.774494	3067.629848	9.77E-04	0.05	50	N/A
23	2.51E+07	2.51E+07	1.19E-07	0.05	50	N/A
30	**3.22E+09**	**3.22E+09**	**9.31E-10**	**0.05**	**50**	**RCHK (No MHC), RCHK (Global MHC) & RCHK(MHC)**

Table 46. *Mushroom Experiment – Feature-Detection Rule Benchmark*



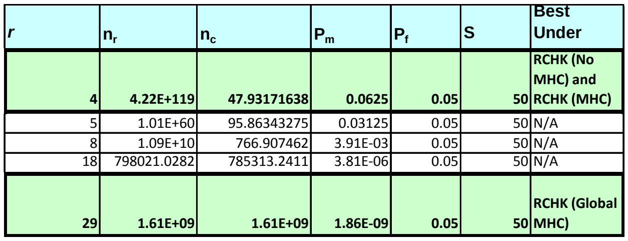
n'	*r*	n_r	n_c	P_m	P_f	S
4	**3**	**Infinity**	**15.977239**	**0.1875**	**0.05**	**50**
4	4	4.22E+119	47.931716	0.0625	0.05	50
5	4	2.54E+181	31.954478	0.09375	0.05	50
5	5	1.01E+60	95.863433	0.03125	0.05	50
8	7	8.92E+23	255.63582	1.17E-02	0.05	50
8	8	1.09E+10	766.90746	3.91E-03	0.05	50
18	17	2.75E+05	261771.08	1.14E-05	0.05	50
18	12	187220.6869	3067.6298	9.77E-04	0.05	50
18	18	7.98E+05	785313.24	3.81E-06	0.05	50
29	28	5.36E+08	5.36E+08	5.59E-09	0.05	50
29	16	4.24E+04	2.62E+04	1.14E-04	0.05	50
29	29	1.61E+09	1.61E+09	1.86E-09	0.05	50

Table 47. *Mushroom Experiment – HD Rule Benchmark*



r	n_r	n_c	P_m	P_f	S
4	Infinity	2.995732274	1	0.05	50
5	Infinity	2.995732274	1	0.05	50
8	Infinity	2.995732277	1	0.05	50
18	Infinity	2.999080074	0.9988837	0.05	50
29	**Infinity**	**5.425494407**	**0.5521584**	**0.05**	**50**
32	Infinity	11.70568444	2.56E-01	0.05	50
36	1.17E+83	68.90788228	4.35E-02	0.05	50
39	4.30E+13	502.2966439	5.96E-03	0.05	50
43	3.72E+04	1.95E+04	1.53E-04	0.05	50
46	7.42E+05	7.29E+05	4.11E-06	0.05	50
50	3.82E+08	3.82E+08	7.85E-09	0.05	50
54	1.89E+12	1.89E+12	1.58E-12	0.05	50

Table 48. *Mushroom Experiment – RCHK Rule Benchmark*



r	n_r	n_c	P_m	P_f	S	Best Under
4	**4.22E+119**	**47.93171638**	**0.0625**	**0.05**	**50**	**RCHK (No MHC) and RCHK (MHC)**
5	1.01E+60	95.86343275	0.03125	0.05	50	N/A
8	1.09E+10	766.907462	3.91E-03	0.05	50	N/A
18	798021.0282	785313.2411	3.81E-06	0.05	50	N/A
29	**1.61E+09**	**1.61E+09**	**1.86E-09**	**0.05**	**50**	**RCHK (Global MHC)**

An interesting point that can be raised when comparing the theoretical n_c values to the actual n_c values, is that the theoretical n_c values assume that each resultant detector, not activated by self, which is added to the repertoire of detectors, C, is optimally placed within the shape space, V.

This implies that there can be no margin for errors, such as overfitting. In reality, however this is not the case i.e. if a randomly generated detector is not activated by self and is therefore added to C, there is no guarantee that the detector is actually optimal. This is illustrated by the degree of overfitting exhibited by the resultant detectors generated under each experiment. Furthermore the theoretical number of detectors, n_r needed to generate n_c optimal detectors is disproportionally large, rendering such an exercise impractical. Generalisation and overfitting are discussed in more detail in the next section.

6.2.12 Generalisation and Overfitting exhibited within the Data Sets

In addition to studying the performance of the feature-detection rule under a number of different conditions and data sets, the tests performed in this chapter also undertook to study the overfitting and generalisation behaviour exhibited by the different detection rules. For each simulation, the GC and OC counts were calculated using the algorithms described in section 3.5.2 and section 3.5.3, respectively. The rate at which both the OC and GC values increased as the number of detectors were increased within the best test groups was observed and the following was discovered:

- for all detection rules, a general trend was observed whereby the rate of the OC count increased as the number of detectors within each test group were increased. This indicates that the degree of overlapping between detectors increased as the number of candidate detectors, N_c were increased; and
- for all detection rules, a general trend was observed whereby the rate of the GC count fluctuated slightly around a singleton value as the number of detectors within each test group were increased.

The GC and OC trends exhibited by each scenario's best test group for each experiment are summarised in Table 49 (where cells following the trends described above are highlighted in green and cells not following the trends described above are highlighted in red).

The graphs illustrating the trends summarised in Table 49 can be found in Appendix C. The cells highlighted in red in Table 49 correspond to situations where the best test group did not perform well within a particular scenario.

All the detection rules used a constant value for the affinity threshold, r, meaning that the maximum generalisation that can be achieved by each individual detector is the same. The maximum generalisation value for each individual detector can in fact be calculated by using equations (5.1) to (5.3) in the previous chapter; that is, by mathematically counting the maximum number of strings that a detector can match. The algorithm introduced by the thesis to calculate GC across a set of detectors (see Figure 12) measures the average number of antigens that are detected by each detector within a detector set. Consequently, a theoretical maximum GC value across a set of detectors is equal to the maximum number of strings that any given detector can theoretically match (calculated by using equations (5.1) to (5.3)) divided by the total number of detectors within the detector set. If the GC value across a set of detectors calculated by the generalisation algorithm (see Figure 12) is equal to the theoretical maximum GC value across a set of detectors (calculated mathematically), then it indicates that the NSA has achieved a good generalisation of the particular problem domain. Some of the test results report GC rates that increase or decrease as the number of detectors increases; this is a direct consequence of the spread of detectors (the location of the detectors within the shape space), that is, GC is actually measuring the average number of non-self strings detected by each detector. Consequently, a low GC value indicates that detectors are not located within critical non-self regions, assuming that the non-self data set comprises a good non-self representation, whereas a high GC value, which approaches the theoretical maximum GC (calculated mathematically), indicates that detectors are located within critical non-self regions. A critical non-self region is a location within the shape space comprising a significant number of non-self cells. The average GC and OC values of the best test group for each scenario across each experiment is contrasted to the theoretical maximum GC and OC values, calculated individually for each best test group, as tabulated in Table 50. Take note that the average GC and OC values reported in Table 50 will never be equal to the theoretical maximum GC and OC values because the size of the non-self data set used for each experiment is far too small. The theoretical maximum GC and OC values should consequently only be used as a guideline and not as a benchmark.

Table 49. *GC and OC Trends*

	Car Data Set		Iris Data Set		Cancer Data Set		Glass Data Set		Mushroom Data Set	
Rule	GC Trend	OC Trend	GC Trend	OC Trend	GC Trend	OC Trend	GC Trend	OC Trend	GC Trend	OC Trend
Feature-detection Rule	GC fluctuates around the same value.	Increases until it is equal to GC.	GC fluctuates around the same value.	Increases until it is equal to GC.	GC fluctuates around the same value.	Is equal to GC and follows the same trend as GC i.e. fluctuates around the same value.	Is equal to 0 and increases sharply at a population size of 5000 to 0.000012.	Is always equal to 0.	GC fluctuates around the same value.	Increases until it is equal to GC.
HD Rule	GC fluctuates around the same value.	Increases until it is equal to GC.	GC fluctuates around the same value.	Increases until it is equal to GC.	GC fluctuates around the same value.	Increases until it is equal to GC.	Is equal to 0 and increases sharply at a population size of 5000 to 0.000012.	Is always equal to 0.	Initially increases, decreases and then slowly increases to a single value.	Increases until it is equal to GC.
RCHK (No MHC)	GC fluctuates around the same value.	Increases until it is equal to GC.	GC fluctuates around the same value.	Increases until it is equal to GC.	GC fluctuates around the same value.	Increases until it is equal to GC.	Is always equal to 0.	Is always equal to 0.	Is equal to 0 and increases sharply at a population size of 5.	Is always equal to 0.
RCHK (Global MHC)	Initially increases and then decreases.	Increases and decreases at the same rate as GC and is equal to GC.	Decreases and then increases slightly.	Is always equal to 0.	Is always equal to 0.	Is always equal to 0.	Is always equal to 0.	Is always equal to 0.	Is always equal to 0.	Is always equal to 0.
RCHK (MHC)	GC fluctuates around the same value.	Increases until it is equal to GC.	Is initially 0, increases sharply and at a population size of 500 and then decreases sharply at a population size of 5000.	Is always equal to 0.	GC fluctuates around the same value.	Increases until it is equal to GC.	Is equal to 0 and increases sharply at 5000 to 0.000002.	Is always equal to 0.	GC fluctuates around the same value.	Increases until it is equal to GC.

Table 50. *Maximum Theoretical GC and OC vs. Actual GC and OC*

	Car Data Set		Iris Data Set		Cancer Data Set		Glass Data Set		Mushroom Data Set	
	n	13	n	21	n	37	n	46	n	58
Feature-Detection Rule	r	5	r	2	r	2	r	23	r	3
	n'	6	n'	3	n'	3	n'	23	n'	4
	Average Reported - GC	18.801	Average Reported - GC	29.01	Average Reported - GC	35.28	Average Reported - GC	1.99	Average Reported - GC	96.97
	Average Reported - OC	18.41	Average Reported - OC	28.98	Average Reported - OC	35.27	Average Reported - OC	0	Average Reported - OC	96.84
	Theoretical Maximum - GC	384	Theoretical Maximum - GC	786432	Theoretical Maximum - GC	51539607552	Theoretical Maximum - GC	8388608	Theoretical Maximum - GC	5.40432E+16
	Theoretical Maximum - OC	384	Theoretical Maximum - OC	786432	Theoretical Maximum - OC	51539607552	Theoretical Maximum - OC	8388608	Theoretical Maximum - OC	5.40432E+16
HD Rule	r	9	r	12	r	18	r	40	r	29
	Average Reported - GC	51.38	Average Reported - GC	16.2	Average Reported - GC	92.87	Average Reported - GC	1.19	Average Reported - GC	149.35
	Average Reported - OC	50.59	Average Reported - OC	15.93	Average Reported - OC	90.15	Average Reported - OC	0	Average Reported - OC	142.17
	Theoretical Maximum - GC	1093	Theoretical Maximum - GC	695860	Theoretical Maximum - GC	86392108636	Theoretical Maximum - GC	10917020	Theoretical Maximum - GC	1.59149E+17
	Theoretical Maximum - OC	1093	Theoretical Maximum - OC	695860	Theoretical Maximum - OC	86392108636	Theoretical Maximum - OC	10917020	Theoretical Maximum - OC	1.59149E+17
RCHK (No MHC)	r	3	r	4	r	5	r	30	r	4
	Average Reported - GC	143.7	Average Reported - GC	8.3	Average Reported - GC	9.62	Average Reported - GC	0	Average Reported - GC	33.48
	Average Reported - OC	143.24	Average Reported - OC	8.25	Average Reported - OC	9.55	Average Reported - OC	0	Average Reported - OC	33.21
	Theoretical Maximum - GC	1024	Theoretical Maximum - GC	131072	Theoretical Maximum - GC	4294967296	Theoretical Maximum - GC	65536	Theoretical Maximum - GC	1.80144E+16
	Theoretical Maximum - OC	1024	Theoretical Maximum - OC	131072	Theoretical Maximum - OC	4294967296	Theoretical Maximum - OC	65536	Theoretical Maximum - OC	1.80144E+16
RCHK (Global MHC)	r	3	r	21	r	37	r	30	r	29
	Average Reported - GC	191.49	Average Reported - GC	8.00E-05	Average Reported - GC	0	Average Reported - GC	0	Average Reported - GC	6.67E-05
	Average Reported - OC	190.96	Average Reported - OC	0	Average Reported - OC	0	Average Reported - OC	0	Average Reported - OC	0
	Theoretical Maximum - GC	1024	Theoretical Maximum - GC	1	Theoretical Maximum - GC	1	Theoretical Maximum - GC	65536	Theoretical Maximum - GC	536870912
	Theoretical Maximum - OC	1024	Theoretical Maximum - OC	1	Theoretical Maximum - OC	1	Theoretical Maximum - OC	65536	Theoretical Maximum - OC	536870912
RCHK (MHC)	r	5	r	21	r	3	r	30	r	4
	Average Reported - GC	21.8	Average Reported - GC	4.00E-05	Average Reported - GC	22.95	Average Reported - GC		Average Reported - GC	100.59
	Average Reported - OC	21.37	Average Reported - OC	0	Average Reported - OC	22.73	Average Reported - OC		Average Reported - OC	100.44
	Theoretical Maximum - GC	256	Theoretical Maximum - GC	1	Theoretical Maximum - GC	17179869184	Theoretical Maximum - GC	65536	Theoretical Maximum - GC	1.80144E+16
	Theoretical Maximum - OC	256	Theoretical Maximum - OC	1	Theoretical Maximum - OC	17179869184	Theoretical Maximum - OC	65536	Theoretical Maximum - OC	1.80144E+16

6.2.13 The Feature-Detection Rule vs. The RCHK Rule with when *r* = *n'*

Chapter 5 argued that the feature-detection is equivalent to the RCHK (MHC) rule if $r = n'$. To empirically validate this argument, the following graphs were created for the test groups where $r = n'$ for both the feature-detection rule and RCHK (MHC) rule for each experiment (the graphs can be found in Appendix C):

- A plot of the OC value as the number of detectors increased.
- A plot of the GC value as the number of detectors increased.
- A plot of the DR value as the number of detectors increased.
- A plot of the FR value as the number of detectors increased.
- A plot of actual population size vs. the target population size, n_c.

The graphs show that regardless of the data set, when $r = n'$ the feature-detection rule had a slightly lower OC, GC value than the RCHK (MHC) rule and equivalent DR and FR value. Interestingly the feature-detection rule was always able to achieve the target population size, n_c whereas the RCHK (MHC) rule was not always able to achieve the target population size.

6.3 Conclusion

The purpose of this chapter was to empirically verify and validate the theoretical analysis of the feature-detection rule performed in chapter 5 by demonstrating that the feature-detection rule performs better than both the HD and the RCHK (No MHC) rules at best case and exhibits performance equivalent to the RCHK (MHC) rule at worst case.

The tests performed in this chapter showed that

- the feature-detection rule was superior (with regards to its DR and FR values) to the RCHK (MHC), RCHK (No MHC), RCHK (Global MHC) and HD for 3 experiments (Car Evaluation, Wisconsin Breast Cancer, and Iris experiments), equivalent to the RCHK (MHC) rule in 1 experiment (Mushroom experiment) and the worst performing rule in 1 data experiment (Glass experiment) because of a very low FR value (0.0047 ± 0.015); and

- the HD rule was superior (with regards to its GC and OC values) to the RCHK (MHC), RCHK, (No MHC), RCHK (Global MHC) and HD rule for 4 experiments (Car Evaluation, Iris, Wisconsin Breast Cancer, and Mushroom experiments). The feature-detection rule was the worst performing rule for 3 of the experiments (Car Evaluation, Mushroom, and Glass experiments) because it has the lowest GC minus OC value.

The application of a single global permutation mask to a set of pre-generated detectors (under the NSA) consistently produced poor results across all of the data sets, thus reaffirming the assertion made in this thesis that permutation masks need to be included in the learning process.

This chapter also studied the effect of different numbers of detectors on the overall GC and OC rates of the feature-detection rule, the RCHK (MHC) rule, the HD and the RCHK (no MHC) rule. It was shown that the OC rate was directly proportional to the number of detectors used across all rules. Interestingly, the chapter showed that the GC count is a quick way to discern whether detectors are located within critical non-self regions, with low GC values indicating that detectors are not located within critical non-self regions.

Chapter 7
Conclusion

"Every day you may make progress. Every step may be fruitful. Yet there will stretch out before you an ever-lengthening, ever-ascending, ever-improving path. You know you will never get to the end of the journey. But this, so far from discouraging, only adds to the joy and glory of the climb."

- Sir Winston Churchill

Although still fairly young, AIS research has proved to be an extremely promising field with regards to a diverse number of application domains. The purpose of this thesis was to examine an individual facet of AIS research, comprising negative selection theory, in order to:

- understand and elaborate on the current thinking within negative selection theory;
- explore how negative selection algorithms are traditionally applied to both binary and real-valued spaces; and
- understand the current limitations imposed by Forrest's original negative selection algorithm operating in binary problem domains.

One such limitation of the traditional matching rules used within NSA, namely, the RCHK rule and the RCBITS rule, is that they are renowned for their simplicity and for inducing undetectable strings/holes. Researchers typically approached this problem by using one of two methodologies: by using another detection rule, such as the HD rule, or by using the concept of a permutation mask to permute the attributes of an antigen before presenting the antigen to a candidate detector. A problem with the former approach is that the HD rule does not necessarily always give the best performance; this was consistently validated against all data sets in the previous chapter. Similarly, a problem with the latter approach, that is, to use permutation masks, was the way in which the permutation masks were implemented, resulting in researchers misconstruing their value and associated benefits. Permutation (MHC) masks are typically applied by AIS researchers in the following manner: a population of detectors is trained under the NSA, and, after training was concluded, a random permutation mask is generated and used by the entire detector set. Since a random permutation mask in fact changes the shape and, thus, the characteristics of a data set, it is not surprising that Stibor *et al.* stated that permutation masks actually shattered the self space, whereas Hofmeyer and Forrest both argued that permutation masks were a mathematically sound way to eradicate holes.

Both researchers are indeed correct, in that permutation masks definitely do eradicate holes induced by both the RCBITS and the RCHK rule. The manner in which they are applied, however, does shatter the self-space, because choosing a single permutation mask for an entire set of detectors is equivalent to taking a random guess.

The viewpoint presented in this thesis is to consider a binary problem domain as a set of characteristics/features that need to be learned by the NSA in order to differentiate successfully between self and non-self. For example, consider a detector, $\mathbf{x}$, under the RCBITS rule, where $\mathbf{x} = (1,0,0,1,0)$ and $r = 3$. Now the detector $\mathbf{x}$ can actually be interpreted as capturing three relationships between the following features of a data set: (1, 2, 3); (2, 3, 4) and (3, 4, 5), respectively. If detection rules were to be examined in this manner, then it is evident that both the RCBITS rule and the RCHK rule will not easily be able to learn the relationship between non-adjacent features. By employing this viewpoint, the thesis introduced a new detection rule, called the feature-detection rule, which selects a subset of both adjacent and or non-adjacent features of an antigen in order to learn the most meaningful relationships between characteristics within a set of antigen.

The thesis also took the viewpoint that multiple relationships, that is, meaningful permutations between features, exist and that these need to be learnt in order to differentiate successfully between self and non-self data. In view of the number of possible permutations of features that could exist in a binary data set of length n, it is very difficult to infer which relationships between features are meaningful and will assist in solving the problem. A much more feasible way of approaching this problem is to include the learning in the NSA; that is, to generate random detectors abiding to the feature-detection rule, where the subset of features selected by a random detector are determined randomly, and to apply the NSA to the random detectors. Consequently, the same approach needs to be applied to the generation of permutation masks for the RCBITS rule or the RCHK rule. That is, a random permutation mask must be generated for each detector abiding to either the RCBITS or RCHK rule, and the NSA is then applied to the detectors. Following this line of thought, the thesis showed that, conceptually, the feature-detection rule is equivalent to the RCHK rule (MHC) if $r = n'$, and if $r < n'$, the feature-detection rule is equivalent to multiple RCHK detectors. The empirical results performed in the previous chapter confirmed that the feature–detection rule performs better than the RCHK (No MHC), RCHK (Global MHC), RCHK (MHC) and HD rule. One negative aspect of the feature-detection rule is that is introduces positional bias because it leverages the RCBITS rule, which is overcome by coupling the generation of the feature vector **p**, together with each detector **x**, generated by the NSA.

The empirical results performed in the previous chapter also showed that if $r = n'$ then the feature-detection rule has a similar DR and FR to the RCHK (MHC) rule in addition to having lower GC and OC values.

The following future work can be undertaken to expand the work presented within this thesis:

- The NSA was shown to suffer from a severe scaling problem that is, when P_M, $P_{f\,,}$ and n_c are fixed, an exponential increase in n_r can be observed. The feature-detection rule is not immune to this problem, and extensions to algorithms such as the linear time detector-generation algorithm and the greedy detector-generating algorithm can be undertaken to address this issue.
- The NSA distributes detectors in a random manner. This was verified by two phenomena occurring within empirical results, in that 1) OC was shown to be directly proportional to the number of detectors used in a particular simulation and 2) the best performing scenarios exhibited both a high GC and OC rate. A mechanism similar to the one employed by the real-valued NSA could be used to optimise the distribution of the detectors so that OC is minimised.

- The applicability of the feature-detection rule can be extended to other AIS paradigms, such as clonal selection theory or network algorithms. Take note however that the effect of positional-bias introduced by the feature-detection rule needs to considered when applying the feature-detection rule to these algorithms.

- The applicability of the feature-detection rule can be extended and investigated within real-valued spaces.

- The feature-detection rule can be extended further such that rules other than the RCBITS rule are used to ascertain whether a detector is activated by an antigen, in an attempt to explicitly remove the positional bias introduced by the feature-detection rule. For instance a series of binary rules can be applied to the position vector **p.**

- The technique used by González *et al.* to visualise an immune system [14] can be used to graphically visualise the shape space induced by the feature-detection rule.

- The technique used by González *et al.* [14] could be contrasted to Sammon's mapping to determine how effective both methods are in attempting to visualise the shape space.

- Immune libraries are exceptionally good at generating diversity within the immune system and should be combined with the NSA and compared to a traditional NSA with randomly generated detectors.

Bibliography

[1] C. A Janeway, P. Travers, M. Walport and M. Shlomchik. Immunobiology: The Immune System in Health and Disease. Garland Science Publishing, 2001.

[2] D. Aha, P. Murphy, C. Merz, E. Keogh, C. Blake and S.Hettich. University of California Irvine Repository of Machine Learning Databases, http://www.ics.uci.edu/~MLRepository.html, 2008.

[3] U. Aickelin and S.Cayzer. The Danger Theory and its Application to Artificial Immune Systems. *In Proceeding of the First International Conference on Artificial Immune Systems*, pages 141-148, 2002.

[4] M. Ayara, J. Timmis, R. De Lemos, L.N. de Castro and R. Duncan. Negative Selection: How to Generate Detectors, *In Proceedings of International Conferences on Artificial Immune Systems*, pages 89-98, 2002.

[5] J. Bastrop, F. Esponda, S. Forrest and M. Glickman. Coverage and Generalisation in an Artificial Immune System. *In Proceedings of the Genetic and Evolutionary Computation Conference*, pages 3-10, 2002.

[6] C. Berek and M. Ziegner. The Maturation of the Immune Response. *In Immunology Today*. 14(8), pages 400-402, 1993.

[7] D.W. Bradley and A.M. Tyrrel. The Architecture for a Hardware Immune System. *In Proceedings of the Third NASA/DoD Workshop on Evolvable Hardware*, pages 193-200, 2001.

[8] P. Bugl. “The Immune System”. Department of Mathematics, University of Hardford. http://uhaweb.hartford.edu/BUGL/immune.htm. March 2001.

[9] F.M. Burnet. The Clonal Selection Theory of Acquired Immunity. *Vanderbilt University Press*, Nashville, T.N., 1959.

[10] M. Cohn. An Alternative to Current Thinking about Positive Selection, Negative Selection and Activation of T-cells. *In Immunology*, 111, pages 375-380, 2004.

[11] P. D'haeseleer, S. Forrest and P. Helman. An Immunological Approach to Change Detection: Algorithms, Analysis and Implications. *In Proceedings of the 9th IEEE Symposium on Computer Security and Privacy*, pages 110-120, 1996.

[12] P. D'haeseleer. An Immunological Approach to Change Detection: Theoretical Results. *In Proceedings of IEEE Symposium on Security and Privacy*, pages 18-27, 1996.

[13] D. Dasgupta and S. Forrest. Novelty Detection in Time Series Data using Ideas from Immunology. *In Proceedings of the 5th International Conference on Intelligent Systems*, 1996.

[14] D.Dasgupta, S. Yu, and N.S. Majumdar. Multilevel Immune Learning Algorithm. *In Soft Computing a Fusion of Foundations, Methodologies and Applications*, 9(3), pages 172-184, 2005.

[15] D. Dasgupta. Advances in Artificial Immune Systems. *In IEEE Computational Intelligence Magazine*, 1(4), pages 40-49, 2006.

[16] D. Dasgupta. Artificial Neural Networks and Artificial Immune Systems: Similarities and Differences. *In IEEE Conferences on Systems, Man and Cybernetics, volume 1*, pages 873-878, 1997.

[17] L.N. de Castro and F.J. von Zuben. Artificial Immune Systems: Part I – Basic Theory and Applications. *In Technical Report – RT DCA 01/99*, pages 95-190, 1999.

[18] L.N. de Castro and F.J. von Zuben. Convergence and Hierarchy of aiNet: Basic Ideas and Preliminary Results. *In Proceedings of 2002 International Conference on Artificial Immune Systems*, pages 231-240, 2002.

[19] L.N. de Castro and F.J. von Zuben. Immune and Neural Network Models: Theoretical and Empirical Comparisons. *In International Journal of Computational Intelligence and Applications*, volume 1, pages 239-257, 2001.

[20] L.N. de Castro and F.J. von Zuben. Learning and Optimisation Using the Clonal Selection Principle. *In Proceedings of IEEE Transactions on Evolutionary Computation*, 6(3), pages 239-251, 2002.

[21] L.N. de Castro and F.J. von Zuben. The Clonal Selection Algorithm with Engineering Applications. *In Proceedings of Genetic and Evolutionary Computation*, pages 36-37, 2000.

[22] L.N. de Castro and J.I. Timmis. Artificial Immune Systems as a Novel Soft Computing Paradigm. Springer, 2003.

[23] L.N. de Castro and J.I. Timmis. Artificial Immune Systems: A New Computational Intelligence Approach. Springer. 2002.

[24] B.N. Domingo and N. Barquet. Smallpox: The Triumph over the Most Terrible of the Ministers of Death. *In Ann. Intern. Med*, 127 (8), pages 635-42, 1997.

[25] M. Ebner, H.G. Breunig and J. Albert. On the use of Negative Selection in an Artificial Immune System. *In Proceedings of the Genetic and Evolutionary Computation Conference*, pages 957-964, 2002.

[26] A.P. Engelbrecht. Computational Intelligence: An Introduction: Second Edition. John Wiley & Sons, 2007.

[27] A.P. Engelbrecht. Fundamentals of Computational Swarm Intelligence. John Wiley & Sons Ltd. 2005.

[28] F. Esponda, E.S. Ackley, S. Forrest and P. Helman. Online Negative Databases. *In Proceedings of International Conference on Artificial Immune Systems 2004*, pages 175-188, 2004.

[29] F. Esponda, S. Forrest and P. Helman. A Formal Framework for Positive and Negative Detection Schemes. *In Proceedings of IEEE Transactions on Systems, Man and Cybernetics*, 34(1), pages 357-373, 2004.

[30] J.D. Farmer, N.H. Packard and A.S. Perelson. The Immune System, Adaptation and Machine Learning. *In Physica D*, 2, pages 187-204, 1986.

[31] S. Forrest and A. S. Perelson. Computation and the immune system. *In SIGBIO Newsletter, Association for Computing Machinery*, Vol. 12, Num. 2 (June, 1992)

[32] S. Forrest and S. Hofmeyer. Engineering an Immune System. *In Graft 4(5)*, pages 5-9, 2001.

[33] S. Forrest, A.S. Perelson, L.Allen and R. Cherukuri. Self-Nonself Discrimination in a Computer. *In Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy*, pages 202-212, 1994.

[34] S. Forrest, S.A. Hofmeyer and A. Somayaji. Computer Immunology. *In Communications of the ACM*, 40(10), pages 88-96, 1996.

[35] S. Forrest, S.A. Hofmeyer, A. Somayaji and T.A. Longstaff. A Sense of Self for Unix Processes. *In Proceedings of IEEE Symposium on Computer Security and Privacy*, pages 120-128, 1996.

[36] A. Freitas and J. Timmis. Revisiting the foundations of Artificial Immune Systems: A problem oriented perspective. *In Volume 2787 of LNCS*, pages 229-241, 2003

[37] D.E. Goldberg. Genetic Algorithms in Search Optimisation and Machine Learning. Addison Wesley, 1989

[38] F. González, D. Dasgupta and J.Gómez. The Effect of Binary Matching Rules in Negative Selection. *In Proceedings of Genetic and Evolutionary Computation Conference*, 1, pages 195-206, 2003.

[39] F. Gonzalez, D. Dasgupta and L.F. Niño. A Randomized Real-valued Negative Selection Algorithm. *In Proceedings of the 2nd International Conference on Artificial Immune Systems*, pages 261-272, 2003.

[40] F.Gonzalez, D. Dasgupta and R. Kozma. Combining Negative Selection and Classification Techniques for Anomaly Detection. *In Proceedings of IEEE Congress on Evolutionary Computation*, pages 705-710, 2002.

[41] A.J. Graaf. Artificial Immune Systems with Evolved Lymphocytes. MsC thesis, University of Pretoria, South Africa, 2004.

[42] J. Greensmith, U. Aickelin and J. Twycross. Articulation and Clarification of the Dendritic Cell Algorithm. *In Volume 4163 of LNCS*, pages 404-417, 2006.

[43] J. Greensmith, U. Aickelin, and S. Cayzer. Introducing dendritic cells as a novel immune-inspired algorithm for anomaly detection. *In ICARIS-05, LNCS 3627*, pages 153–167, 2005.

[44] E. Hart and P. Ross. An Immune System Approach to Scheduling in Changing Environments. *In Proceedings of Genetics and Evolutionary Computation Conference*, pages 1559-1565, 1999.

[45] R. Hightower, S. Forrest and A.S. Perelson. The Baldwin Effect in the Immune System: Learning by Somatic Hypermutation. *In Adaptive Individuals in Evolving Populations*, Addison-Wesley, pages 159-167, 1996.

[46] R. Hightower, S. Forrest and A.S. Perelson. The Evolution of Emergent Organization in the Immune System Gene Libraries. *In Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 344-350, 1995.

[47] A. Hofmeyer and S. Forrest. Architecture of an Artificial Immune System. *In Evolutionary Computation Journal*, 8(4), pages 443-473, 2000.

[48] S. Hofmeyer and S. Forrest. Immunity by Design: An Artificial Immune System. *In Proceedings of the Genetic and Evolutionary Conference*, pages 1289-1296, 1999.

[49] S. Hofmeyer. An Immunological Model of Distributed Detection and its Application to Computer Security. PhD thesis, University of New Mexico, 1999.

[50] J.H. Holland, K.J. Holyoak, R.E. Nisbett and T.Thagard. Induction: Processes of Inference, Learning and Discovery. The MIT Press. 1986.

[51] N.K. Jerne, J. Cocteau. Idiotypic Networks and Other Preconceived Ideas. *In Immunological Reviews 79*, 1, pages 5-24, 1984.

[52] N.K. Jerne. Towards a Network Theory of the Immune System. *In Annals of Immunology (Inst. Pasteur)*, 125C, pages 373-389, 1974.

[53] Z. Ji and D. Dasgupta. Real Valued Negative Selection Algorithms with Variable Size Detectors. *In Genetic and Evolutionary Computation Part 1*, 3102, pages 287-298, 2004.

[54] Z. Ji and D. Dasgupta. Revisiting Negative Selection Algorithms. *In Evolutionary Computation Archive*, 15(2): pages 223-251, 2007.

[55] J. Kim and P. Bentley. An Evaluation of Negative Selection in an Artificial Immune System for Network Intrusion Detection. *In Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1330-1337, 2001.

[56] J.Kim and P.J. Bentley. Immune Memory in the Dynamic Clonal Selection Algorithm. *In Proceedings of the First International Conference on Artificial Immune Systems*, pages 57-65, 2002.

[57] J. Kim and P.J. Bentley. Towards an Artificial Immune System for Network Intrusion Detection: An Investigation of Dynamic Clonal Selection. *In Proceedings of Congress on Evolutionary Computation,* pages 1015-1020, 2002.

[58] J. Kim, A. Ong and R. Overill. Design of an Artificial Immune System as a Novel Anomaly Detector for Combating Financial Fraud in the Retail Sector. *In Congress on Evolutionary Computation (CEC-2003)*, pages 405-412, 2003.

[59] F.Liu, L. Bai and L. Jiao. Intrusion Detection based on Adaptive Resonance Theory and Artificial Network Clustering. *In Lecture Notes in Computer Science*, 3611, pages 780-783, 2005.

[60] P. Matzinger. The Danger Model in its Historical Context. *In Scandinavian Journal of Immunology*, 54, pages 4-9, 2001.

[61] P. Matzinger. The Real Function of the Immune System. http://cmmg.biosci.wayne.edu/asg/polly.html, 2004.

[62] Merck Incorporated. Immunology; Allergic Disorders. The Merck Manual of Diagnosis and Therapy, http://www.merck.com/mmpe/sec13.html, 2000.

[63] A. Morrison, C.Ross, M. Chalmers. Fast multidimensional scaling, springs and interpolation. *Information Visualization Volume 2*, pages 68–77, 2003.

[64] T. Oda. A Spam Detecting Artificial Immune System. . MsC thesis, Carleton University Canada, 2005.

[65] M. Opera and S. Forrest. Simulated Evolution of Antibody Gene Libraries under Pathogen Selection. *In Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, 4, pages 3793-3798, 1998

[66] M. Opera M and S. Forrest. How the Immune System generates Diversity: Pathogen Space Coverage with Random Evolved Antibody Libraries. *In Proceedings of Genetic and Evolutionary Computation Conference*, 2, pages 1651-1656, 1999.

[67] J.K. Percus, O. Percus and A.S. Perelson. Probability of Self-nonself Discrimination. *In Theoretical and Experimental Insights into Immunology*, Springer, 1996.

[68] J.K. Percus, O.E. Percus and A.S. Perelson. Predicting the Size of the Antibody Combining Region from Consideration of Efficient Self-NonSelf Discrimination. *In Journal of Theoretical Biology*, 91, pages 645-670, 1993.

[69] A.S. Perelson and G.F. Oster. Theoretical Studies of Clonal Selection: Minimal Antibody Repertoire Size and Reliability of Self-NonSelf Discrimination. *In Journal of Theoretical Biology*, 81, pages 645-670, 1979.

[70] A. Perelson, R. Hightower and S. Forrest. Evolution and Somatic Learning in v-region Genes. *In Research in Immunology*, 147, pages 202-208, 1996.

[71] A.S. Perelson. Immune Network Theory. *In Immunological Review*, 110, pages 5-36, 1989.

[72] A. Pietzwoski, B. Satzger, W. Trumler and T. Ungerer. Using Positive and Negative Selection from Immunology for Detection of Anomalies in a Self Protecting Middleware. *In INFORMATIK 2006 -- Informatik für Menschen*, P-93, pages 161-168, 2006.

[73] M.A. Potter and K. de Jong. The Coevolution of Antibodies for Concept Learning. *In Proceedings of the Fifth International Conference on Parallel Problem Solving from Nature*, 1498, pages 530-539, 1998.

[74] J.O. Ramsay. Maximum likelihood estimation in multidimensional scaling. *In Psychometrica 42*, pages 241-66. 1977

[75] J.W. Sammon. A Non Linear Mapping for Data Structure Analysis. *In Proceedings of IEEE Transactions on Computers*, C-18, pages 401-409, 1969.

[76] A. Secker, A. Freitas and J.Timmis. AISEC an artificial immune system for email classification. *In Proceedings of the Congress on Evolutionary Computation*, pages 131-139, 2003.

[77] R. Singh and R.N. Sengupta. Bankruptcy Prediction in Artificial Immune Systems. *In Lecture Notes in Computer Science*, 4628, pages 1611-3349, 2007.

[78] D.J Smith, S. Forrest and S. Perelson. Immunological Memory is Associative. *In Workshop Notes, Workshop 4: Immunity Based Systems of International Conference on Multiagent Systems*, pages 62-70, 1996.

[79] D.J. Smith. S. Forrest. A.S. Perelson and D.H. Ackley. Modelling the Effects of Prior Infection on Vaccine Efficacy. *In Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, pages 363-368, 1997.

[80] T.Stibor, J. Timmis and C.Eckert. On Permutation Masks in Hamming Negative Selection. *In Proceedings of the Sixth International Conference of Artificial Immune Systems (ICARIS 2006)*, pages 122-135, 2006.

[81] T. Stibor, J.Timmis and C. Eckert. A Comparative Study of Real Valued Negative Selection to Statistical Anomaly Techniques. *In International Conference of Artificial Immune Systems 2005*, pages 262-275, 2005.

[82] T.Stibor, P. Mohr and J. Timmis. Is Negative Selection Appropriate for Anomaly Detection. *In Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, pages 321-328, 2005.

[83] J.B. Tenenbaum, V. de Sliva, J.C. Langford.A global geometric framework for nonlinear dimensionality reduction. *In science 2000 volume 290*, pages 2319–2323, 2000.

[84] J. Timmis. Artificial Immune Systems: A Novel Data Analysis Technique Inspired by Immune Network Theory. PhD thesis, University of Wales, August 2000.

[85] J.H. Walker and S.M. Garrett. Dynamic Function Optimisation: Comparing the Performance of Clonal Selection and Evolution Strategies. *In Lecture Notes in Computer Science*, 2787, pages 273-284, 2003.

[86] J.A. White and S.N. Garett. Improved Pattern Recognition with Artificial Clonal Selection. *In Proceedings of the Second International Conference of Artificial Immune Systems (ICARIS 2003)*, pages 181-193, 2003.

[87] S.T. Wierzchon. Discriminative Power of the Receptors activated by the k-contiguous Bits Rule. *In Journal of Computer Science and Technology*, 1(3), pages 1 -13, 2000.

[88] S.T. Wierzchon. Generating Optimal Repertoire of Antibody Strings in an Artificial Immune System. *In Proceedings of International Symposium on Intelligent Information Systems*, pages119-133, 2000.

[89] X. Yue, A. Abraham, Z.Chi, Y. Hao and H. Mo. Artificial Immune System Inspired Behavior Based Anti Spam Filter. *In Soft Computing*, 11(8), pages 729-740, 2007.

[90] Xun Yue; Zhongxian Chi; Yanyou Hao and Hongwei Mo. Incremental Clustering Algorithm of Data Stream Based on Artificial Immune Network. *In Proceedings of the Sixth World Congress on Intelligent Control and Automation*, 1, pages 4021–4025, 2006.

Appendix A - Acronyms

AIN	Artificial immune network
AINET	Artificial immune network for data analysis
AIS	Artificial immune system
ALC	Artificial lymphocyte
APC	Antigen-presenting cell
CSA	Clonal selection algorithm
DCA	Dendritic Cell Algorithm
DR	Detection rate
FN	False negative
FP	False positive
FR	False-alarm rate
GC	Generalisation count
HD	Hamming distance
HS	Hamming separation
Ig	Immunoglobulin
MHC	Major histocompatibility complex
MILA	Multilevel immune learning algorithm
NIS	Natural immune system
NSMutate	Negative selection with guided mutation
NSA	Negative selection algorithm
OC	Overfitting count
RCBITS	R-contiguous bits
RCHK	R-chunks
TH	T-helper lymphocytes
TN	True negative
TP	True positive
TS	T-suppressor lymphocyte

Appendix B - Glossary

A' A particular AIS algorithm.

$b.s$ A binary string with b appended to the beginning of the string, where $b \in \{0,1\}$.

C A set of artificial lymphocytes generated by a particular AIS algorithm.

c The centre of a real-valued detector in hyper space.

c_0 The estimated coverage of the problem space by a set of detectors C.

$\mathbf{D}$ A matrix that is structurally similar to $\mathbf{M}$, where each entry $\mathbf{D}_i[s]$ represents the number of unmatched fully specified bit strings corresponding to the template represented by i. Each entry $\mathbf{D}_i[s] = \mathbf{M}_i[s] * \mathbf{M}'_i[s]$, where $s \in S$.

$\mathbf{D}'$ A matrix that is structurally similar to $\mathbf{M}$ where each entry $\mathbf{D}'_i[s]$ represents the number of unmatched fully specified bit strings corresponding to the template represented by i. Each entry $\mathbf{D}'_i[s] = \mathbf{M}_i[s] * \mathbf{M}'_i[s]$**,** where $s \in C$.

D_{TRAIN} A training set of self/non-self vectors depending on the AIS algorithm being employed.

D_{TEST} A test set of self/non-self vectors depending on the AIS algorithm being employed.

$f_{EUCLIDEAN}(\mathbf{w}, \mathbf{m})$ Returns the Euclidean distance between vectors $\mathbf{w}$ and $\mathbf{m}$**.**

$f_{HD}(\mathbf{w}_1, \mathbf{w}_2)$ Returns the hamming distance between two binary vectors $\mathbf{w}_1$ and $\mathbf{w}_2$.

$f_{MATCH}(\mathbf{w}_1, \mathbf{w}_2)$	A generic matching function that determines the affinity between vectors $\mathbf{w}_1$ and $\mathbf{w}_2$.
$f_{MAX}(\mathbf{x})$	Returns the maximum affinity of a detector.
$f_{PERMUTE}(\mathbf{w}, \mathbf{m})$	Applies a permutation mask **m,** to a vector, **w**.
$f_{RCBITS}(\mathbf{w}_1, \mathbf{w}_2)$	Returns the number of contiguous bits shared by two binary vectors $\mathbf{w}_1$ and $\mathbf{w}_2$.
$f_{RCHK}(\mathbf{w}_1, \mathbf{w}_2, w)$	Returns the number of contiguous bits between vectors $\mathbf{w}_1$ and $\mathbf{w}_2$starting at position, w in $\mathbf{w}_1$.
$f_{FEATURE}(\mathbf{w}_1, \mathbf{p})$	Given a binary vector $\mathbf{w_1} = (w_1, w_2, \ldots, w_n)$ and a vector of integer positions $\mathbf{p} = (p_1, p_2, \ldots, p_{n'})$ where $p_1 \leq n$ and $n' \leq n$ construct a vector $\mathbf{w}_\mathbf{1}' = (w_{p1}, w_{p2}, \ldots, w_{pn'})$ by using the positions stipulated in **p**.
G	A graph containing all possible templates, $t_{i,w}$, induced by a set of self-strings, S.
g	The average generalisation of a particular AIS algorithm.
$\mathbf{h}$	A crossover hole.
H_{NS}	A set of holes for a given non-self set.
H_S	A set of holes for a given self-set.
$\mathbf{m}$	A permutation mask.
$\mathbf{M}$	A matrix representing the number of non-matching left completions for a template, $t_{i,s}$.
$\mathbf{M}'$	A matrix representing the number of non-matching left completions for a template, $t_{i,s}$.

$\mathbf{M}_i[s]$ The number of right completions of $t_{i,s}$ that is unmatched by any string within the self-set, S.

msc The maximum coverage of the problem space by a set of detectors, C

n The dimensionality of a particular artificial lymphocyte.

n' The dimensionality of an artificial lymphocyte under the feature-detection rule, where $n' \leq n$.

n_{alph} The number of symbols in an alphabet. For example, the binary alphabet contains two symbols.

n_c The number of candidate artificial lymphocytes to train.

n_r The number of detectors needed by the NSA that would need to be created in order to generate n_c detectors.

N_S A set comprising non-self vectors/binary strings.

o The average overfitting of a particular AIS algorithm.

o_i The number of self/non-self patterns that are overfitted by an artificial lymphocyte, $\mathbf{x}_i \in C$.

P_S The probability of a random string not matching any self-string/vector.

P_f The probability that n_c detectors fail to detect an intrusion.

P_M The probability that two random strings match under a specific affinity-matching function.

$P_{M'}$ The probability that two random strings do not match under a specific affinity-matching function.

P_{mutate} The probability that a detector is mutated.

r	The affinity threshold of an artificial lymphocyte.
r_s	Self-radius (used in the V-detector algorithm).
S	A set comprising self-vectors/self-binary strings.
S_r	A set of all possible windows/templates of size r in S.
S_{rself}	A set of all windows/templates matching at least one self string, $s \in S$ with $S_{rself} \subseteq S_r$.
$S_{rnonself}$	A set of all windows or templates that do not match a single string $s \in S$, with $S_{rnonself} \subseteq S_r$.
s	A binary string of length n.
$\hat{s}$	A binary string without its leftmost bit.
$\hat{s}$	A binary string without its rightmost bit.
$s.b$	A binary string with b appended to the end of the string, where $b \in \{0,1\}$.
T	The total number of strings unmatched in a particular set.
$t_{i,w}$	A template where w denotes the binary string of length r and i denotes the starting position of w.
V	Entire shape space comprising all self- and non-self vectors.
V_r	The detection region of an artificial lymphocyte.
w	The starting position of the detection window used by the RCHK rule.
$\mathbf{x}$	An artificial lymphocyte/detector of length n with coordinates $(x_1, x_2, \ldots, x_n)$.
$\mathbf{x}_{ij}$	The j^{th} element of a detector $\mathbf{x}_i = (x_1, x_2, \ldots, x_n)$.
$\mathbf{y}$	An antigen/non-self artifact of length n with coordinates

$(y_1, y_2, \ldots, y_n)$

$\mathbf{y}'$ A subset of features selected from **y** such that $\mathbf{y}' = (y_1, y_2, \ldots, y_n)$ and each $y_i \in \mathbf{y}$ and the length of $\mathbf{y}'$ is equal to Y_i: A set of antigens detected by an artificial lymphocyte $\mathbf{x}_i$, that is, $Y_i = \{y_i \in N_s | f_{MATCH}(\mathbf{x}_i, \mathbf{y}_i) = true\}$.

z A self-cell/artifact of length n with coordinates $(z_1, z_2, \ldots, z_n)$.

Appendix C – Graphs and Trends

Appendix C is comprised of all of the graphs used to illustrate the trends observed within the best test groups for each of the experiments executed in chapter 6. The appendix is grouped into 5 sections, where each section corresponds to a particular experiment performed in chapter 6. For the sake of completeness, the DR and FR trends are also illustrated in addition to the GC and OC trends discussed in chapter 6. The significance of the DR and FR trends is that they graphically illustrate the effect of an increasing average population size on the DR and FR values.

C.1 The Car Evaluation Data Set Trends

This section contains graphs that illustrate the OC, GC, DR, and FR trends observed for each scenario's best test groups for the Car Evaluation experiment. The section is concluded with graphs illustrating the OC, GC, DR, FR, and target population size, N_c vs. actual average population size trends for the feature-detection rule and RCHK (MHC) rule test groups where $r = n'$.

C.1.1 Feature-Detection Rule Trends

The GC and OC trends of the feature-detection rule for the best scenario illustrated in Figure 48 show that:

- OC increases sharply as the average population size increases until it is equal to GC at an average population size of 5000. The OC trend is bad because it illustrates that the amount of overfitting increases as the number of detectors are increased i.e. the NSA is distributing detectors in a manner such that their detection regions overlap.
- GC initially decreases as the average population size increases and then settles on a value of 18. The GC trend is essentially equal to an average of 18 and the fluctuation in the GC value is caused by the randomness that occurs when executing each test within the test group, i.e. each test within a particular test

group is executed 30 times where each test execution entails that a test set and training set are randomly selected from the self-set in addition to randomly generating detectors.

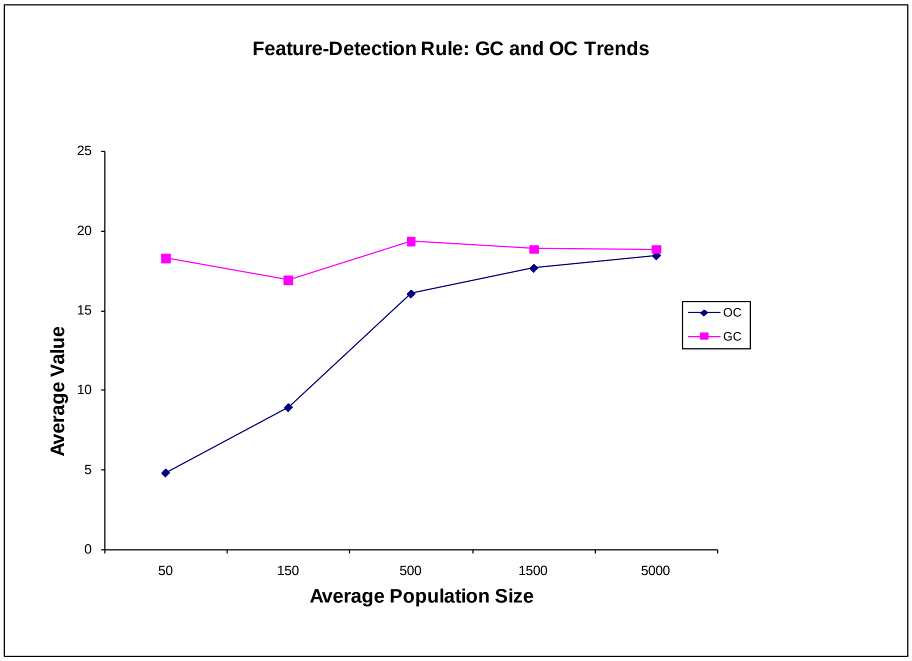


Figure 48. *Car Evaluation Data Set – Feature-Detection Rule GC and OC Trends*

The DR and FR trends of the feature-detection rule for the best scenario illustrated in Figure 49 show that:

- DR increases as the average population size increases until it is equal to 1 at a population size of 5000.
- FR only increases marginally as the average population size increases and is equal to 0.02 at an average population size of 5000.

The DR and FR trends are good because the DR value is equal to 1 whilst the FR is almost equal to 0, meaning that the DR - FR value is maximised.

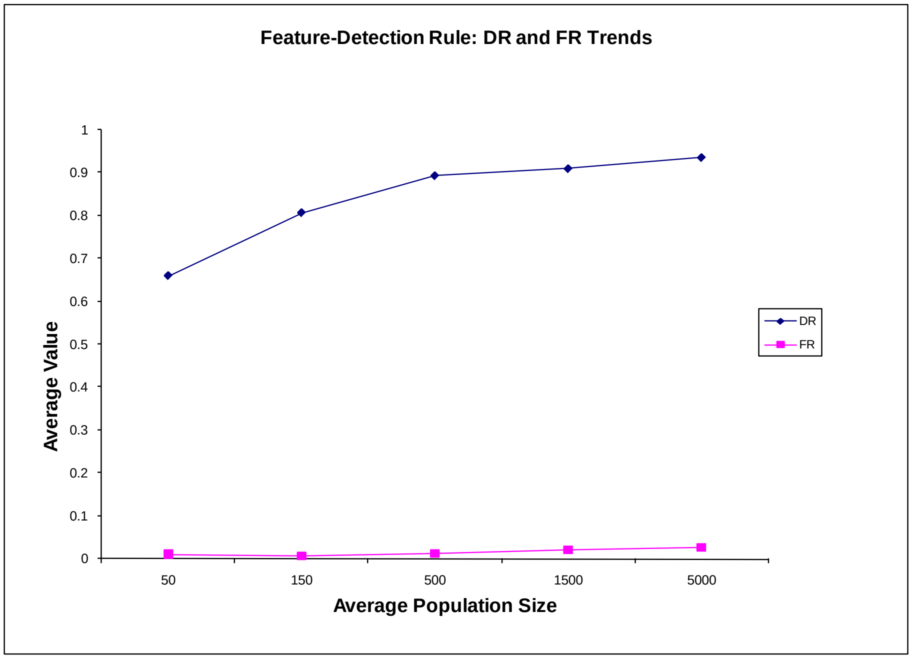


Figure 49. *Car Evaluation Data Set – Feature-Detection Rule DR and FR Trends*

C.1.2 HD Rule Trends

The GC and OC trends of the HD rule for the best scenario illustrated in Figure 50 show that:

- GC fluctuates slightly around a value of 50 as the average population size increases. The GC value is essentially equal to an average of 50 and the fluctuation in the GC value is caused by the randomness that occurs when executing each test within the test group.
- OC increases sharply as the average population size increases and is almost equal to GC at an average population size of 1990.2. The OC trend is bad because it illustrates that the amount of overfitting increases as the number of detectors are increased, i.e. the NSA is distributing detectors in a manner such that their detection regions overlap.

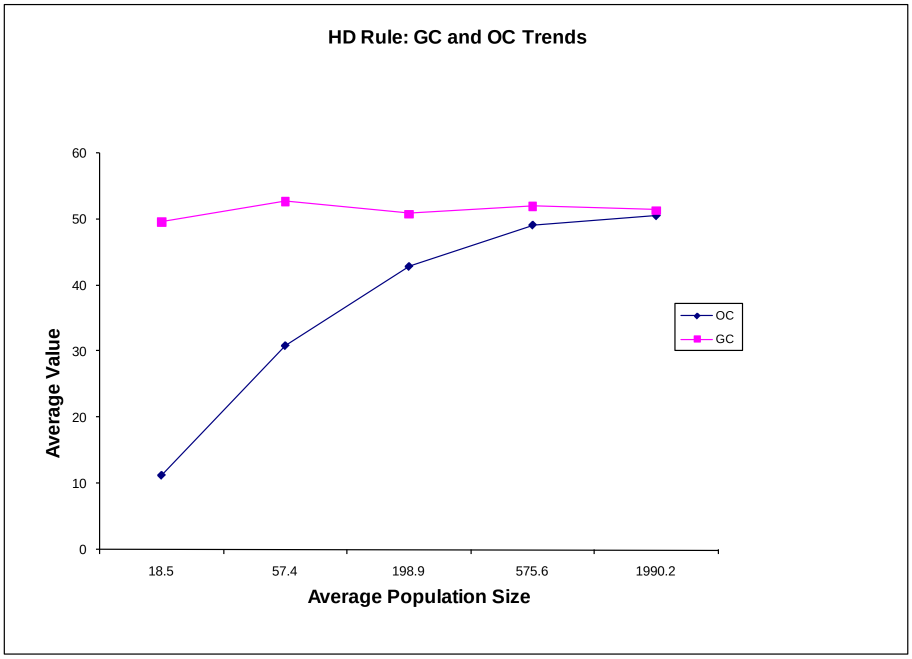


Figure 50. *Car Evaluation Data Set - HD Rule GC and OC Trends*

The DR and FR trends of the HD rule for the best scenario illustrated in Figure 51 show that:

- DR increases slowly from 0.64 to 0.76 as the average population size increases.
- FR is relatively constant (fluctuating by an average of 0.02) at 0.02.

The DR and FR trends are good and illustrate that an average population size of 198.9 is sufficient to obtain the same DR and FR value at an average population size of 1990.2.

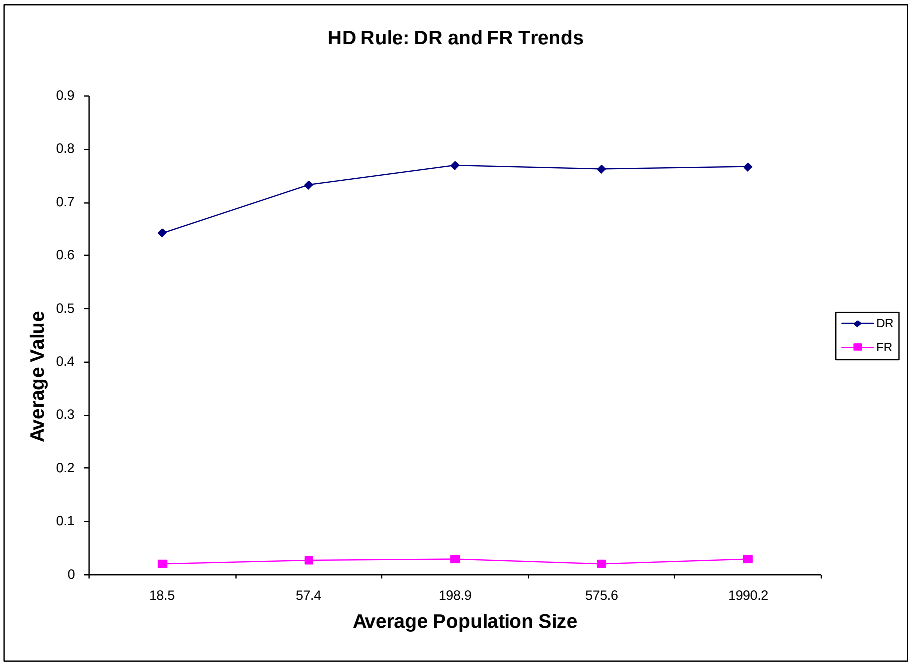


Figure 51. *Car Evaluation Data Set - HD Rule DR and FR Trends*

C.1.3 RCHK (No MHC) Rule Trends

The GC and OC trends of the RCHK (No MHC) rule for the best scenario illustrated in Figure 52 show that:

- GC is initially equal to 153 at an average population size of 37.1 and then settles on 143 at an average population size of 3677. GC is essentially equal to an average of 143 and the fluctuation in the GC value is caused by the randomness that occurs when executing each test within a test group.
- OC is initially equal to 115 at an average population size of 37.1 and gradually increases as the average population size increases until it is equal to GC. The OC trend is bad because it illustrates that the amount of overfitting increases as the number of detectors are increased, i.e. the NSA is distributing detectors in a manner such that their detection regions overlap.

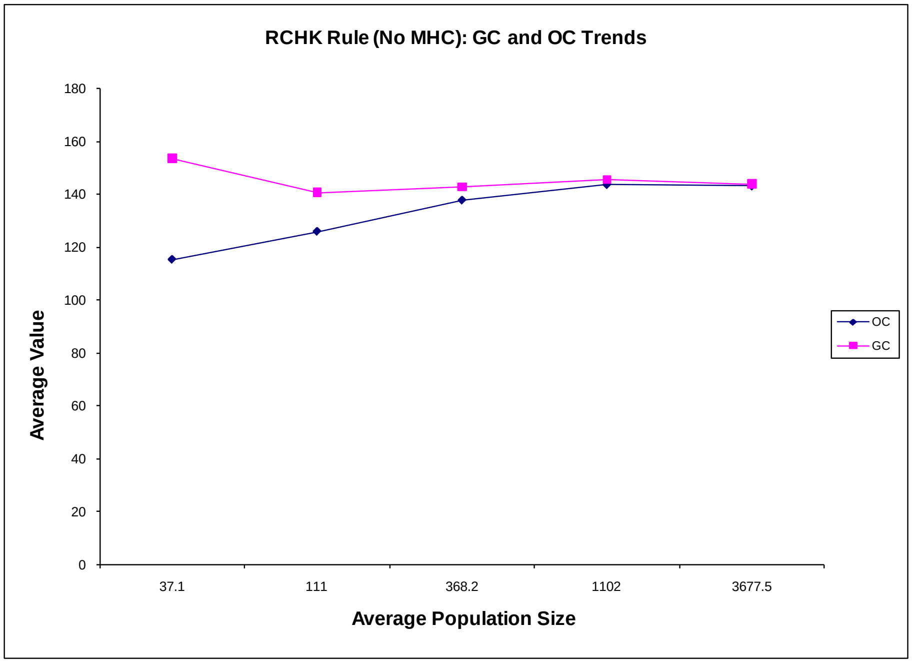


Figure 52. *Car Evaluation Data Set - RCHK (No MHC) Rule GC and OC Trends*

The DR and FR trends of the RCHK (No MHC) rule for the best scenario illustrated in Figure 53 show that:

- DR increases slightly as the average population size increases.
- FR is constant at 0 as the average population size increases.

The DR and FR trends are good and illustrate that an average population size of 368.2 is sufficient to maintain the DR and FR value obtained by an average population size of 3677.5.

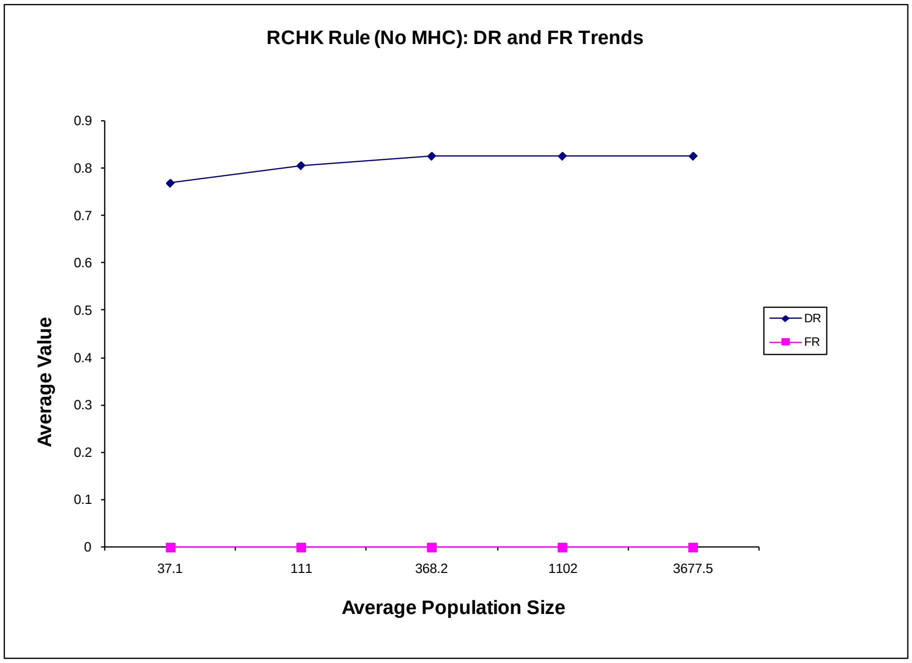


Figure 53. *Car Evaluation Data Set - RCHK (No MHC) Rule DR and FR Trends*

C.1.4 RCHK (Global MHC) Rule Trends

The GC and OC trends of the RCHK (No MHC) rule for the best scenario illustrated in Figure 54 show that both GC and OC increase and then decreases at the same rate as the average population size increases. The GC and OC trends are bad because they illustrate that the average amount of overfitting exhibited by the NSA under the RCHK rule (No MHC) is equivalent to the average amount of generalisation exhibited by the NSA regardless of the average population size.

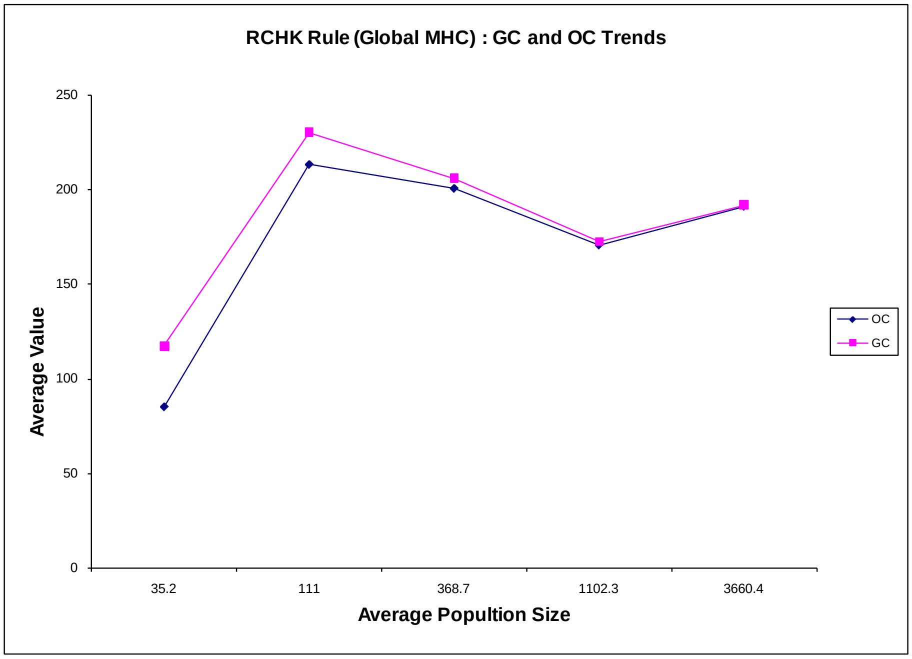


Figure 54. *Car Evaluation Data Set - RCHK (Global MHC) Rule GC and OC Trends*

The DR and FR trends of the RCHK (No MHC) rule for the best scenario illustrated in Figure 55 show that:

- DR is initially equal to 0.61 at an average population size of 35.2, decreases to 0.49 at an average population size of 111 and then slowly increases to 0.69 at an average population size of 3660.
- The FR trend resembles a parabola with a negative gradient in that it increases as the average population size increases, approaches a turning point at an average population size of 368.7 and then decreases.

The DR and FR trends are not good because the DR – FR rate is low. The randomness illustrated by the DR and FR trends is a direct consequence of the application of a single randomly generated global permutation mask to a set of detectors generated with the NSA. The random permutation mask changes the shape space of the resultant detectors in a non-meaningful way.

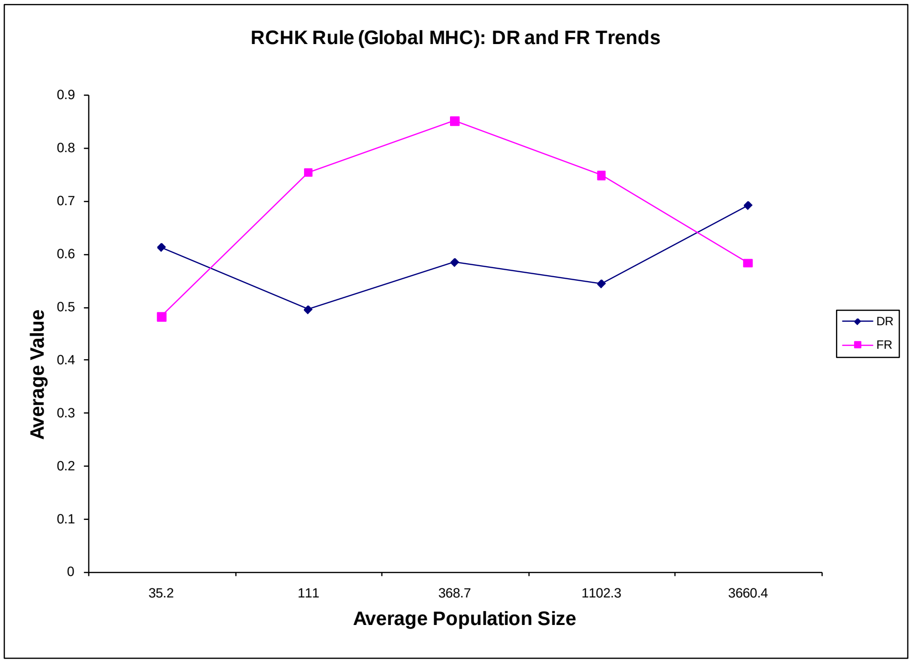


Figure 55. *Car Evaluation Data Set - RCHK (Global MHC) Rule DR and FR Trends*

C.1.5 RCHK (MHC) Rule Trends

The GC and OC trends of the RCHK (MHC) rule for the best scenario illustrated in Figure 56 show that:

- GC fluctuates around 21 as the average population size increases. The GC trend is essentially equal to an average of 21 and the fluctuation in the GC value is caused by the randomness that occurs when executing each test within the test group.

- OC is equal to 5 at an average population size of 44 and increases sharply as the average population size increases until it is equal to GC. The OC trend is bad because it illustrates that the amount of overfitting increases as the number of detectors are increased, i.e. the NSA is distributing detectors in a manner such that their detection regions overlap.

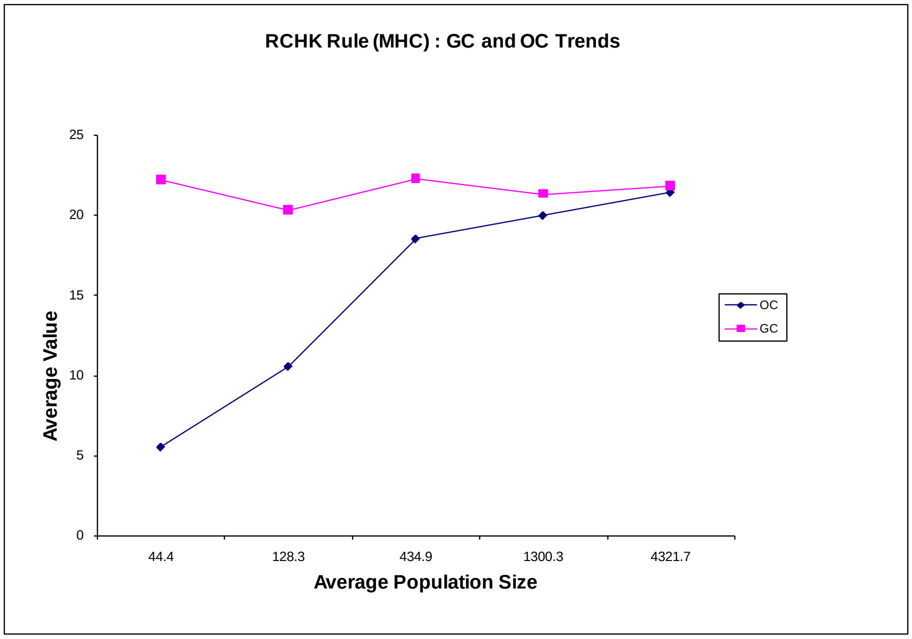


Figure 56. *Car Evaluation Data Set - RCHK (MHC) Rule GC and OC Trends*

The DR and FR trends of the RCHK (MHC) rule for the best scenario illustrated in Figure 57 show that:

- DR increases as the average population size increases.
- FR is initially zero at an average population size of 44.4 and starts increasing slightly at an average population size of 434.9.

The DR and FR trends are good and illustrate that an average population size of 4321.7 is required to maximise the DR value.

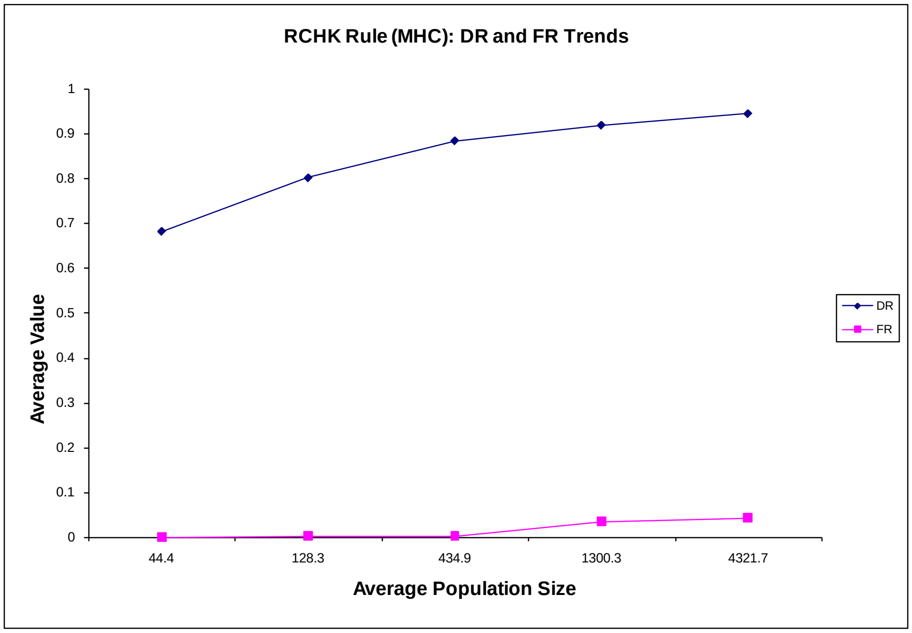


Figure 57. *Car Evaluation Data Set RCHK (MHC) Rule - DR and FR Trends*

C.1.6 RCHK (MHC) Rule vs. Feature Detection-Rule Trends where *r* = *n'*

A comparison between the OC, GC, DR, FR values and actual population size for the feature-detection rule where $r = n'$ and the RCHK (MHC) rule for the best scenarios is presented in Figure 58 to Figure 62 respectively. The figures show that the feature-detection rule has a slightly lower OC and GC value than the RCHK (MHC) rule and that the DR and FR values for the feature-detection rule and RCHK (MHC) rule are almost identical. Figure 62 shows that the feature-detection rule is always able to achieve the target population size, N_c, whereas the RCHK (MHC) rule is not always able to achieve the target population size.

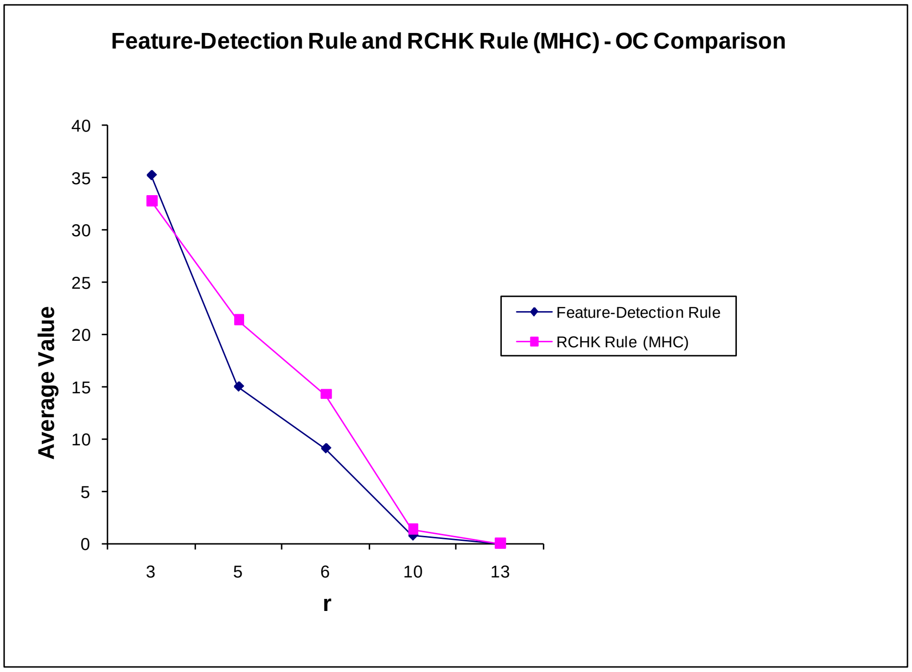


Figure 58. *Car Evaluation Data Set: Feature-Detection Rule and RCHK (MHC) Rule – OC Comparison*

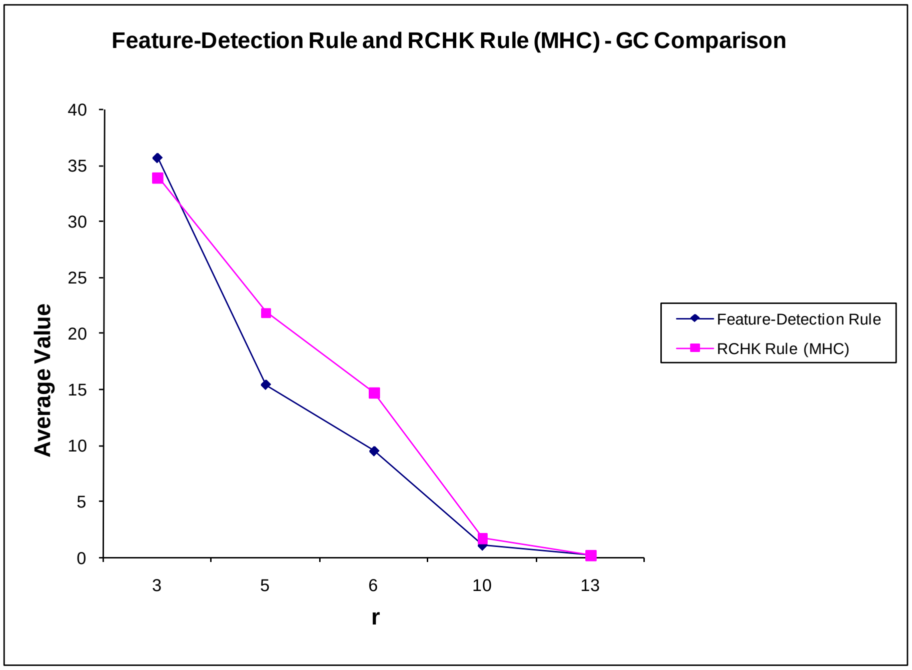


Figure 59. *Car Evaluation Data Set: Feature-Detection Rule and RCHK (MHC) Rule – GC Comparison*

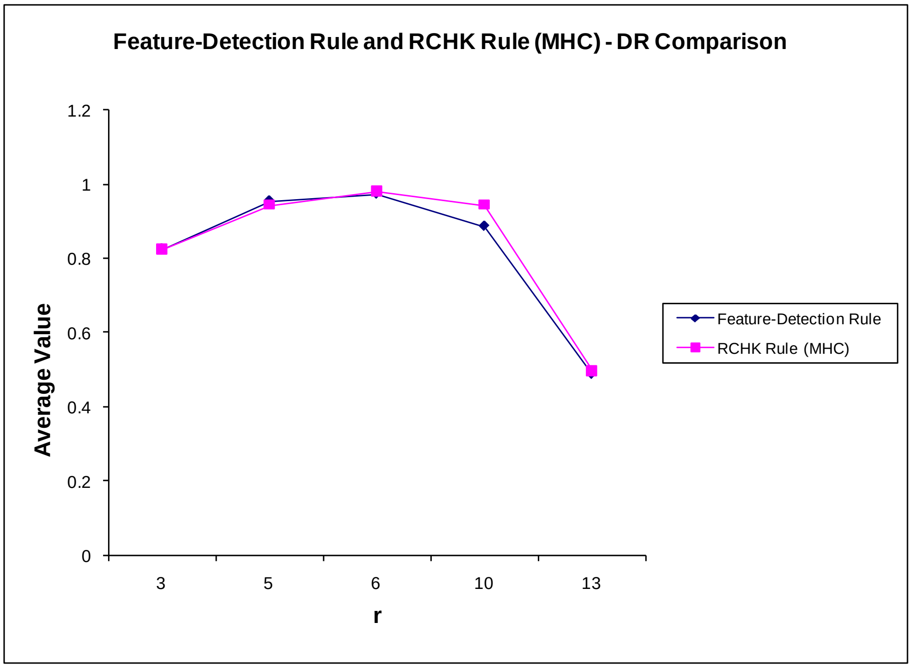


Figure 60. *Car Evaluation Data Set: Feature-Detection Rule and RCHK (MHC) Rule – DR Comparison*

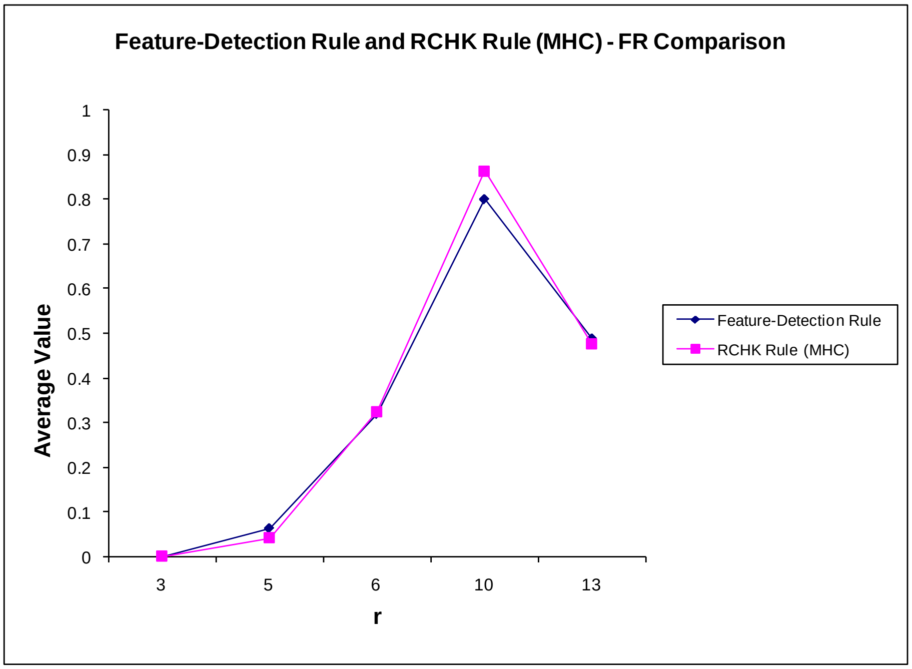


Figure 61. *Car Evaluation Data Set: Feature-Detection Rule and RCHK (MHC) Rule – FR Comparison*

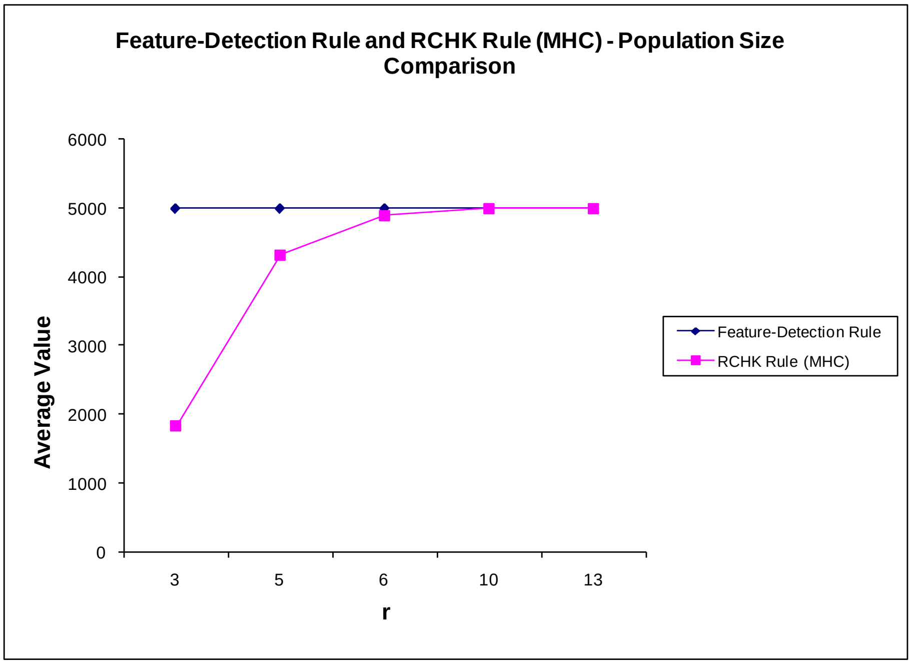


Figure 62. *Car -Evaluation Data Set: Feature-Detection Rule and RCHK (MHC) Rule – Population Size Comparison*

C.2 Iris Experiment Trends

This section contains graphs that illustrate the OC, GC, DR, and FR trends observed for each scenario's best test groups for the Iris experiment. The section is concluded with graphs illustrating the OC, GC, DR, FR, and target population size, N_c vs. actual population size trends for the feature-detection rule and RCHK (MHC) test groups where $r = n'$.

C.2.1 Feature-Detection Rule Trends

The GC and OC trends of the feature-detection rule for the best scenario illustrated in Figure 63 show that:

- GC is equal to 26.62 at an average population size of 50 and only increases slightly as the average population size increases. The GC trend is essentially equal to an average of 26.62 and the fluctuation in the GC value is caused by the randomness that occurs when executing each test within the test group.

- OC is equal to 23.45 at an average population size of 50 and increases slightly until it is equal to GC as the average population size increases. The OC trend is bad as it illustrates that the generated detectors exhibit the same amount of overfitting and generalisation regardless of the average population size.

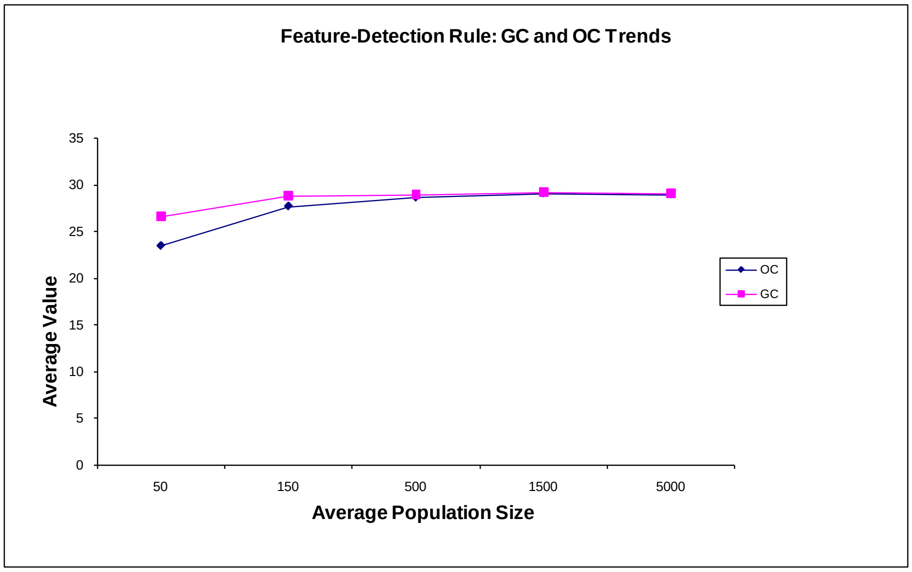


Figure 63. *Iris Data Set – Feature-Detection Rule GC and OC Trends*

The DR and FR trends of the feature-detection rule for the best scenario illustrated in Figure 64 show that:

- DR is constant as the average population size increases.
- FR increases as the average population size increases to 500 and starts gradually decreasing as the average population size increases. The FR trend is bad because it highlights that the FR value increases as more detectors are added to the candidate detector set, i.e. as the average population size increases.

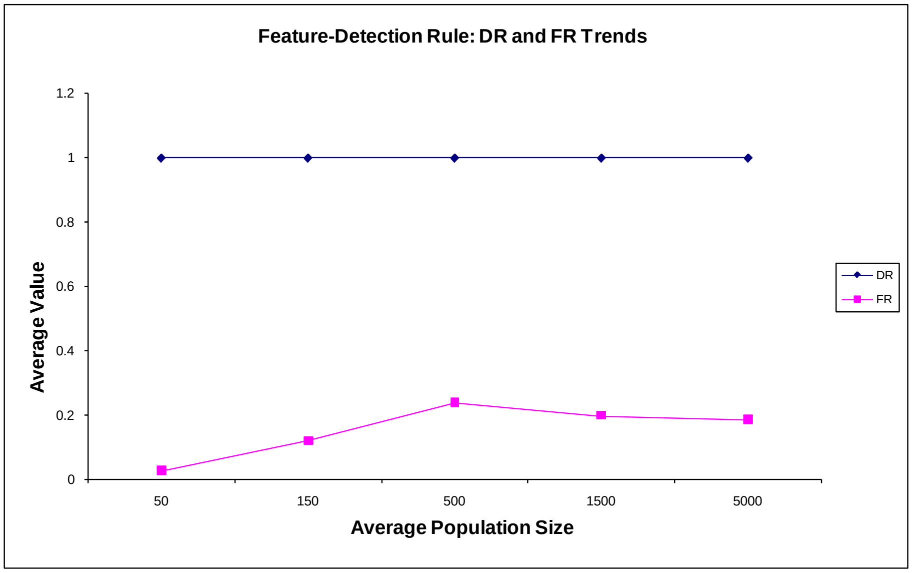


Figure 64. *Iris Data Set – Feature-Detection Rule DR and FR Trends*

C.2.2 HD Rule Trends

The GC and OC trends of the HD rule for the best scenario illustrated in Figure 65 show that:

- GC is equal to 15.95 at an average population size of 4.5 and then increases slightly to 18.41 at an average population size of 14.9 where after it gradually decreases to 16.2. The GC trend is essentially equal to an average of 16.2 and the fluctuation in the GC value is caused by the randomness that occurs when executing each test within the test group.

- OC increases as the average population size increases until it is almost equal to GC at an average population size of 512.7. The OC trend is bad because it illustrates that the amount of overfitting increases as the number of detectors are increased, i.e. the NSA is distributing detectors in a manner such that their detection regions overlap.

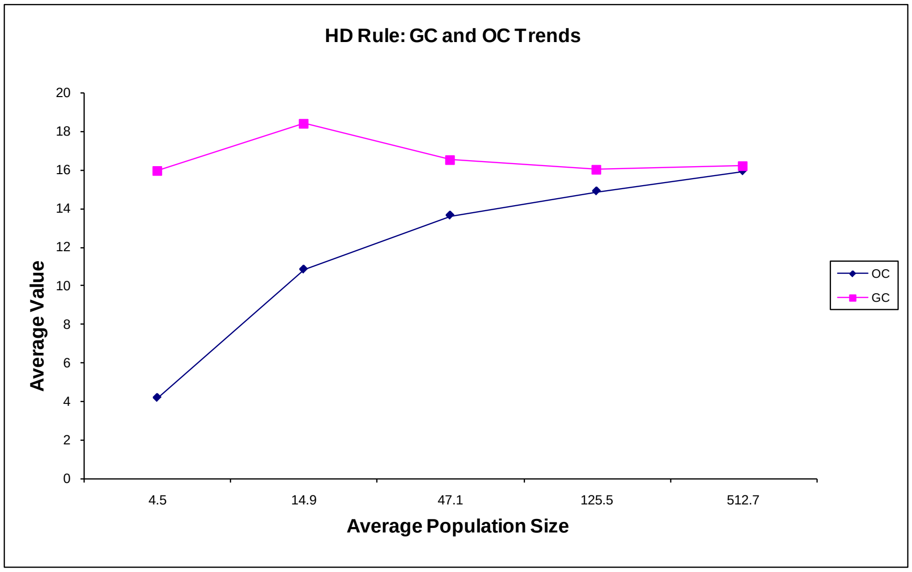


Figure 65. *Iris Data Set - HD Rule GC and OC Trends*

The DR and FR trends of the HD rule for the best scenario illustrated in Figure 66 show that:

- DR increases as the average population size increases i.e. an average population size of 512.7 is required to maximise the detection rate.
- FR increases sharply as the average population size increases and is greater than DR at an average population size of 512.7. The FR trend is bad because it highlights the fact that as the average population size increases (which is necessary to maximise DR) so does the FR value.

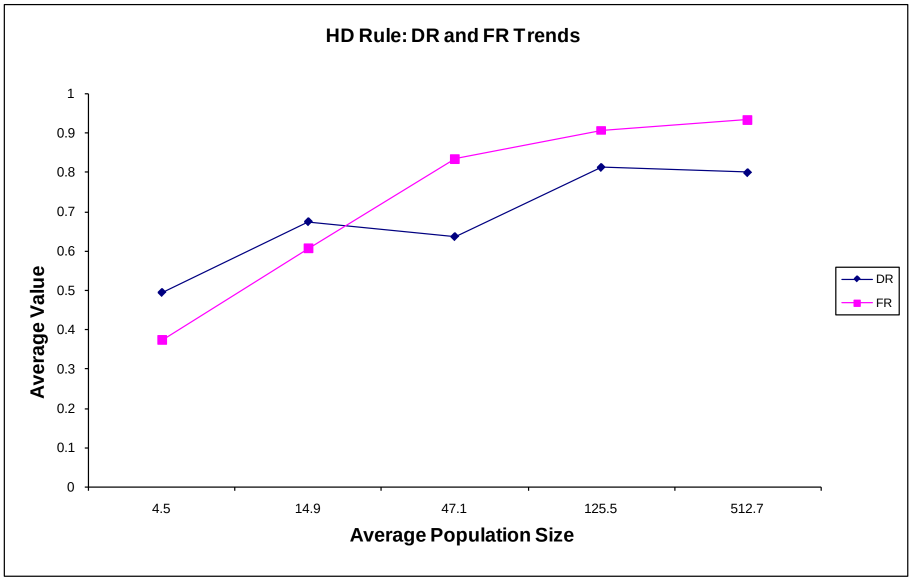


Figure 66. *Iris Data Set – Feature-Detection Rule DR and FR Trend*

C.2.3 RCHK (No MHC) Rule Trends

The GC and OC trends of the RCHK (No MHC) rule for the best scenario illustrated in Figure 67 show that:

- GC is relatively the same as the average population size increases and the fluctuation in the GC value is caused by the randomness that occurs when executing each test within the test group.
- OC increases slowly as the average population size increases until it is equal to GC. The OC trend is bad because it illustrates that the amount of overfitting increases as the number of detectors are increased, i.e. the NSA is distributing detectors in a manner such that their detection regions overlap.

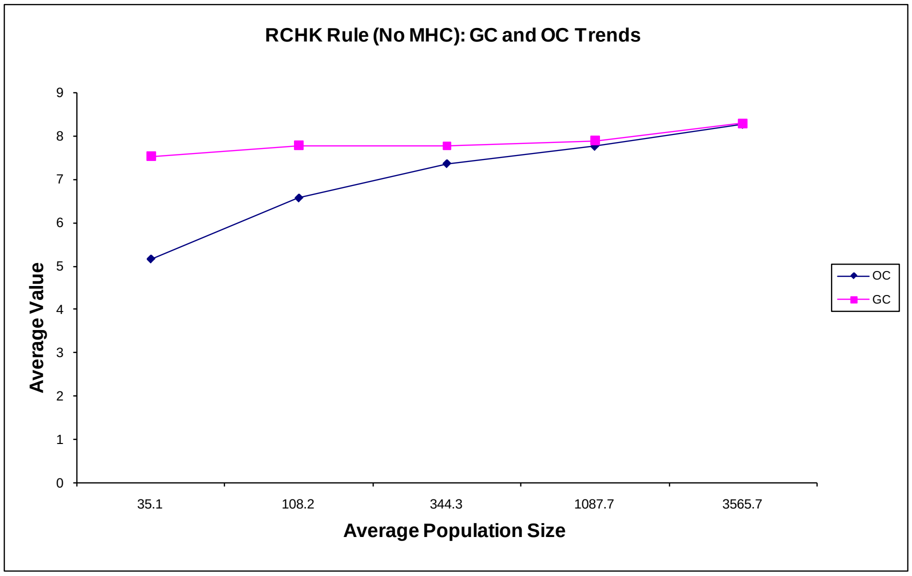


Figure 67. *Iris Data Set - RCHK (No MHC) Rule GC and OC Trends*

The DR and FR trends of the RCHK (No MHC) rule for the best scenario illustrated in Figure 68 show that:

- DR fluctuates slightly around 0.75 as the average population size increases. The DR trend effectively illustrates that an increase in the average population size has no effect in improving the resultant DR value.
- FR initially increases as the average population size increases and starts decreasing slightly at an average population size of 1087.7. The FR trend is bad because it signifies that the FR value increases as the average population size increases.

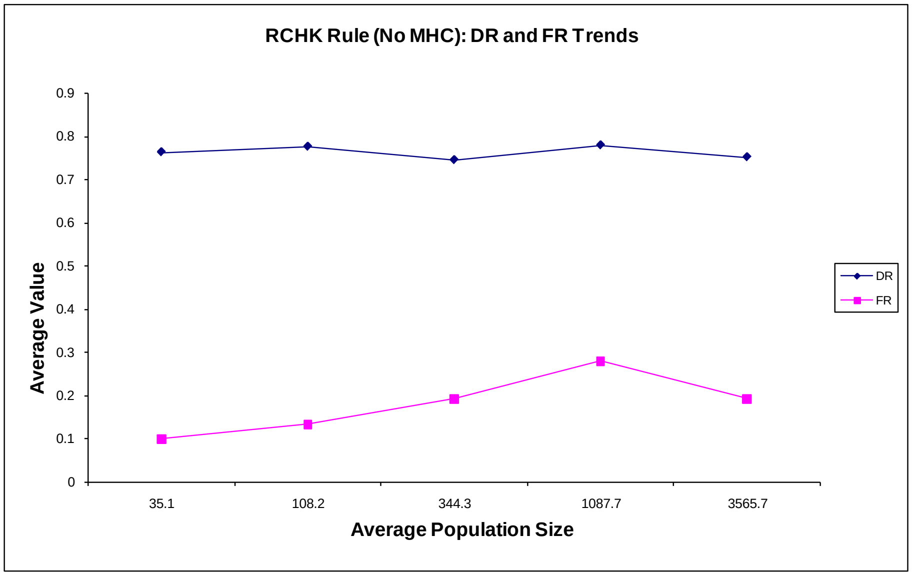


Figure 68. *Iris Data Set - RCHK (No MHC) Rule DR and FR Trends*

C.2.4 RCHK (Global MHC) Rule Trends

The GC and OC trends of the RCHK (Global MHC) rule for the best scenario illustrated in Figure 69 show that:

- GC is initially equal to 0.0019 at an average population size of 50 and then decreases to 0 as the average population size increases. The GC trend is bad because the amount of generalisation decreases as the average population size increases.
- OC is constant at 0 as the average population size increases. The OC trend is good because it illustrates the NSA is distributing detectors in a manner such that their detection regions do not overlap.

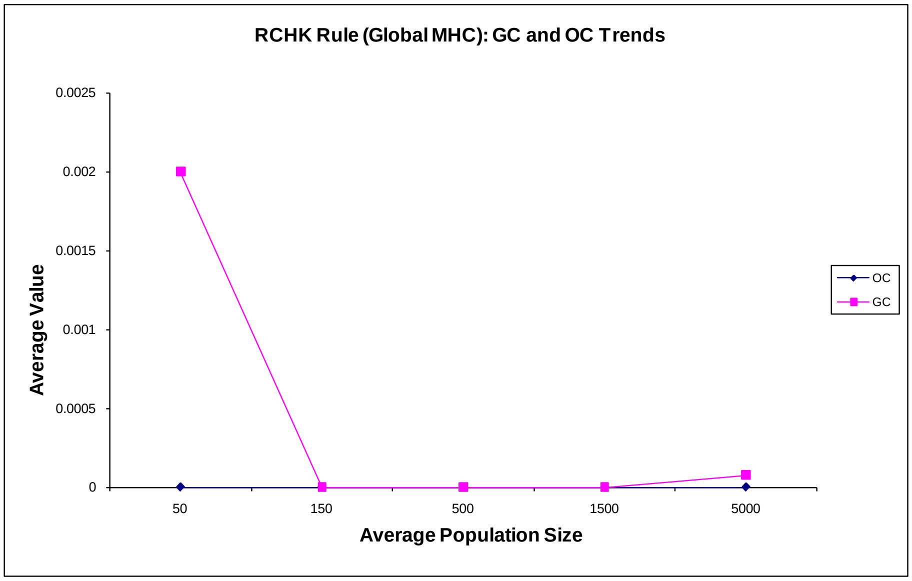


Figure 69. *Iris Data Set - RCHK (Global MHC) Rule DR and FR Trends*

The DR and FR trends of the RCHK (Global MHC) rule for the best scenario illustrated in Figure 70 show that:

- DR is constant at 0.5 as the average population size increases. The DR trend illustrates that the magnitude of the average population size does not affect the DR value.
- FR is constant at 0 as the average population size increases. The FR trend is good because it is constantly 0 regardless of the average population size.

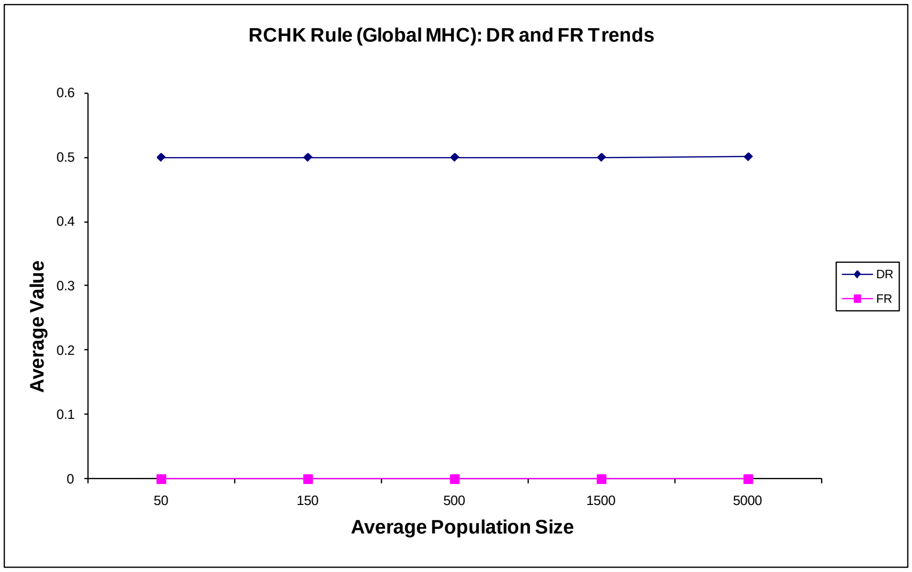


Figure 70. *Iris Data Set - RCHK (Global MHC) Rule DR and FR Trends*

C.2.5 RCHK (MHC) Rule Trends

The GC and OC trends of the RCHK (MHC) rule for the best scenario illustrated in Figure 71 show that:

- GC is initially constant at 0. As the average population size increases, GC suddenly increases to 1.33 at an average population size of 1500 and then decreases to 0.0004. The GC trend is bad because it is effectively equal to 0, indicating that none of the generated detectors are able to generalise across the data set.

- OC is constant at 0 as the average population size increases. The OC trend is good because it illustrates that the generated detectors are distributed in a manner such that their detection regions do no overlap.

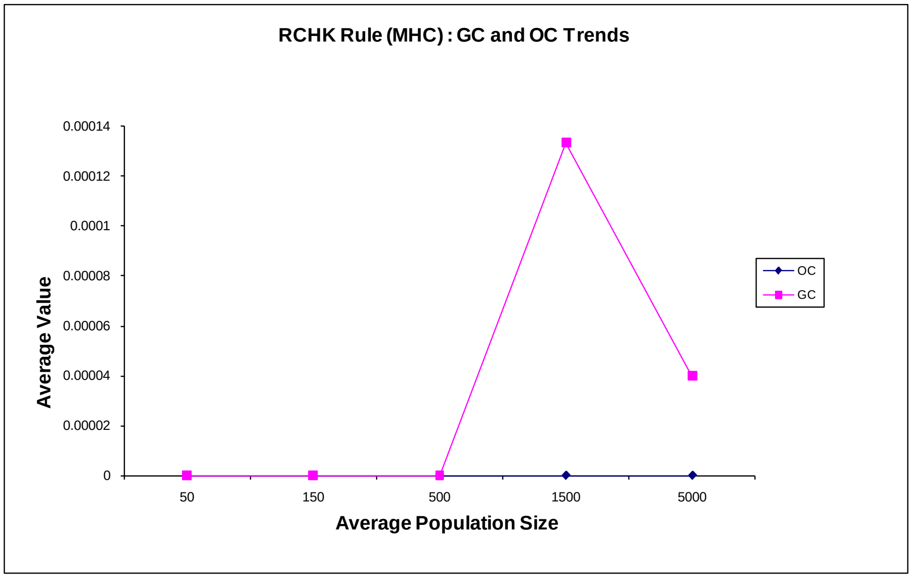


Figure 71. *Iris Data Set - RCHK (MHC) Rule GC and OC Trends*

The DR and FR trends of the RCHK (MHC) rule for the best scenario illustrated in Figure 72 show that:

- DR is constant at 0.5 as the average population size increases. The DR trend illustrates that the magnitude of the average population size has no effect on the DR value. The DR trend is bad because it illustrates that the generated detectors will never be able to achieve a maximum DR rate of 1.
- FR is constant at 0 as the average population size increases. The FR rate is good because it is constant at 0.

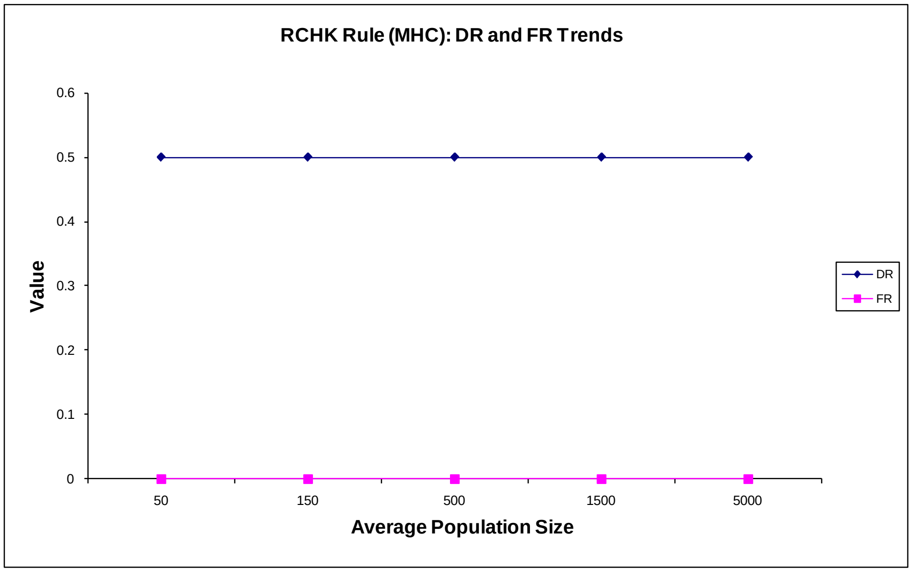


Figure 72. *Iris Data Set - RCHK (MHC) Rule DR and FR Trends*

C.2.6 RCHK (MHC) Rule vs. Feature-Detection Rule Trends

A comparison between the OC, GC, DR, FR and actual population size for the feature-detection rule where $r = n'$ and the RCHK (MHC) rule for the best scenarios is presented in Figure 73 to Figure 77 respectively. The figures show that the feature-detection rule has slightly lower OC, GC values than the RCHK (MHC) rule while the DR and FR values are very similar. Figure 77 shows that the feature-detection rule is always able to achieve the target population size, N_c, whereas the RCHK (MHC) rule is not always able to achieve the target population size.

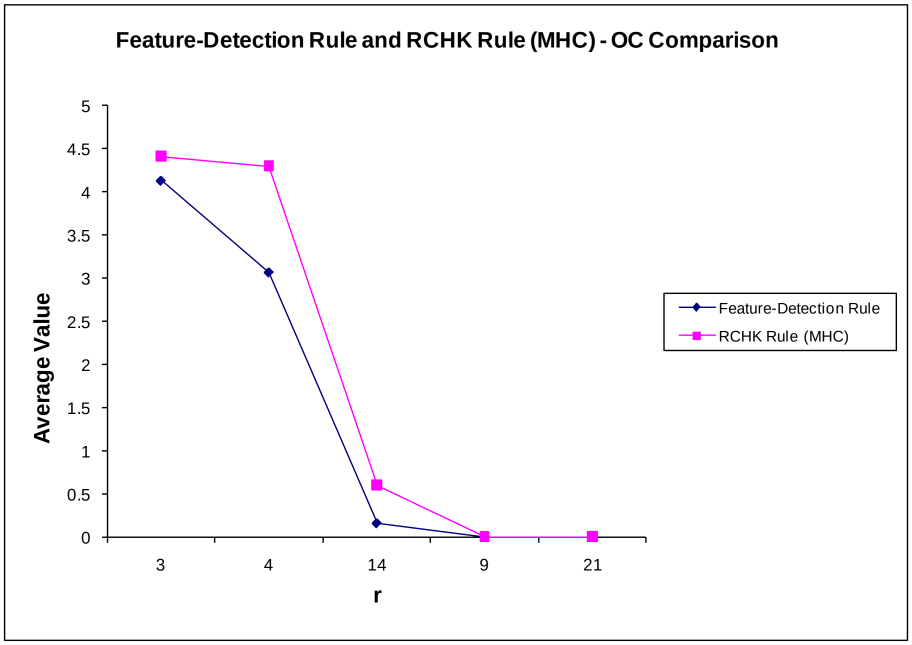


Figure 73. *Iris Data Set: Feature-Detection Rule and RCHK (MHC) Rule – OC Comparison*

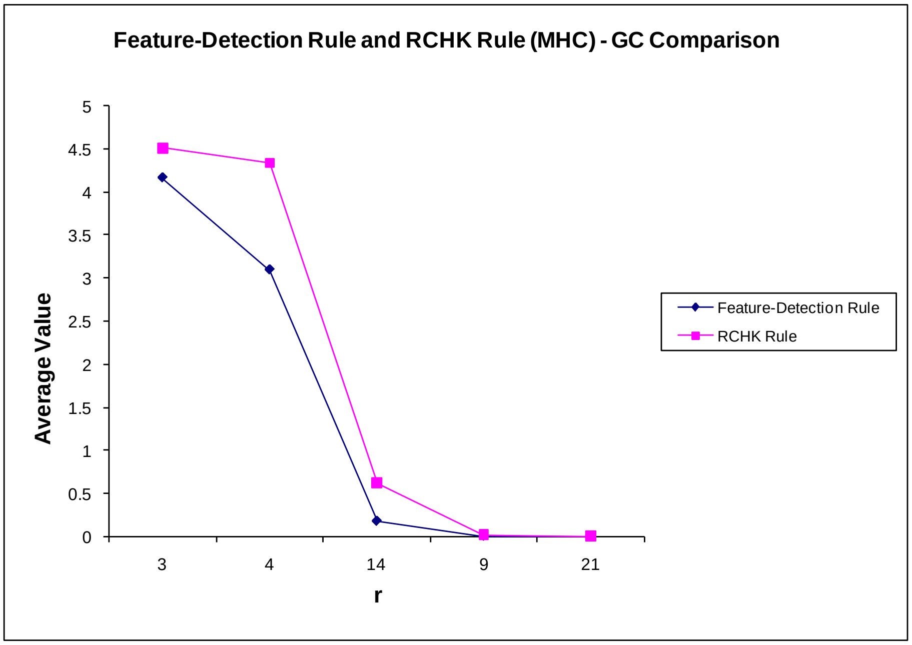


Figure 74. *Iris Data Set: Feature-Detection Rule and RCHK (MHC) Rule – GC Comparison*

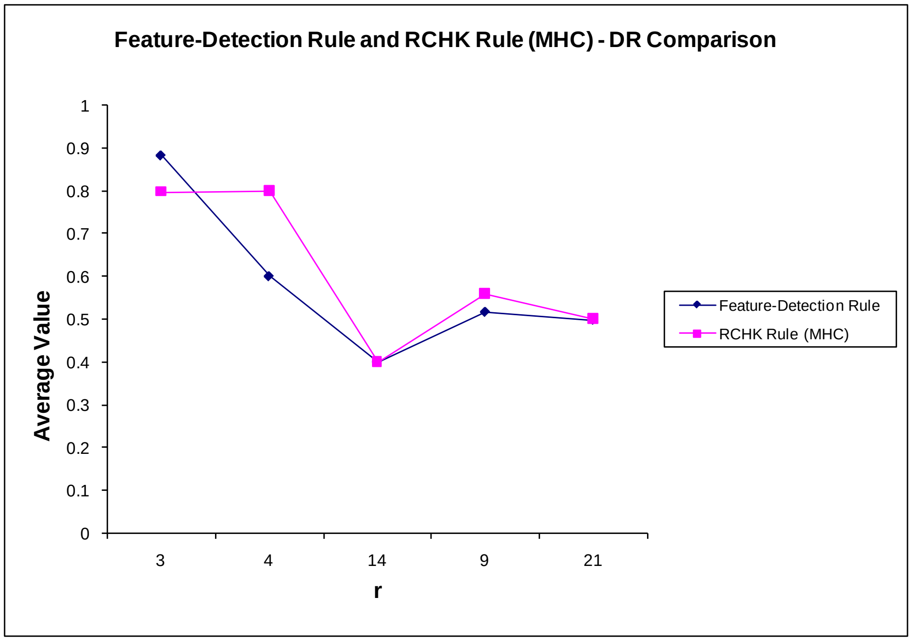


Figure 75. *Iris Data Set: Feature-Detection Rule and RCHK (MHC) Rule – DR Comparison*

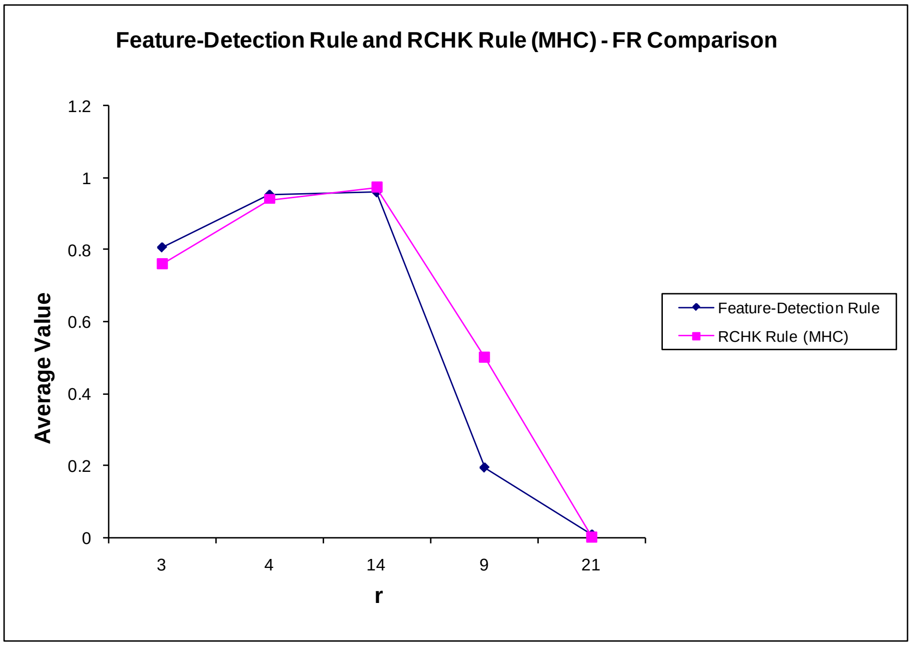


Figure 76. *Iris Data Set: Feature-Detection Rule and RCHK (MHC) Rule – FR Comparison*

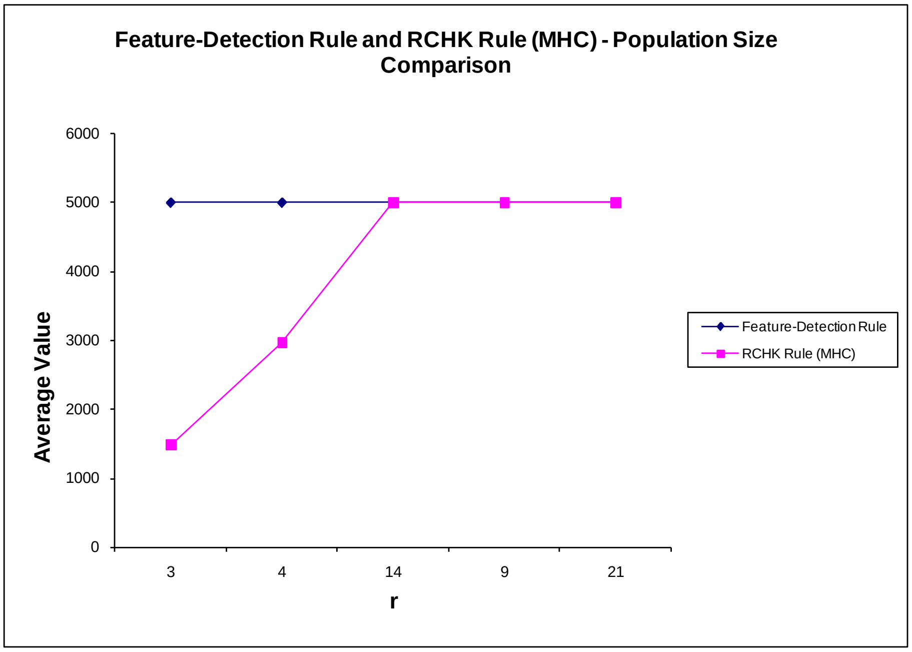


Figure 77. *Iris Data Set: Feature-Detection Rule and RCHK (MHC) Rule – Population Size Comparison*

C.3 Wisconsin Breast Cancer Experiment Trends

This section contains graphs that illustrate the OC, GC, DR, and FR trends observed for each scenario's best test groups for the Wisconsin Breast Cancer experiment. The section is concluded with graphs illustrating the OC, GC, DR, FR, and target population size, N_c vs. actual average population size trends for the feature-detection rule and RCHK (MHC) rule test groups where $r = n'$.

C.3.1 Feature-Detection Rule Trends

The GC and OC trends of the feature-detection rule for the best scenario illustrated in Figure 78 show that:

- GC is initially equal to 34 at an average population size of 48.4, increases slightly as the average population size increases, starts decreasing at an average population size of 1405.6 and then starts increasing again at 47404.2. The GC trend is essentially equal to an average of 34 and the fluctuation in the GC value is caused by the randomness that occurs when executing each test within the test group.

- OC is initially equal to 29 at an average population size of 48.4, increases until it is equal to GC and then follows the same trend as GC. The OC trend is bad as it illustrates that the generated detectors exhibit the same amount of overfitting and generalisation regardless of the average population size.

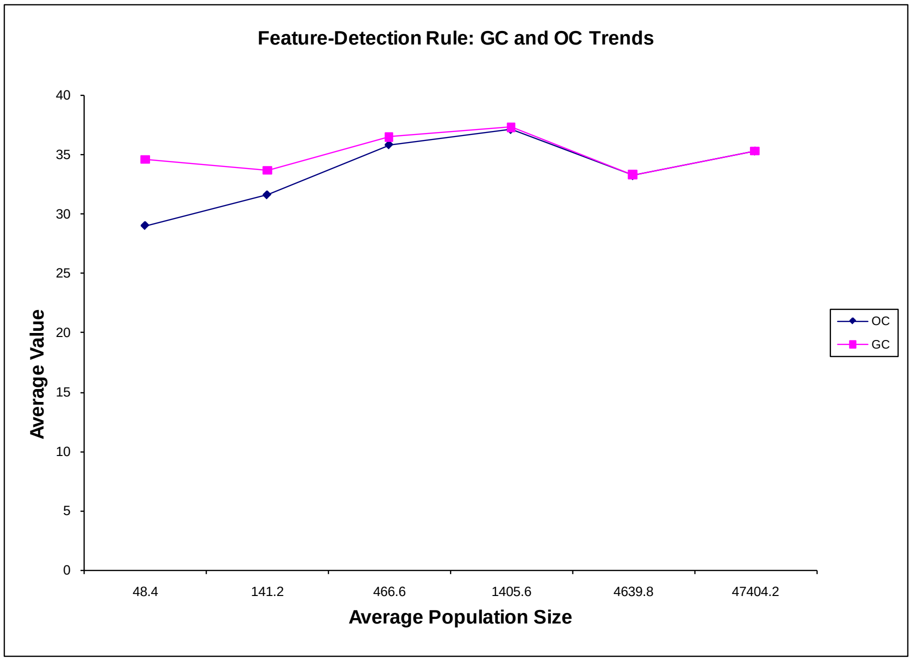


Figure 78. *Wisconsin Breast-Cancer Data Set – Feature-Detection Rule GC and OC Trends*

The DR and FR trends of the feature-detection rule for the best scenario illustrated in Figure 79 show that:

- DR is initially equal to 0.88 at an average population size of 48.4, increases as the average population size increases and then remains constant at 1.

- FR is initially equal to 0.02 at an average population size of 48.4 and increases slightly as the average population size increases.

The DR and FR trends are good and illustrate that the DR value is maximised at an average population size of 466.6.

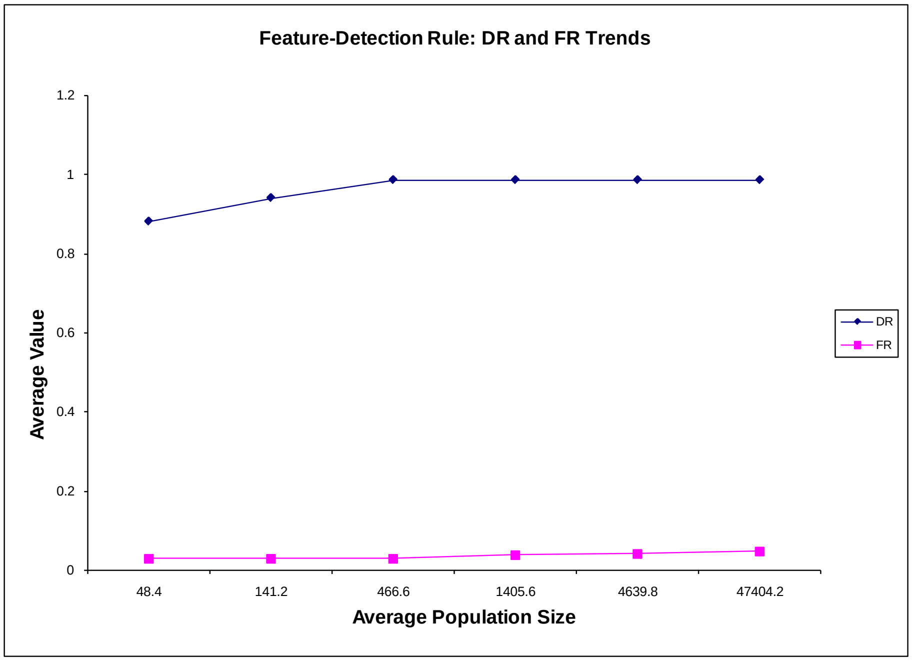


Figure 79. *Wisconsin Breast-Cancer Data Set –Feature-Detection Rule DR and FR Trends*

C.3.2 HD Rule Trends

The GC and OC trends of the HD rule for the best scenario illustrated in Figure 80 show that:

- GC is initially equal to 73.56 at an average population size of 1.6, increases to 92.34 at an average population size of 5.1, decreases to 86.82 at a population size of 13.1 and then increases to 92.87 as the average population size increases. The GC trend is essentially equal to an average of 92 and the fluctuation in the GC value is caused by the randomness that occurs when executing each test within the test group.
- OC increases as the average population size increases until it is almost equal to GC. The OC trend is bad because it illustrates that the amount of overfitting increases as the number of detectors are increased, i.e. the NSA is distributing detectors in a manner such that their detection regions overlap.

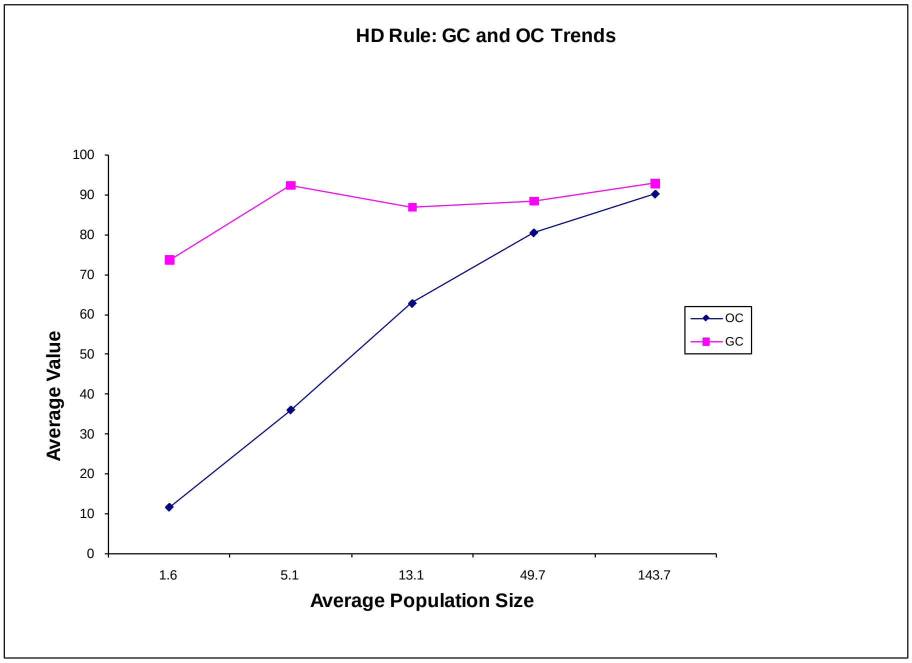


Figure 80. *Wisconsin Breast-Cancer Data Set - HD Rule GC and OC Trends*

The DR and FR trends of the HD rule for the best scenario illustrated in Figure 81 show that:

- DR increases as the average population size increases.
- FR initially increases at a low rate as the average population size increases until the average population size is equal to 13.1 where FR starts increasing at a higher rate.

The DR trend is good because the DR value increases as the population size increases whereas the FR trend is bad because the FR value increases as the population size increases.

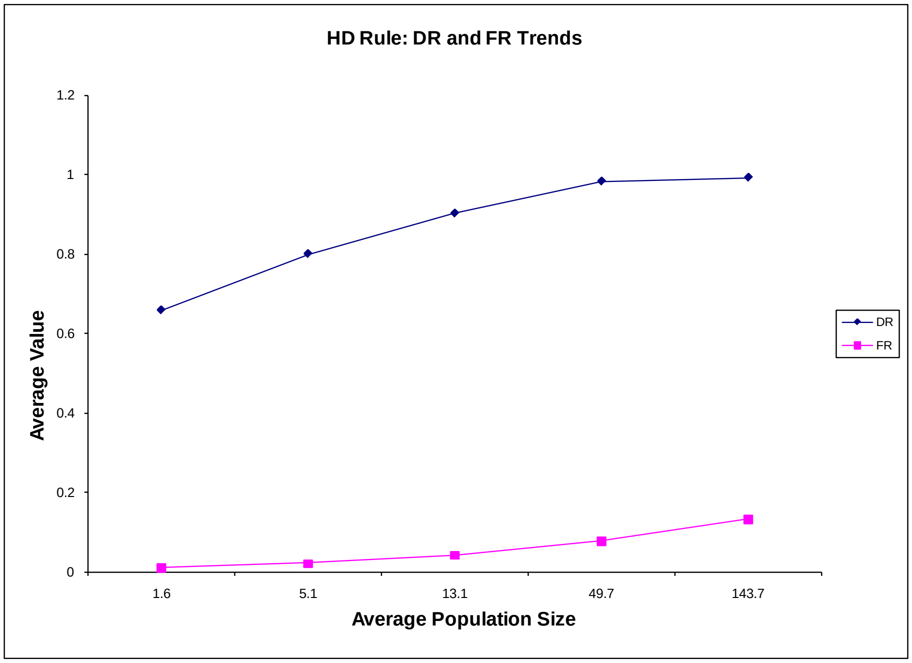


Figure 81. *Wisconsin Breast-Cancer Data Set - HD Rule DR and FR Trend*

C.3.3 RCHK (No MHC) Rule Trends

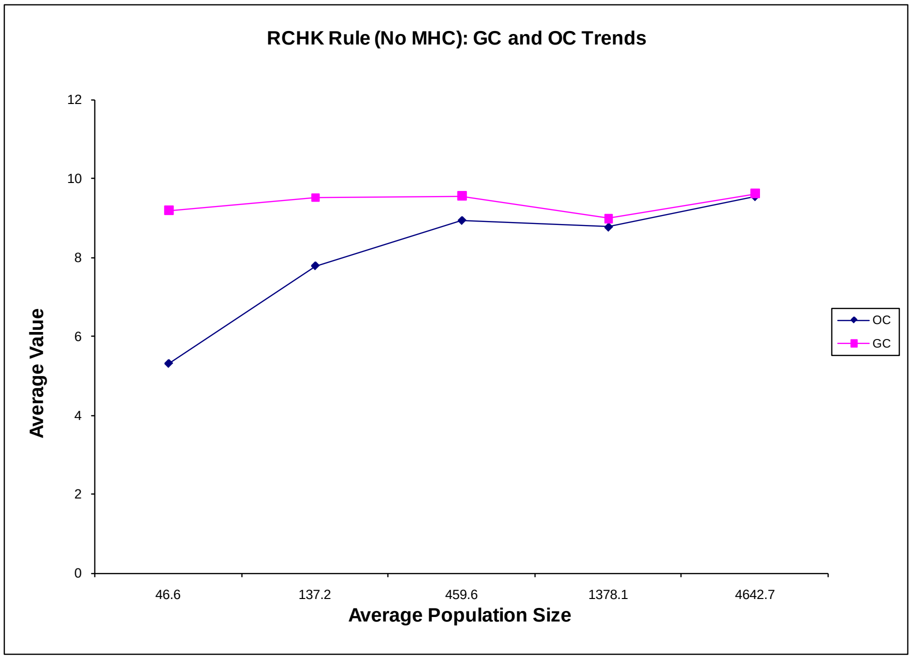


Figure 82. *Wisconsin Breast-Cancer Data Set - RCHK (No MHC) Rule DR and FR Trends*

C.3.4 RCHK (Global MHC) Rule Trends

The GC and OC trends of the RCHK (Global MHC) rule for the best scenario illustrated in Figure 83 show that both GC and OC are 0 as the average population size increases. The GC trend is bad because it indicates that the generated detectors are not able to generalise across the data set whereas the OC trend is good because it indicates that the generated detectors are distributed in a manner such that their detection regions do not overlap.

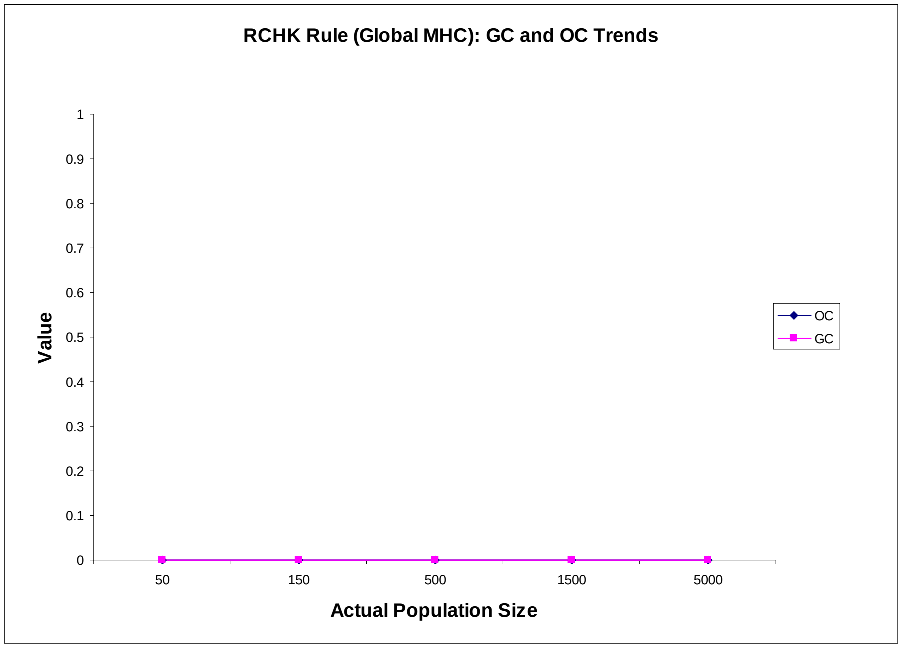


Figure 83. *Wisconsin Breast-Cancer Data Set - RCHK (Global MHC) Rule GC and OC Trends*

The DR and FR trends of the RCHK (Global MHC) rule for the best scenario illustrated in Figure 84 show that:

- DR is constant at 0.5 as the average population size increases.
- FR is constant at 0 as the average population size increases.

The DR and FR trends are a result of TP always being equal to 1 and FN always being equal to 1.

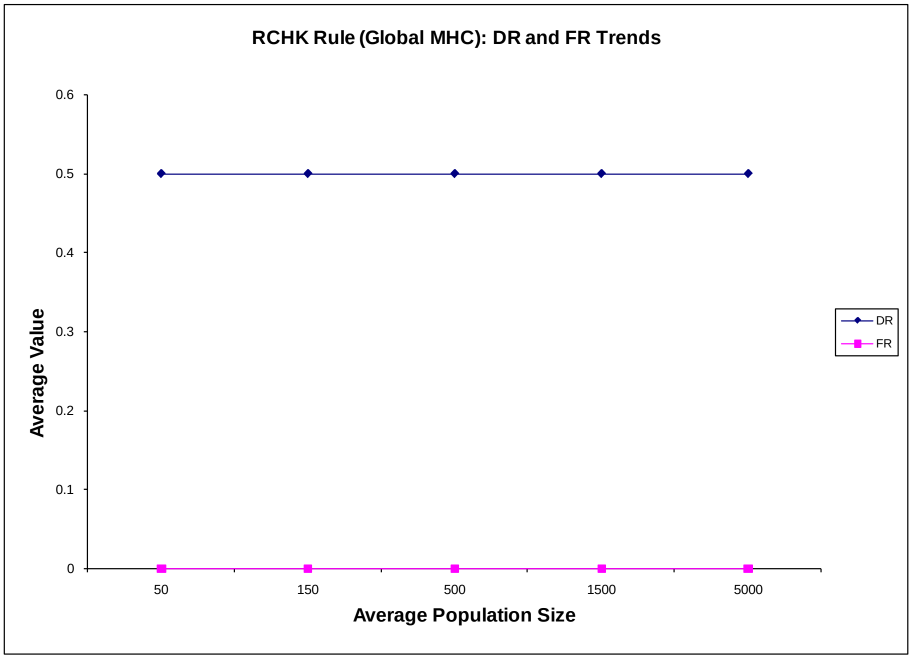


Figure 84. *Wisconsin Breast-Cancer Data Set - RCHK (Global MHC) Rule DR and FR Trends*

C.3.5 RCHK (MHC) Rule Trends

The GC and OC trends of the RCHK (MHC) rule for the best scenario illustrated in Figure 85 show that:

- GC is initially equal to 21 at an average population size of 12.8 and fluctuates between 21 and 23 as the population size increases. The GC trend is essentially equal to an average of 22 and the fluctuation in the GC value is caused by the randomness that occurs when executing each test within the test group.
- OC increases sharply as the average population size increases until it is equal to GC. The OC trend is bad because it illustrates that the amount of overfitting increases as the number of detectors are increased, i.e. the NSA is distributing detectors in a manner such that their detection regions overlap.

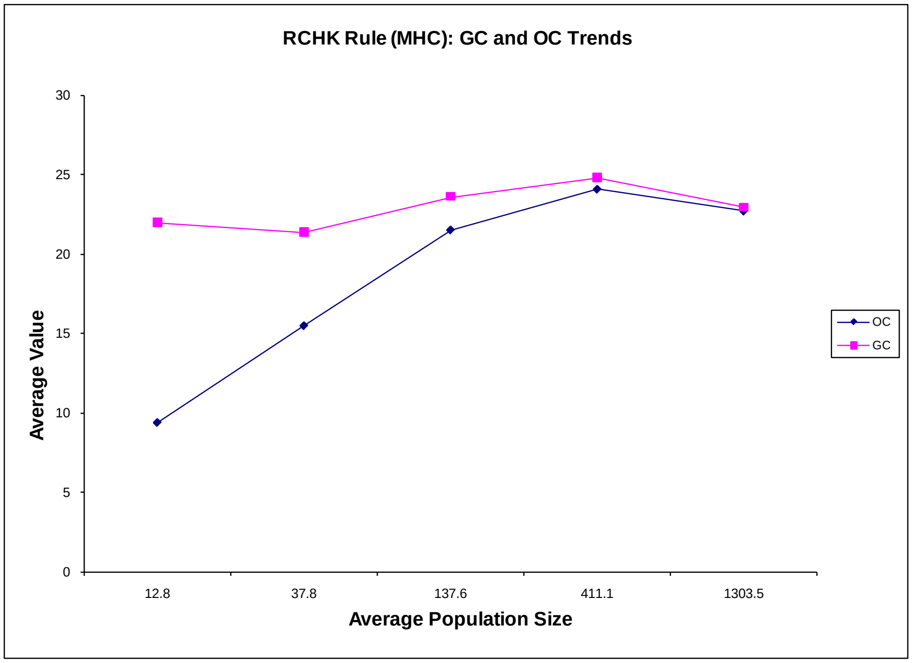


Figure 85. *Wisconsin Breast-Cancer Data Set - RCHK (MHC) Rule GC and OC Trends*

The DR and FR trends of the RCHK (MHC) rule for the best scenario illustrated in Figure 86 show that:

- DR increases as the average population size increases until it is equal to 1.
- FR increases as the average population size increases.

The DR trend is good because it approaches 1 as the average population size increases whereas the FR trend is bad because it increases more rapidly as the average population size increases.

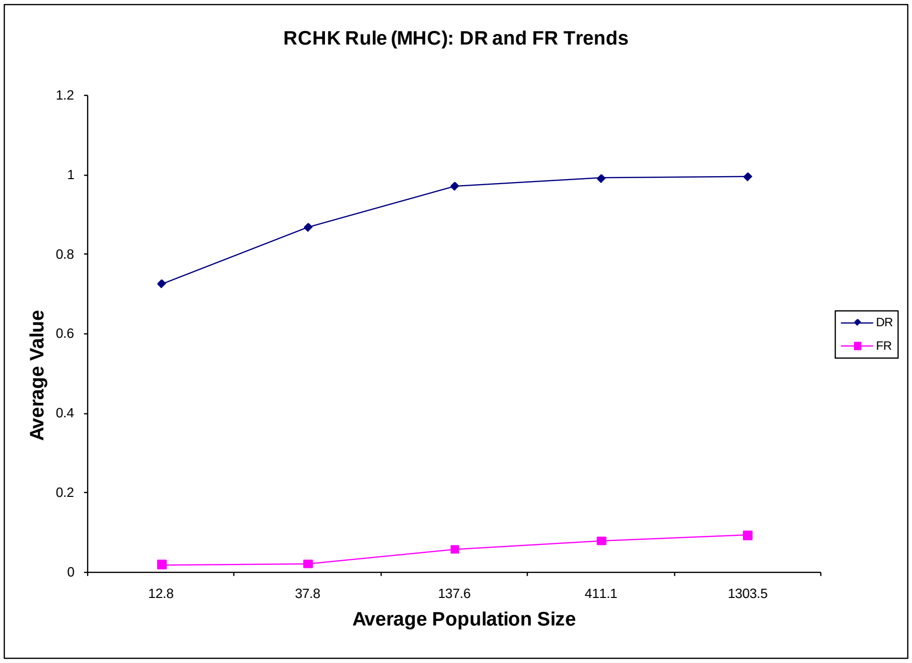


Figure 86. *Wisconsin Breast-Cancer Data Set - RCHK (MHC) Rule DR and FR Trend*

C.3.6 RCHK (MHC) Rule vs. Feature-Detection Rule Trends

A comparison between the OC, GC, DR, FR and actual population size for the feature-detection rule where $r = n'$ and the RCHK (MHC) rule for the best scenarios is presented in Figure 87 to Figure 91 respectively. The figures show that the feature-detection rule has slightly lower OC, GC, DR and FR values than the RCHK (MHC) rule. Figure 91 shows that the feature-detection rule is always able to achieve the target population size, N_c, whereas the RCHK (MHC) rule is not always able to achieve the target population size.

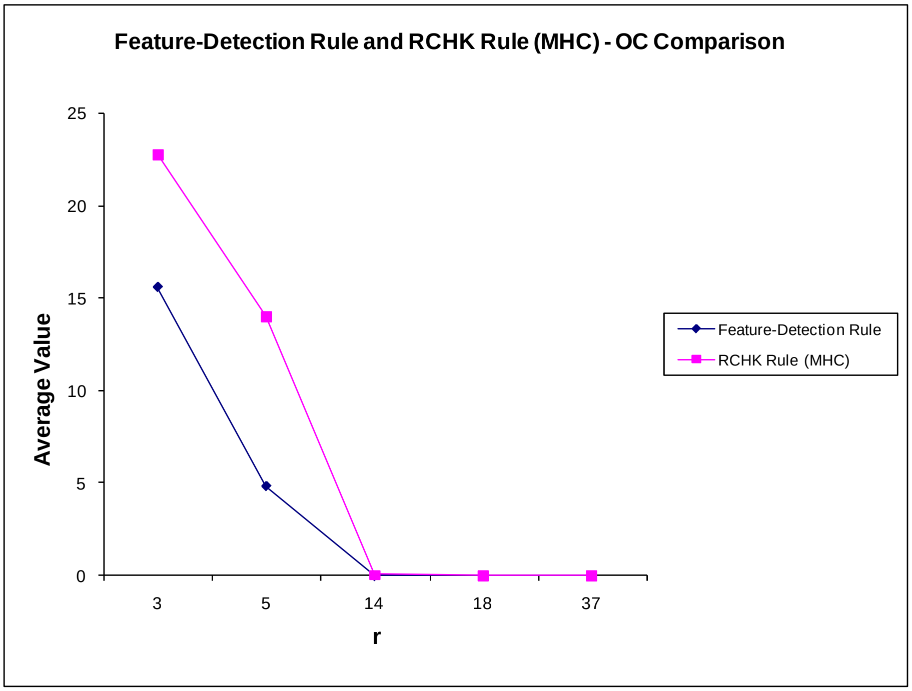


Figure 87. *Wisconsin Breast-Cancer Data Set: Feature-Detection Rule and RCHK (MHC) – Rule OC Comparison*

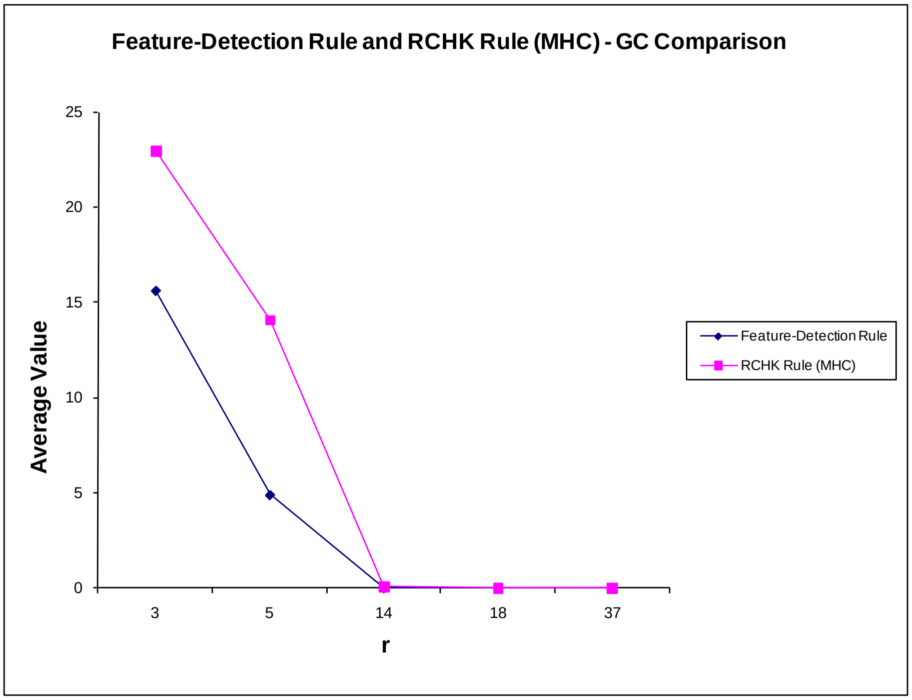


Figure 88. *Wisconsin Breast-Cancer Data Set: Feature-Detection Rule and RCHK (MHC) Rule – GC Comparison*

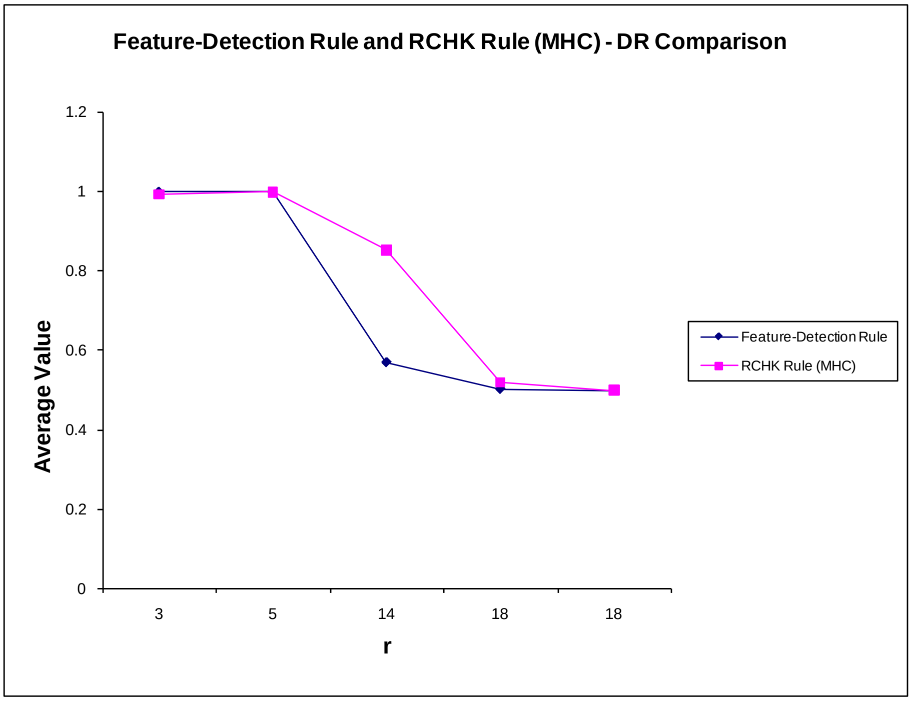


Figure 89. *Wisconsin Breast-Cancer Data Set: Feature-Detection Rule and RCHK (MHC) Rule – DR Comparison*

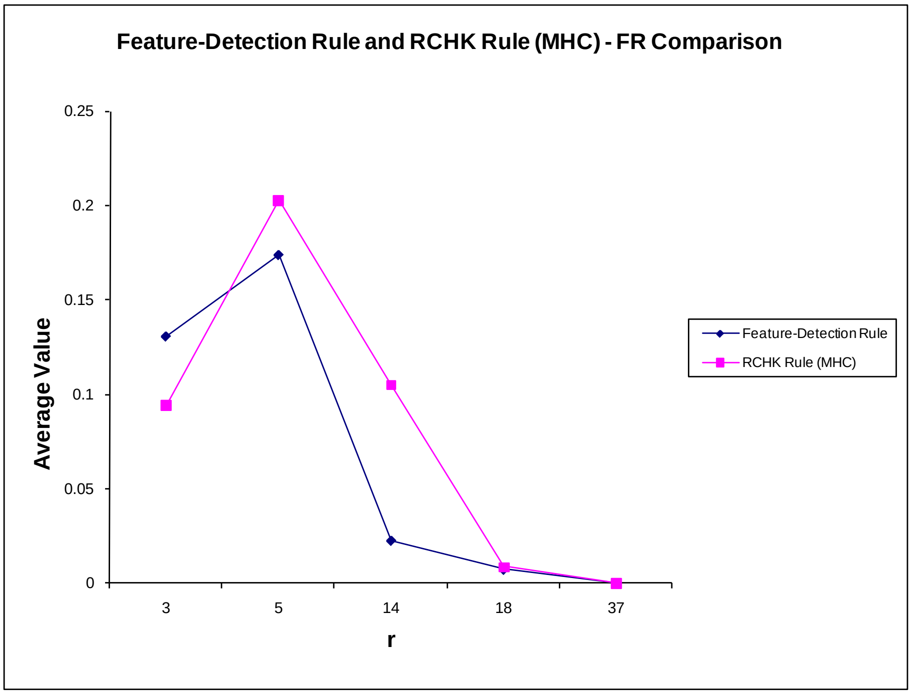


Figure 90. *Wisconsin Breast-Cancer Data Set: Feature-Detection Rule and RCHK (MHC) Rule – FR Comparison*

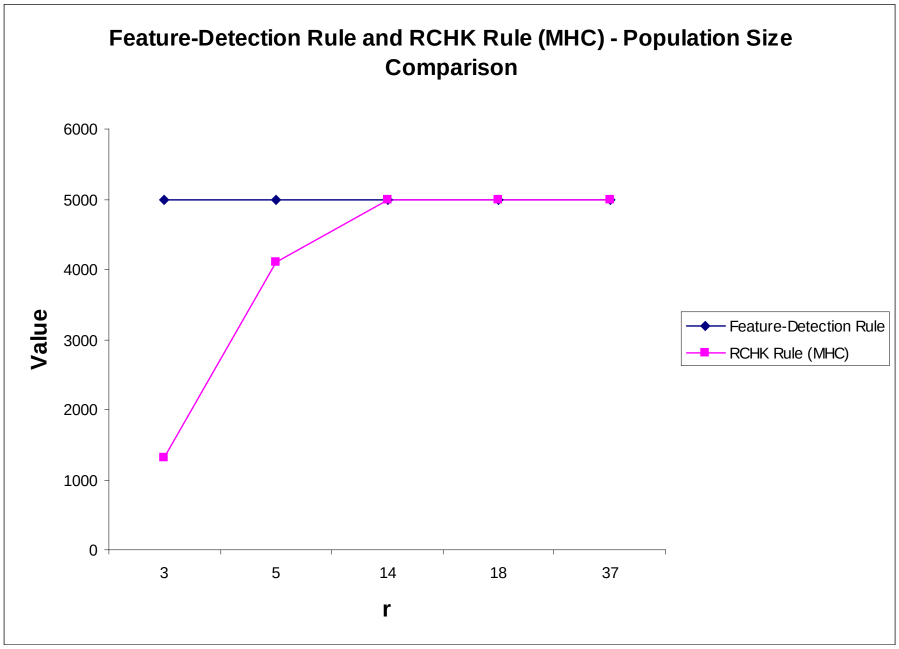


Figure 91. *Wisconsin Breast-Cancer Data Set: Feature-Detection Rule and RCHK (MHC) Rule – Population Size Comparison*

C.4 Glass Experiment Trends

This section contains graphs that illustrate the OC, GC, DR, and FR trends observed for each scenario's best test groups for the Glass experiment. The section is concluded with graphs illustrating the OC, GC, DR, FR, and target population size, N_c vs. actual average population size trends for the feature-detection rule and RCHK (MHC) rule test groups where $r = n'$.

C.4.1 Feature-Detection Rule Trends

The GC and OC trends of the feature-detection rule for the best scenario illustrated in Figure 92 show that:

- GC is initially 0 as the average population size increases and then suddenly increases to 0.000012 at an average population size of 50000.
- OC is constant at 0 as the average population size increases.

The GC trend is bad because it illustrates that the generated detectors are not able to generalise across the data set.

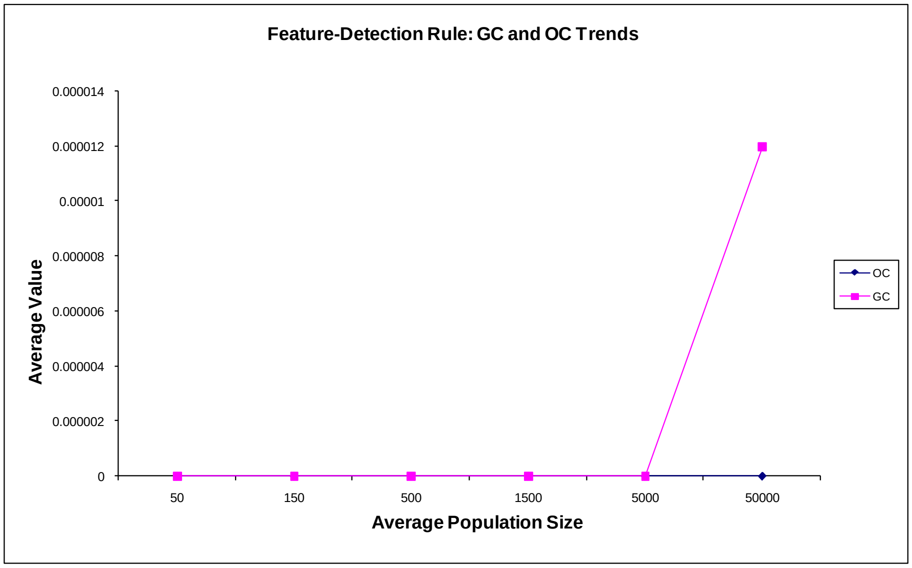


Figure 92. *Glass Data Set – Feature-Detection Rule GC and OC Trends*

The DR and FR trends of the feature-detection rule for the best scenario illustrated in Figure 93 show that:

- DR is constant at 0.49 as the average population size increases.
- FR is constant at zero as the average population size increases and then increases slightly when the average population size reaches 50000.

The DR trend is bad because it does not improve as the magnitude of the average population size increases whereas the FR trend is good because it is constantly zero regardless of the magnitude of the average population size.

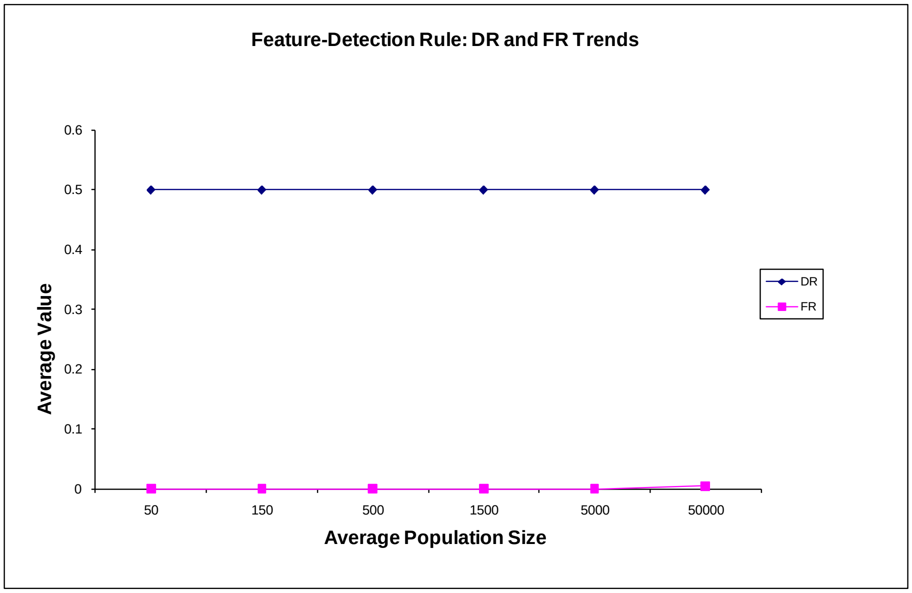


Figure 93. *Glass Data Set – Feature-Detection Rule DR and FR Trends*

C.4.2 HD Rule Trends

The GC and OC trends of the HD for the best scenario illustrated in Figure 94 show that:

- GC is initially 0 as the average population size increases and then increases sharply to 0.000012 at an average population size of 50000. The GC trend is bad because it illustrates that the generated detectors are not able to generalise across the data set.
- OC is constant at 0 as the average population size increases. The OC trend is good because it indicates that the NSA is distributing the detectors in a manner such that their detection regions do not overlap.

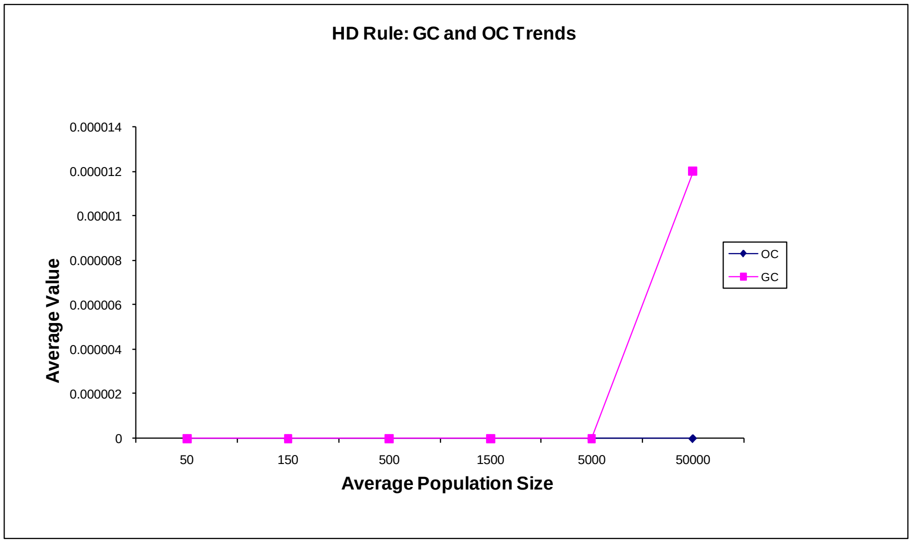


Figure 94. *Glass Data Set - HD Rule GC and OC Trend*

The DR and FR trends of the HD for the best scenario illustrated in Figure 95 show that:

- DR is constant at 0.5 as the average population size increases.
- FR is constant at 0 as the average population size increases and increases slightly to 0.009 at an average population size of 50000.

The DR trend is bad because it illustrates that the DR value cannot be improved by increasing the magnitude of the average population size whereas the FR trend is good because it is approximately zero as the average population size increases.

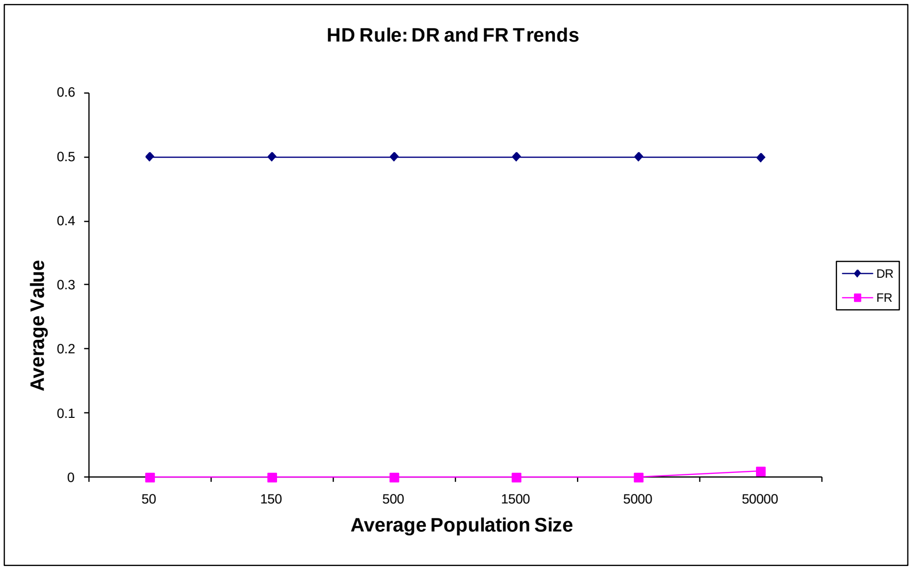


Figure 95. *Glass Data Set - HD Rule DR and FR Trends*

C.4.3 RCHK (No MHC) Rule Trends

The GC and OC trends of the RCHK (No MHC) rule for the best scenario illustrated in Figure 96 show that both GC and OC are constant at 0 as the average population size increases. The GC trend is bad because it illustrates that the generated detectors are not able to generalise across the dataset whereas the OC trend is good because it illustrates that the detectors are distributed in a manner such that their detection regions do not overlap.

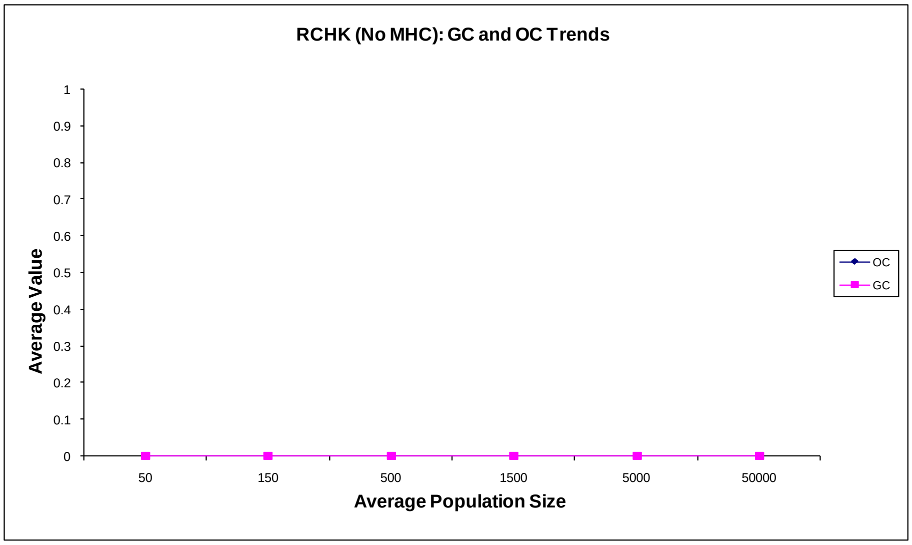


Figure 96. *Glass Data Set - RCHK (No MHC) Rule GC and OC Trends*

The DR and FR trends of the RCHK (No MHC) rule for the best scenario illustrated in Figure 97 show that:

- DR is constant at 0.5 as the average population size increases.
- FR is constant at 0 as the average population size increases.

The DR trend is bad because it illustrates that increasing the magnitude of the average population size does not improve the DR value. The FR trend is good because it is constant at 0 regardless of the magnitude of the average population size.

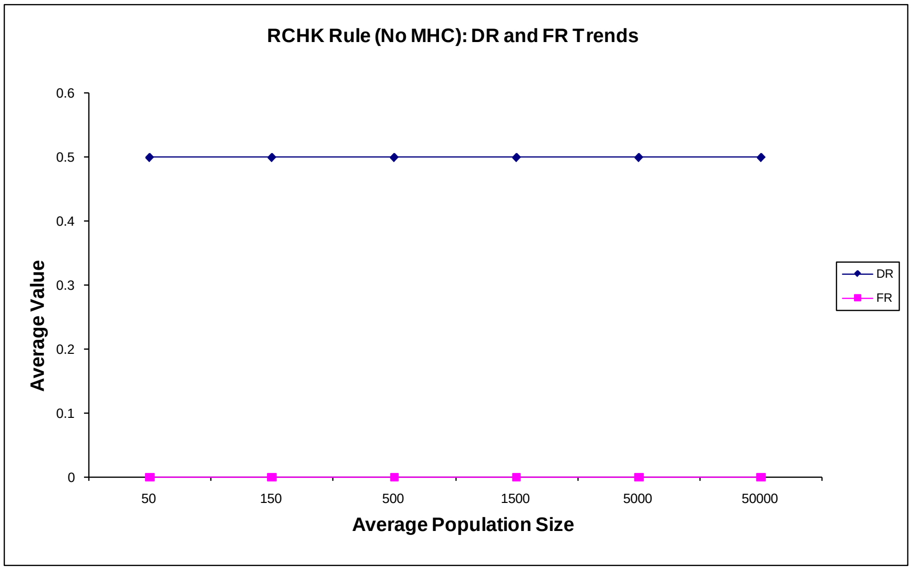


Figure 97. *Glass Data Set - RCHK (No MHC) Rule DR and FR Trends*

C.4.4 RCHK (Global MHC) Rule Trends

The GC and OC trends of the RCHK (Global MHC) rule for the best scenario illustrated in Figure 98 show that both GC and OC are constant at 0 as the average population size increases. The GC trend is bad because it illustrates that the generated detectors are not able to generalise across the data set whereas the OC trend is good because it illustrates that the generated detectors are distributed in manner such that their detection regions do not overlap.

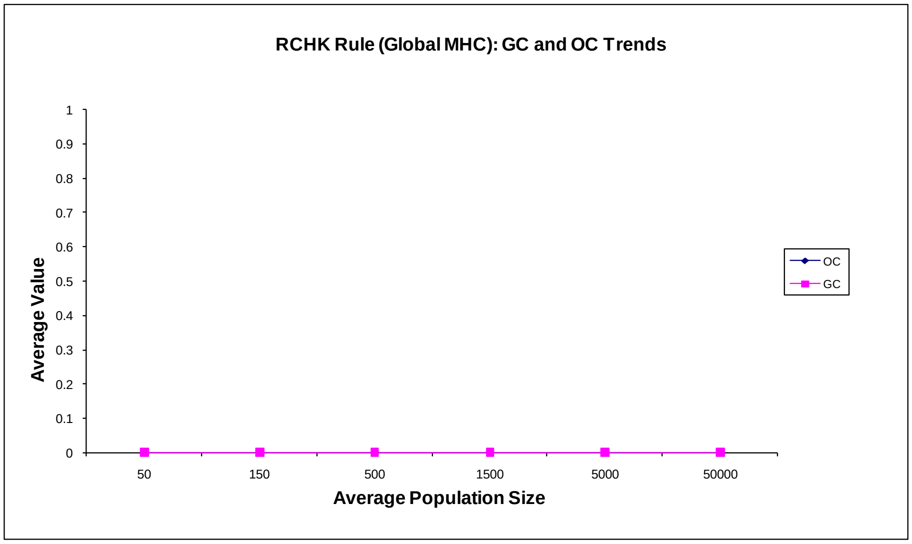


Figure 98. *Glass Data Set - RCHK (Global MHC) Rule GC and OC Trends*

The DR and FR trends of the RCHK (Global MHC) rule for the best scenario illustrated in Figure 99 show that:

- DR is constant at 0.5 as the average population size increases.
- FR is constant at 0 as the average population size increases.

The DR trend is bad because it illustrates that increasing the magnitude of the average population size does not improve the DR value. The FR trend is good because it is constant at 0 regardless of the magnitude of the average population size.

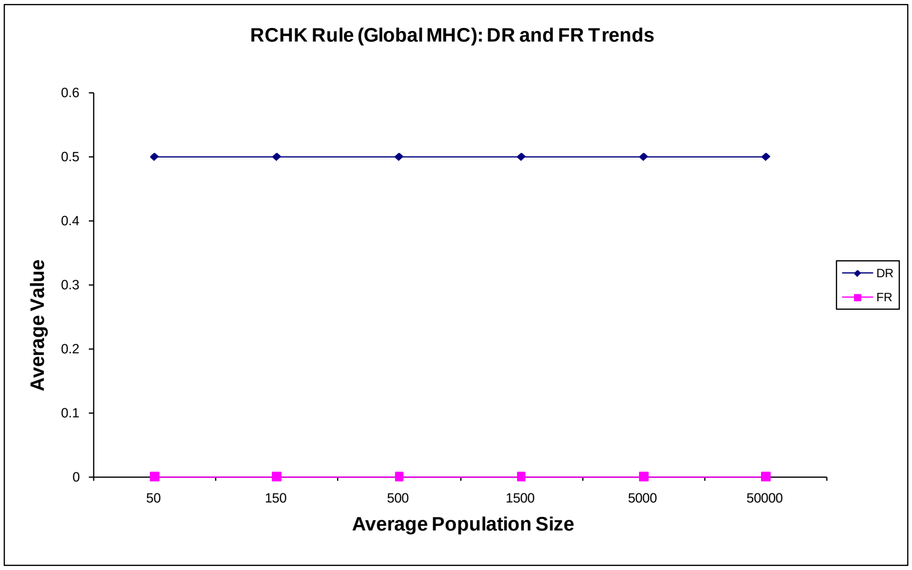


Figure 99. *Glass Data Set - RCHK (Global MHC) Rule DR and FR Trends*

C.4.5 RCHK (MHC) Rule Trends

The GC and OC trends of the RCHK Rule (MHC) for the best scenario illustrated in Figure 100 show that:

- GC is initially 0 as the average population size increases and suddenly increases to 2×10^6 at an average population size of 50000.
- OC is constantly 0 as the average population size increases.

The GC trend is bad because it illustrates that the generated detectors are not able to generalise across the data set whereas the OC trend is good because it illustrates that the generated detectors are distributed in manner such that their detection regions do not overlap.

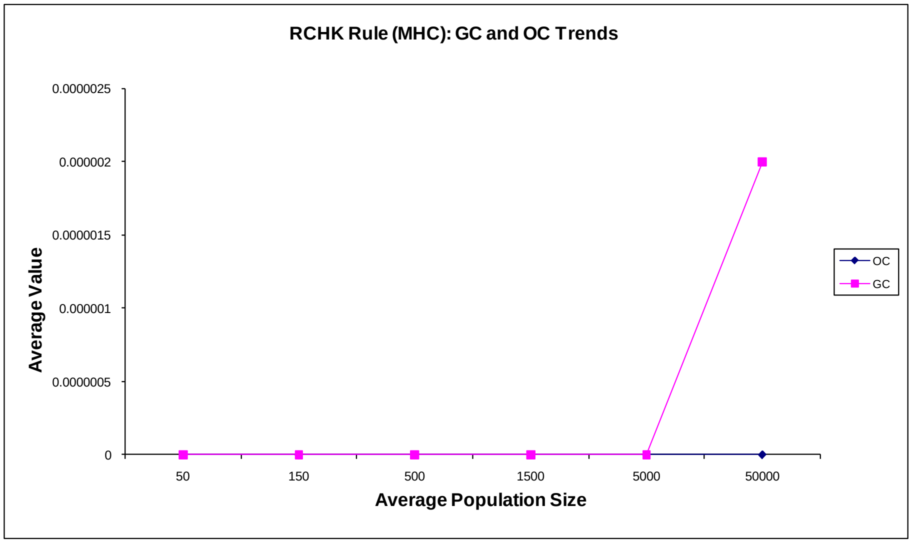


Figure 100. *Glass Data Set - RCHK (MHC) Rule GC and OC Trends*

The DR and FR trends of the RCHK (MHC) rule for the best scenario illustrated in Figure 101 show that:

- DR is constant at 0.5 as the average population size increases.
- FR is constant at 0 as the average population size increases.

The DR trend is bad because it illustrates that increasing the magnitude of the average population size does not improve the DR value. The FR trend is good because it is constant at 0 regardless of the magnitude of the average population size

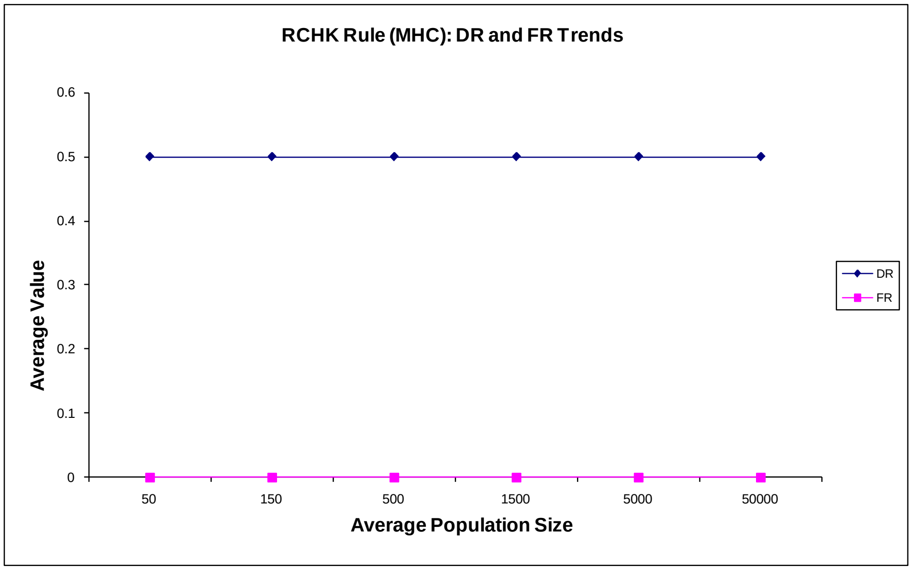


Figure 101. *Glass Data Set - RCHK (MHC) Rule DR and FR Trends*

C.4.6 RCHK (MHC) Rule vs. Feature-Detection Rule Trends

A comparison between the OC, GC, DR, FR and actual population size for the feature-detection rule where $r = n'$ and the RCHK (MHC) rule for the best scenarios is illustrated in Figure 102 to Figure 106 respectively. The figures show that the feature-detection rule has a lower OC and GC value than the RCHK (MHC) rule and that DR and FR values for the feature-detection rule and RCHK rule (MHC) are almost identical. Figure 106 shows that the feature-detection rule is always able to achieve the target population size, N_c, whereas the RCHK (MHC) rule is not always able to achieve the target population size.

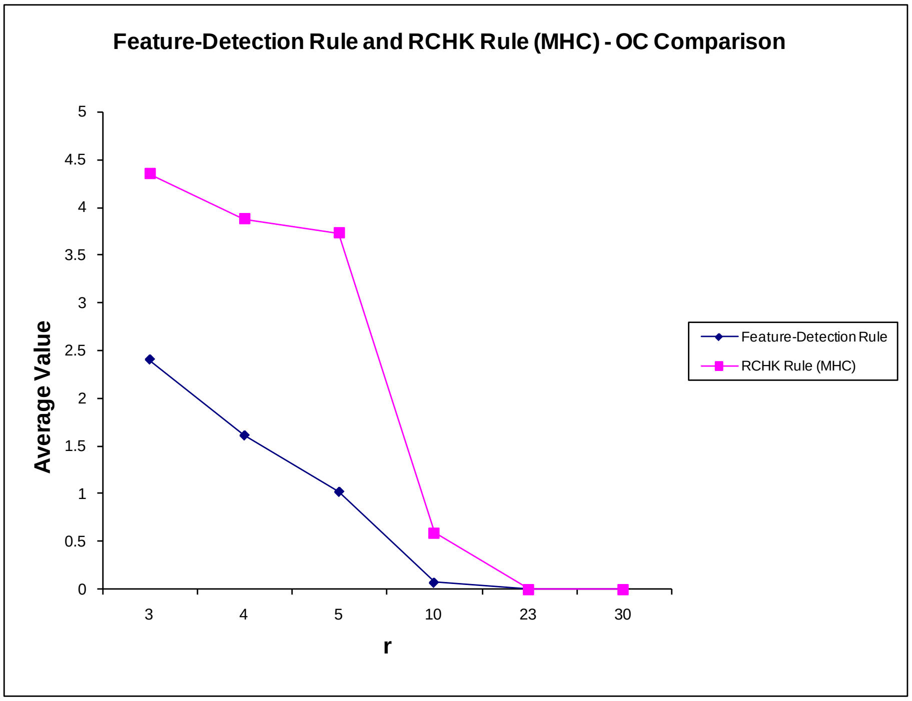


Figure 102. *Glass Data Set: Feature-Detection Rule and RCHK (MHC) Rule – OC Comparison*

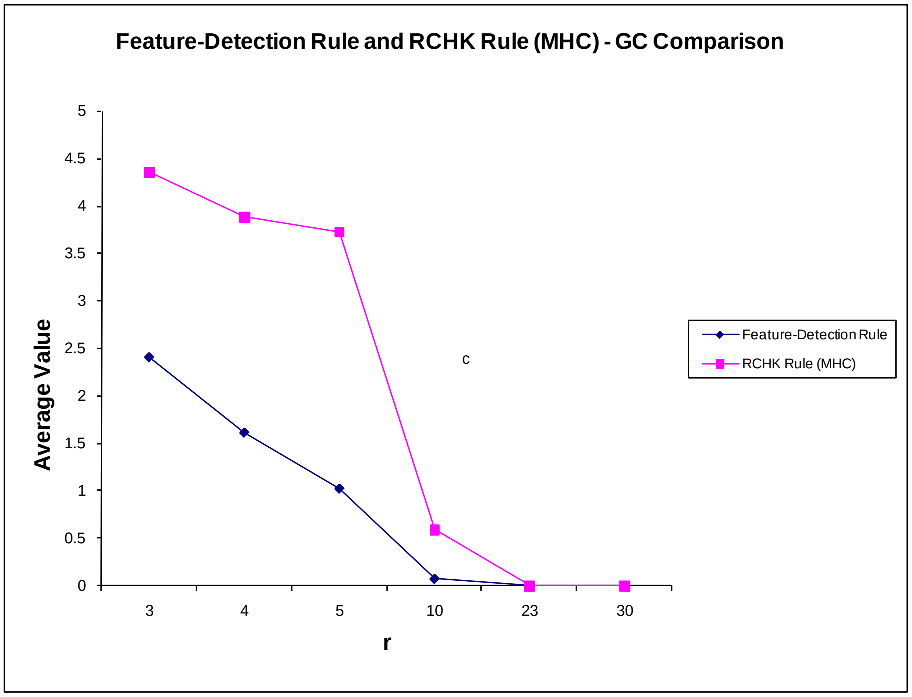


Figure 103. *Glass Data Set: Feature-Detection Rule and RCHK (MHC) Rule – GC Comparison*

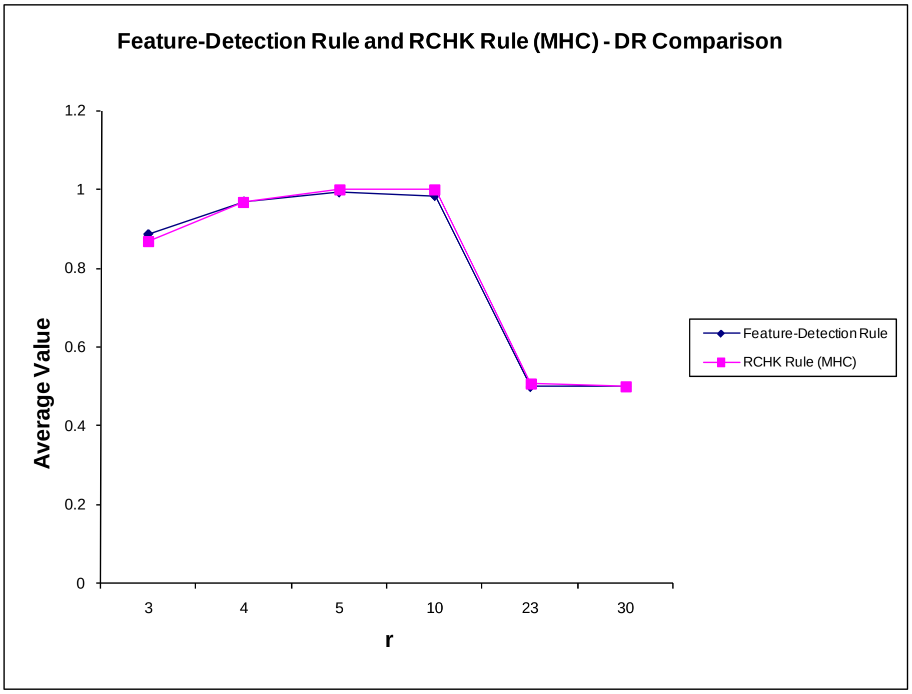


Figure 104. *Glass Data Set: Feature-Detection Rule and RCHK (MHC) Rule – DR Comparison*

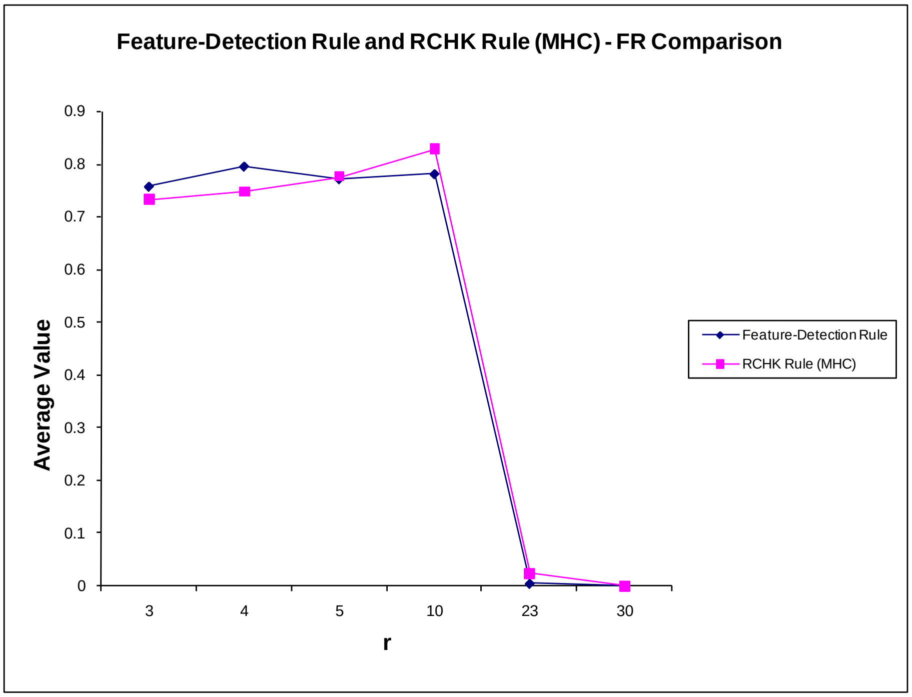


Figure 105. *Glass Data Set: Feature-Detection Rule and RCHK (MHC) Rule – FR Comparison*

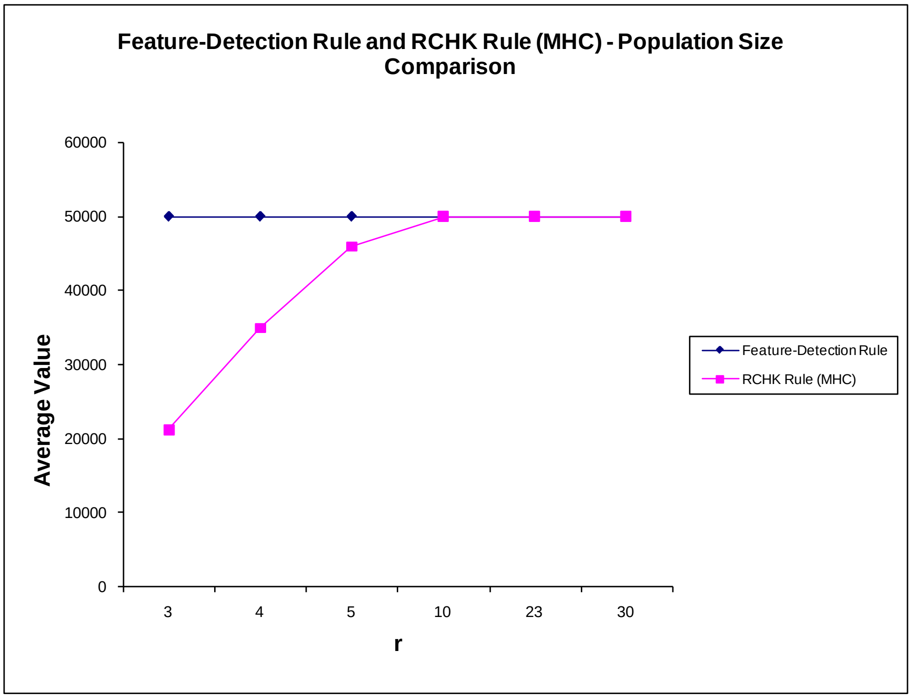


Figure 106. *Glass Data Set: Feature-Detection Rule and RCHK (MHC) Rule – Population Size Comparison*

C.5.0 Mushroom Data Set Trends

This section contains graphs that illustrate the OC, GC, DR, and FR trends observed for each scenario's best test groups for the Mushroom experiment. The section is concluded with graphs illustrating the OC, GC, DR, FR, and target population size, N_c, vs. actual average population size trends for the feature-detection rule and RCHK (MHC) rule test groups where $r = n'$.

C.5.1 Feature-Detection Rule Trends

The GC and OC trends of the feature-detection rule for the best scenario illustrated in Figure 107 show that:

- GC is initially equal to 110 at an average population size of 50 and then slowly decreases to 96 as the average population size increases.
- OC increases as the average population size increases until it is equal to GC. The OC trend is bad because it illustrates that the amount of overfitting increases as

the number of detectors are increased, i.e. the NSA is distributing detectors in a manner such that their detection regions overlap.

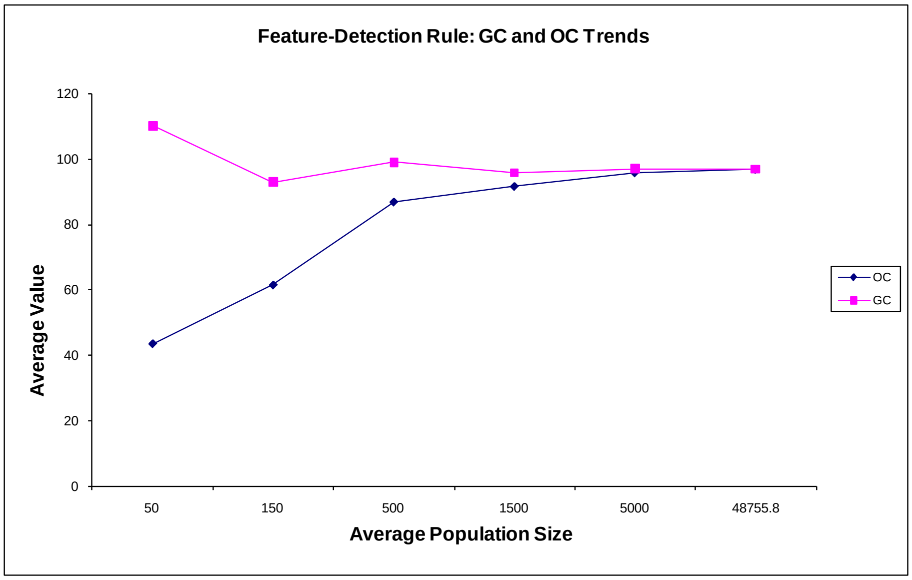


Figure 107. *Mushroom Data Set – Feature-Detection Rule GC and OC Trends*

The DR and FR trends of the feature-detection rule for the best scenario illustrated in Figure 108 show that:

- DR is initially 0.76 when the average population size is 50 and then increases to 1 as the average population size increases.
- FR is constant at 0 as the average population size increases.

The DR trend is good because it illustrates that DR approaches 1 as the magnitude of the average population size is increased. The FR trend is good because it illustrates that FR is constant at 0 regardless of the magnitude of the average population size.

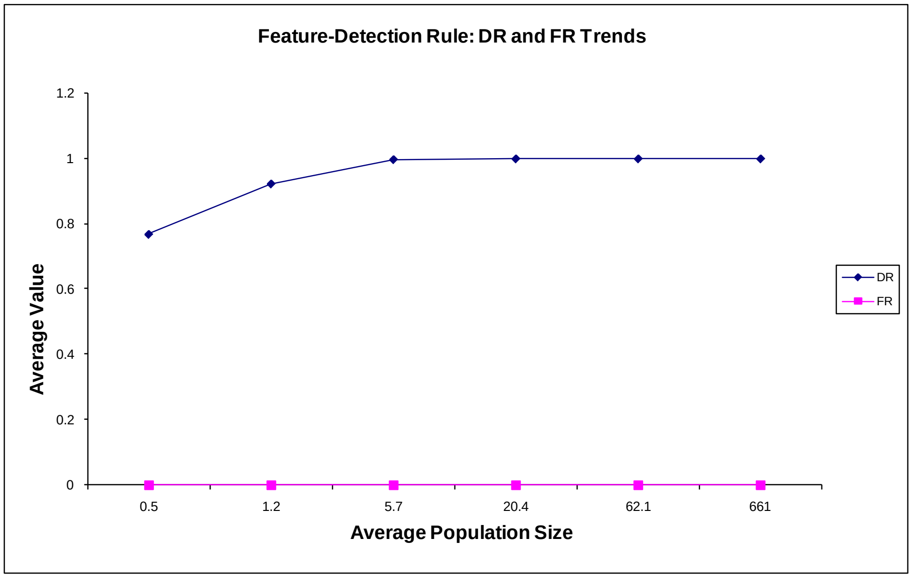


Figure 108. *Mushroom Data Set – Feature-Detection Rule DR and FR Trend*

C.5.2 HD Rule Trends

The GC and OC trends of the HD rule for the best scenario illustrated in Figure 109 show that:

- GC is initially equal to 62.9 at an average population size of 0.5, increases sharply to 156.94 at an average population size of 1.2, decreases sharply to 100.69 at an average population size of 5.7 and then slowly increases to 149.35 as the average population size increases. The GC trend is good because it indicates that the average generalisation exhibited by the generated detectors increases as the average population size increases.

- OC initially increases slowly as the average population size increases and starts increasing at a rapid rate at an average population size of 5.7 until it is almost equal to GC at a population size of 661. The OC trend is bad because it illustrates that the amount of overfitting increases as the number of detectors are increased, i.e. the NSA is distributing detectors in a manner such that their detection regions overlap.

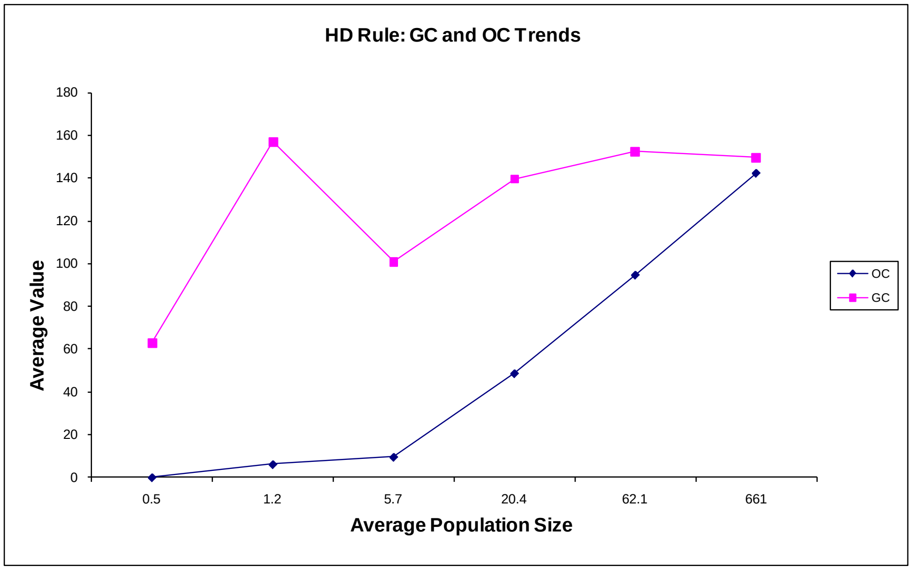


Figure 109. *Mushroom Data Set - HD Rule GC and OC Trends*

The DR and FR trends of the HD rule for the best scenario illustrated in Figure 110 show that:

- DR is initially constant at 0.5 and starts increasing at an average population size of 5.7 until it is equal to 0.99.
- FR is initially constant at 0 as the average population size increases and then starts increasing at an average population size of 62.1.

The DR trend is good because it illustrates that the DR value approaches one as the average population size increases. The FR trend is good because it illustrates that there is only a slight increase in the FR value as the average population size increases.

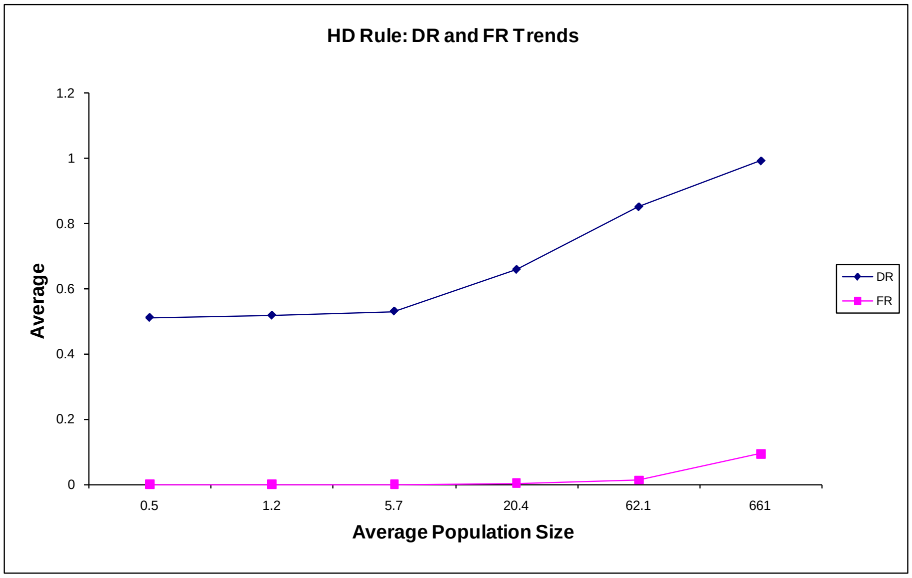


Figure 110. *Mushroom Data Set - HD Rule DR and FR Trends*

C.5.3 RCHK (No MHC) Rule Trends

The GC and OC trends of the RCHK (No MHC) rule for the best scenario illustrated in Figure 111 show that:

- GC is initially equal to 27.16 at an average population size of 18.7 and then slowly increases to 33.6 as the average population size increases. The GC trend is essentially equal to an average of 27 and the fluctuation in the GC value is caused by the randomness that occurs when executing each test within the test group.
- OC increases until it is equal to GC as the average population size increases. The OC trend is bad because it illustrates that the amount of overfitting increases as the number of detectors are increased, i.e. the NSA is distributing detectors in a manner such that their detection regions overlap.

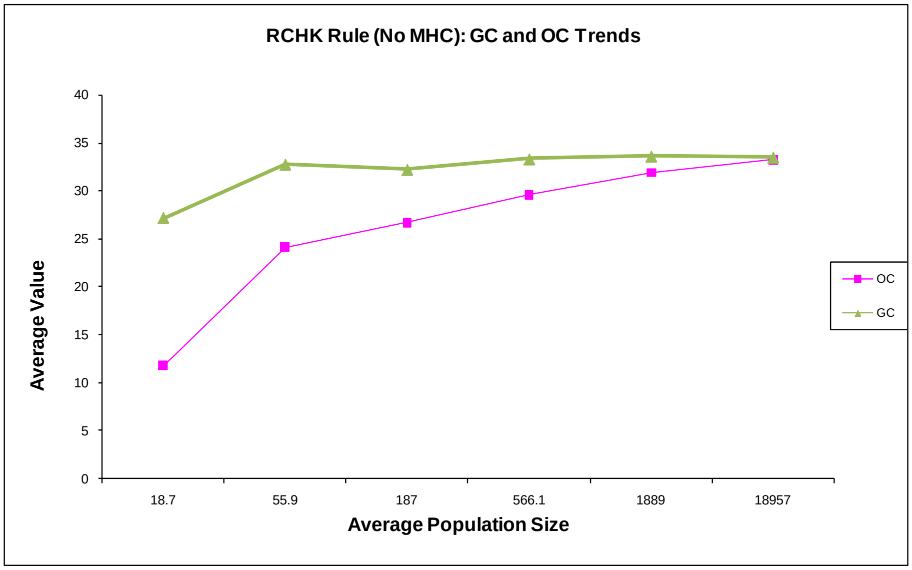


Figure 111. *Mushroom Data Set - RCHK (No MHC) Rule GC and OC Trends*

The DR and FR trends of the RCHK (No MHC) rule for the best scenario illustrated in Figure 112 show that:

- DR is initially constant and starts increasing at an average population size of 187 until it is equal to 1.
- FR is constant at 0 as the average population size increases.

The DR trend is good because it indicates that the average DR value approaches 1 as the average population size increases. The FR trend is good because it is constant at 0 as the population size increases.

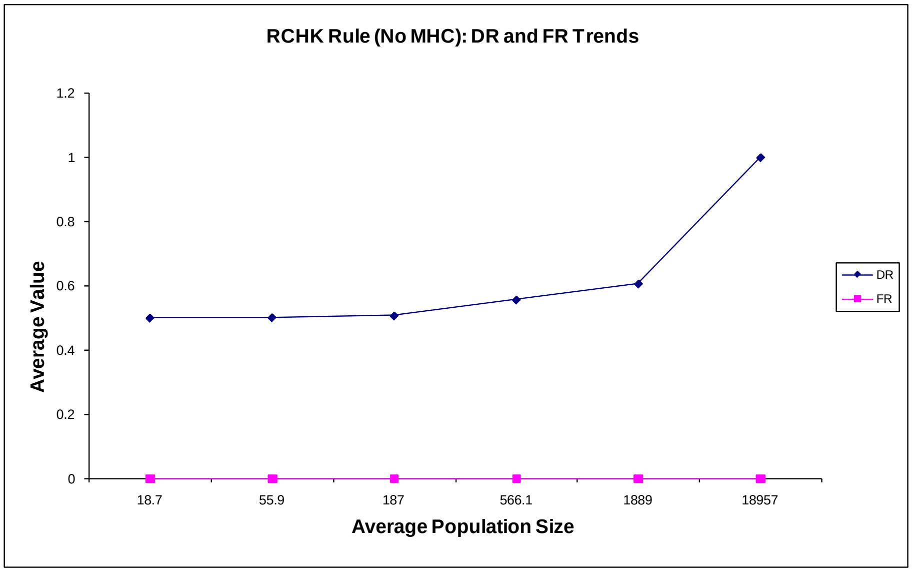


Figure 112. *Mushroom Data Set - RCHK (No MHC) Rule DR and FR Trends*

C.5.4 RCHK (Global MHC) Rule Trends

The GC and OC trends for the RCHK (Global MHC) rule for the best scenario illustrated in Figure 113 show that both GC and OC are 0 as the average population size increases. The GC trend is bad because it illustrates that the generated detectors are unable to generalise whereas the OC trend is good because it illustrates that the detectors are distributed in a manner such that their detection regions do not overlap.

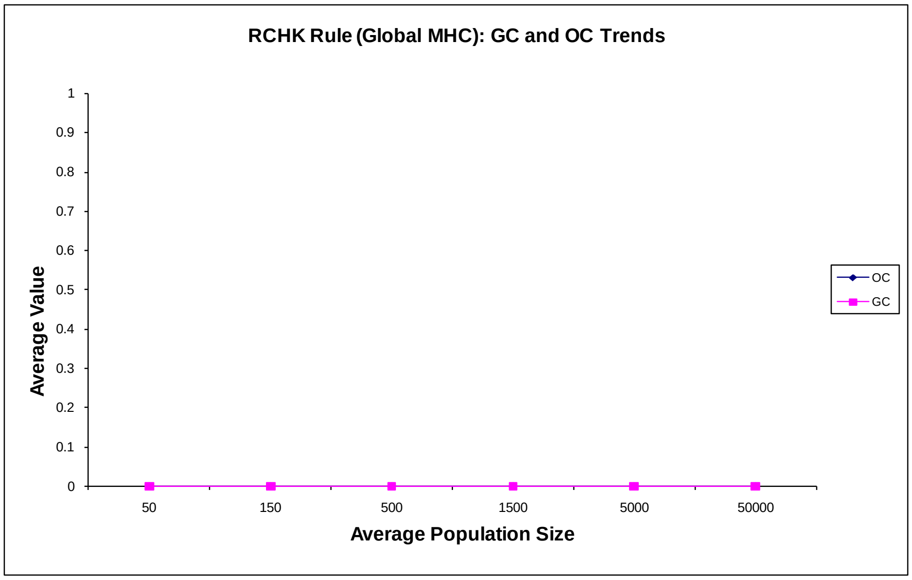


Figure 113. *Mushroom Data Set - RCHK (Global MHC) Rule GC and OC Trends*

The DR and FR trends of the RCHK (Global MHC) rule for the best scenario illustrated in Figure 114 show that:

- DR is constant at 0.5 as the average population size increases.
- FR is constant at 0 as the average population size increases.

The DR trend is bad because it illustrates that increasing the magnitude of the average population size does not increase the DR value. The FR trend is good because it is constant at 0 as the average population size increases.

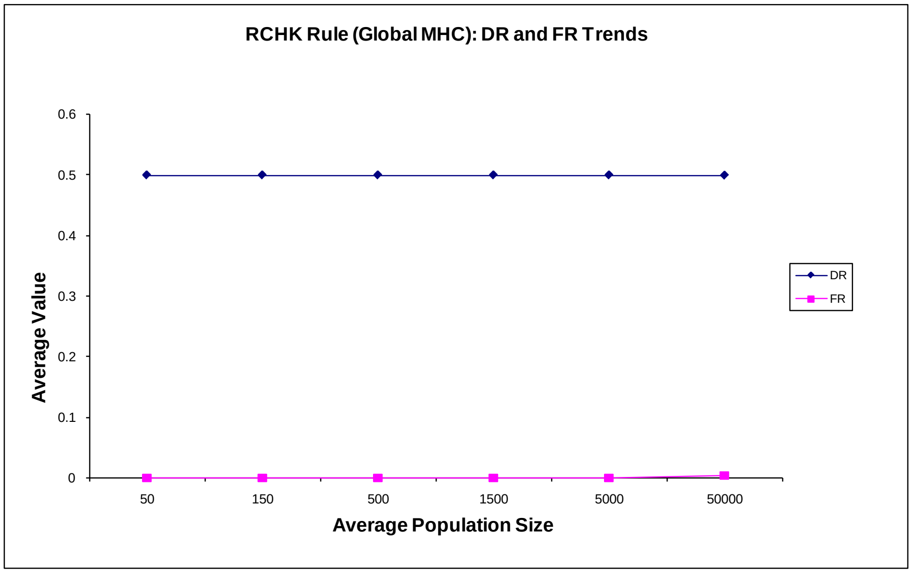


Figure 114. *Mushroom Data Set - RCHK (Global MHC) Rule DR and FR Trends*

C.5.5 RCHK (MHC) Rule Trends

The GC and OC trends of the RCHK (MHC) rule for the best scenario illustrated in Figure 115 show that:

- GC is initially equal to 89 at an average population size of 39.6, increases slightly as the average population size approaches 400.7 and then decreases to 100. The GC trend is essentially equal to an average of 89 and the fluctuation in the GC value is caused by the randomness that occurs when executing each test within the test group.

- OC increases until it is equal to GC as the average population size increases. The OC trend is bad because it illustrates that the amount of overfitting increases as the number of detectors are increased, i.e. the NSA is distributing detectors in a manner such that their detection regions overlap.

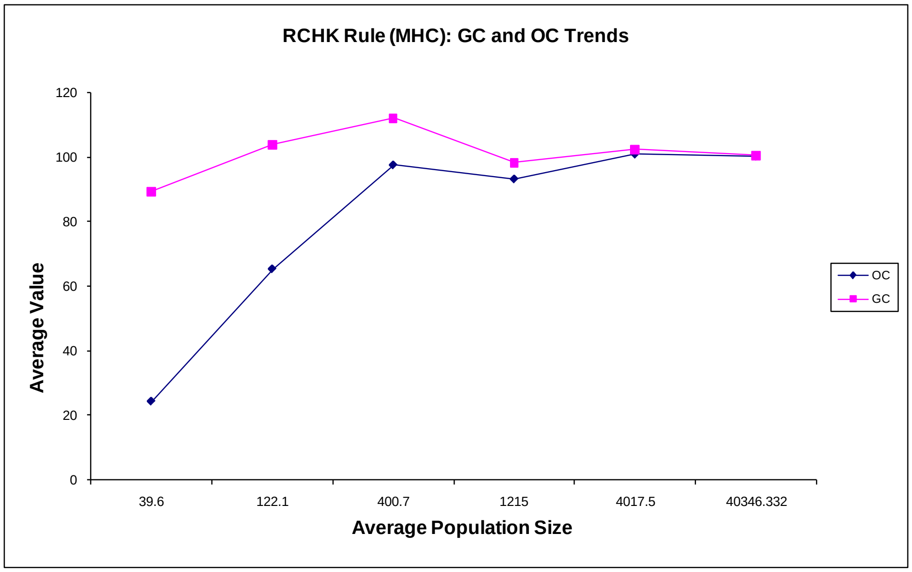


Figure 115. *Mushroom Data Set - RCHK (MHC) Rule GC and OC Trends*

The DR and FR trends of the RCHK (MHC) rule for the best scenario illustrated in Figure 115 show that:

- DR increases as the average population size increases until it is equal to 1.
- FR is constant at 0 as the average population size increases.

The DR trend illustrates that DR is maximised at an average population size of 400.7. The FR trend is good because it is constant at 0 as the population size increases.

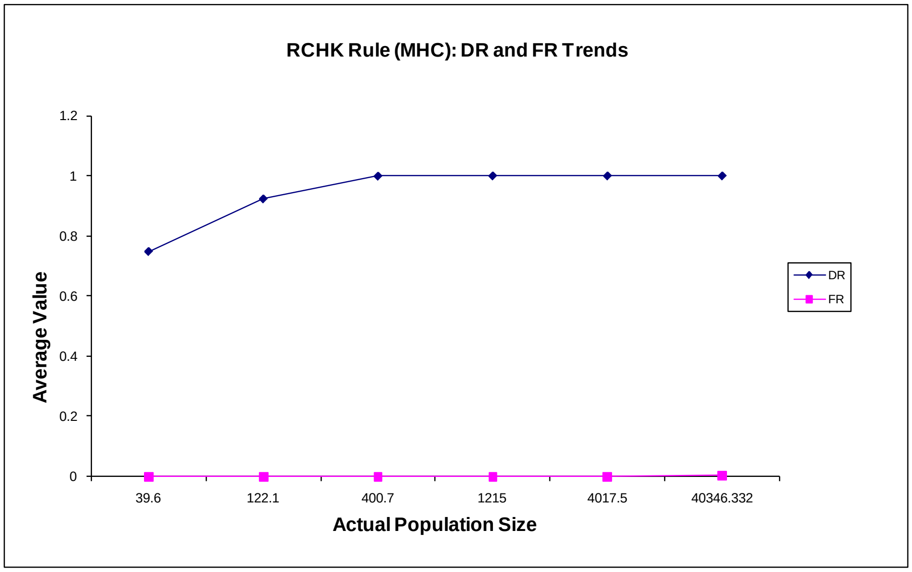


Figure 116. *Mushroom Data Set - RCHK (MHC Rule) DR and FR Trend*

C.5.6 RCHK (MHC) Rule vs. Feature Detection Rule Trends

A comparison between the OC, GC, DR, FR and actual population size for the feature-detection rule where $r = n'$ and the RCHK (MHC) rule for the best scenarios is presented in Figure 117 to Figure 121 respectively. The figures show that the feature-detection rule has lower OC, GC, FR and DR values than the RCHK (MHC) rule. Figure 121 shows that the feature-detection rule is always able to achieve the target population size, N_c, whereas the RCHK (MHC) rule is not always able to achieve the target population size.

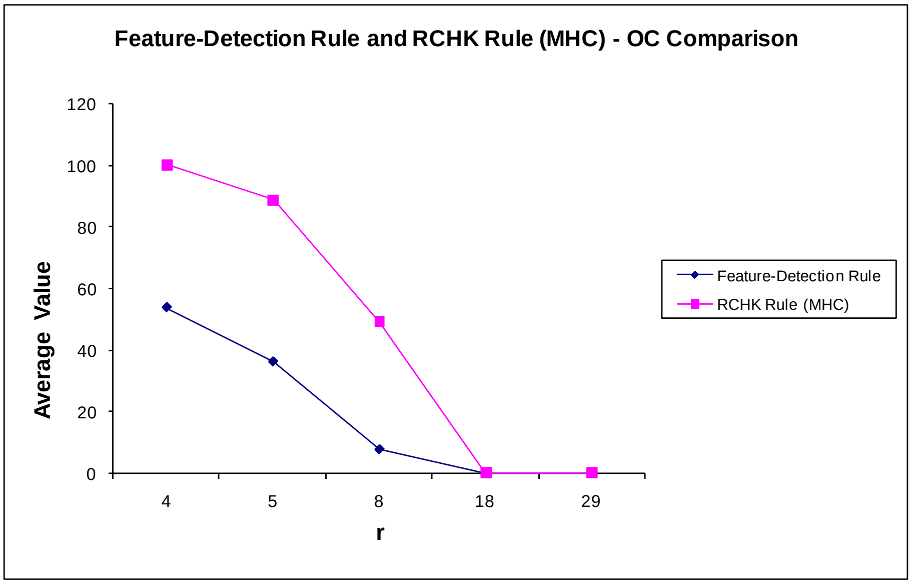


Figure 117. *Mushroom Data Set: Feature-Detection Rule and RCHK (MHC) Rule – OC Comparison*

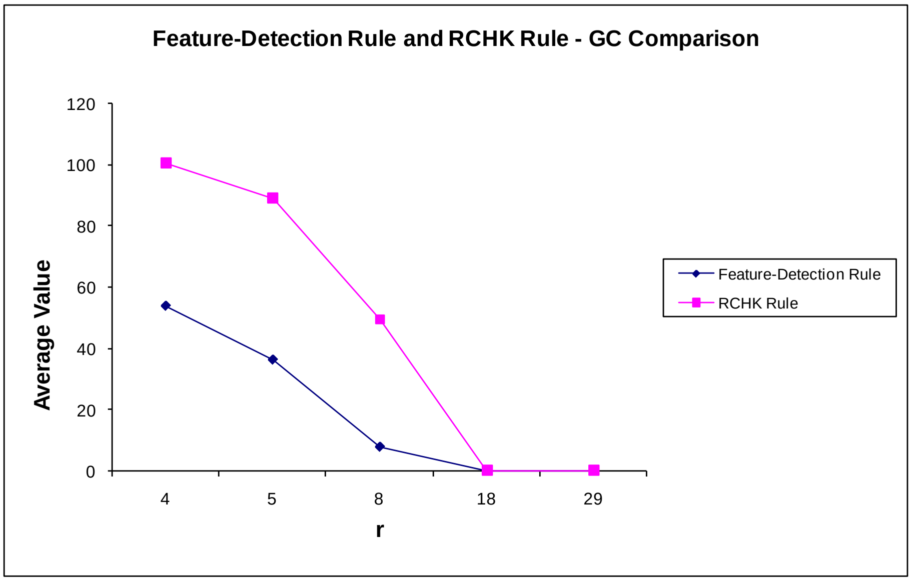


Figure 118. *Mushroom Data Set: Feature-Detection Rule and RCHK (MHC) Rule – GC Comparison*

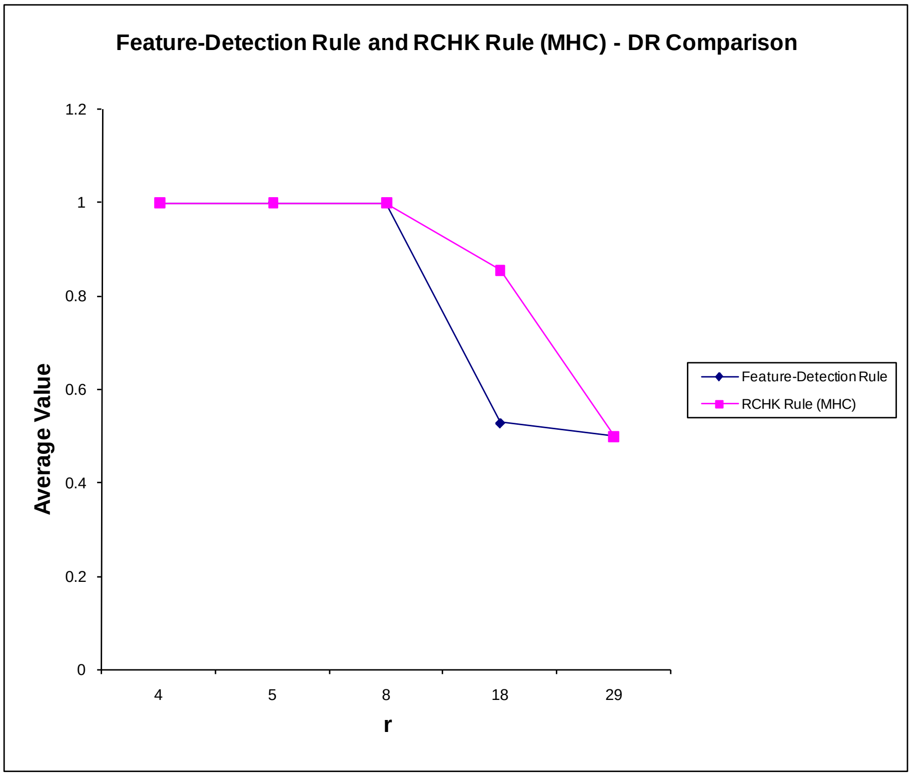


Figure 119. *Mushroom Data Set: Feature-Detection Rule and RCHK (MHC) Rule – DR Comparison*

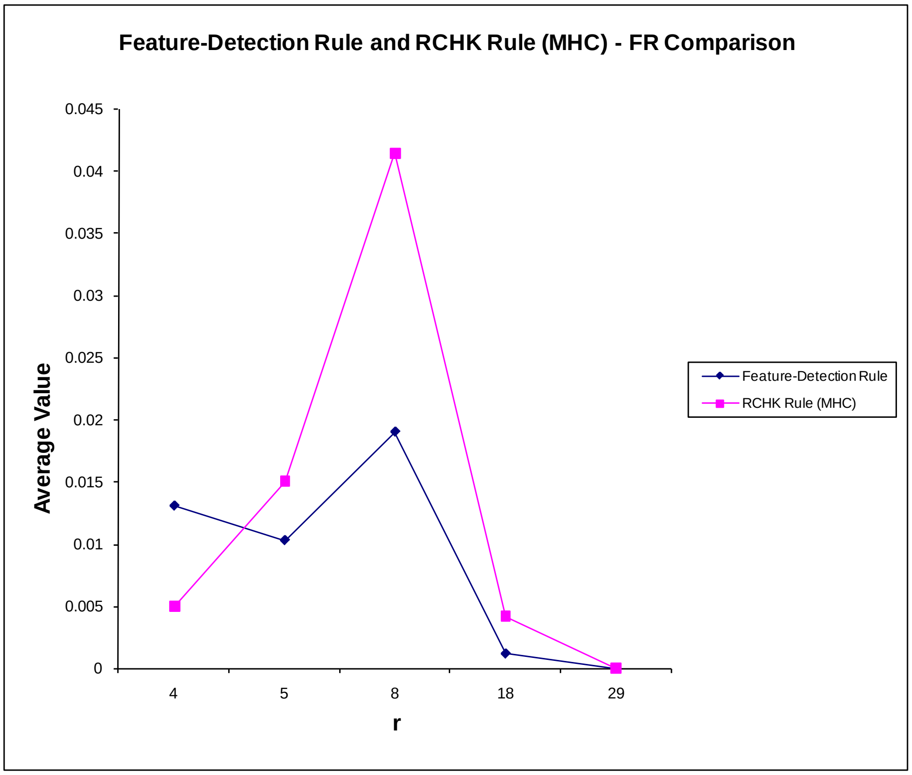


Figure 120. *Mushroom Data Set: Feature-Detection Rule and RCHK (MHC) Rule – FR Comparison*

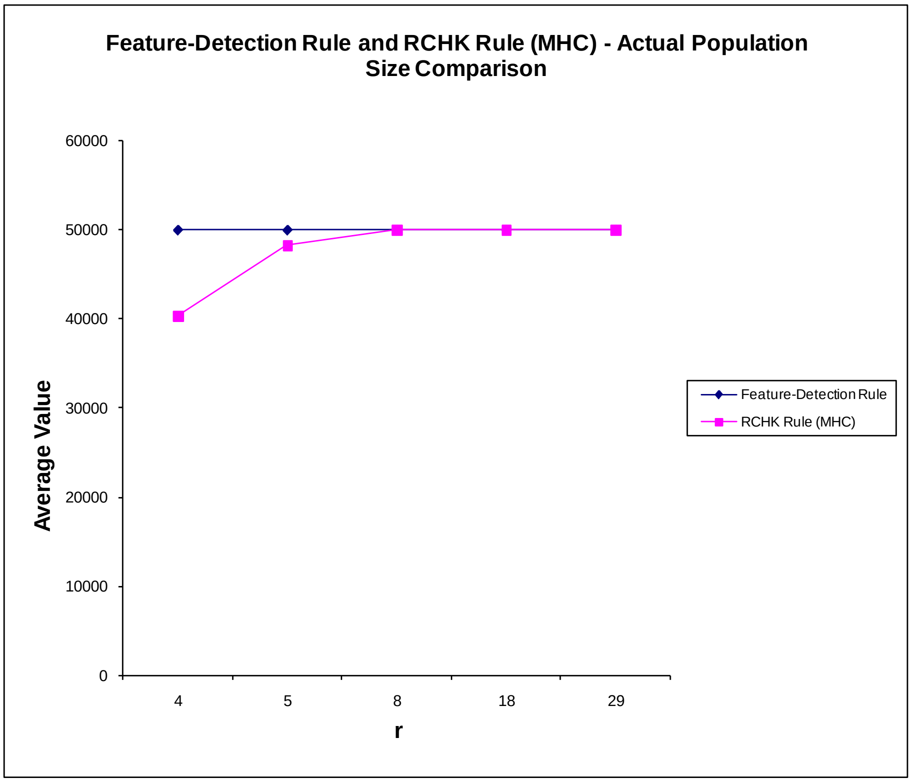


Figure 121. *Mushroom Data Set: Feature-Detection Rule and RCHK (MHC) Rule – Population Size Comparison*