ARTIFICIAL IMMUNE SYSTEMS FOR INFORMATION FILTERING: FOCUSING ON PROFILE ADAPTATION

Nurulhuda Firdaus Mohd Azmi

Doctor of Philosophy

University of York
Department of Computer Science

January 2014

Abstract

The human immune system has characteristics such as *self-organisation*, *robustness* and *adaptivity* that may be useful in the development of adaptive systems. One suitable application area for adaptive systems is Information Filtering (IF). Within the context of IF, learning and adapting user profiles is an important research area. In an individual profile, an IF system has to rely on the ability of the user profile to maintain a satisfactory level of filtering accuracy for as long as it is being used. This thesis explores a possible way to enable Artificial Immune Systems (AIS) to filter information in the context of profile adaptation. Previous work has investigated this issue from the perspective of self-organisation based on Autopoetic Theory. In contrast, this current work approaches the problem from the perspective of diversity inspired by the concept of dynamic clonal selection and gene library to maintain sufficient diversity. An immune-inspired IF for profile adaptation is proposed and developed. This algorithm is demonstrated to work in detecting relevant documents by using a single profile to recognize a user's interests and to adapt to changes in them. We employed a virtual user tested on a web document corpus to test the profile on learning of an emerging new topic of interest and forgetting uninteresting topics. The results clearly indicate the profile's ability to adapt to frequent variations and radical changes in user interest. This work has focused on textual information, but it may have the potential to be applied in other media such as audio and images in which adaptivity to dynamic environments is crucial. These are all interesting future directions in which this work might develop.

Contents

List of Tables

List of Figures

List of Algorithms

Acknowledgments

"In the name of Allah, Most Gracious, Most Merciful".
My highest gratitude to Allah the Almighty, by his blessing and permission I have completed the thesis. I also would like to acknowledge my appreciation to the following persons for their unconditional support.

To Prof. Jon Timmis and Dr. Fiona Polack for the supervision, ideas, guidance and support towards completing this research.

To my parents (Mohd Azmi Hj Mohamad and Masriah Hj Omar), my brothers, my sisters and my in-laws for all their support over the years and prayer.

To Jamil my dearest husband for the support and patience that made this thesis possible. His advice, understanding, prayer and endless love has help me to face a difficulty time.

To my loving children Nurin Hanani, Raina Nazneen and Nur Ayra Qairina; you have been your mother's true companion all along the research and during the writing of this thesis and you have given me so much strength and courage for me to finally complete the thesis.

To my friends, AIS Group and YCCSA Group for the ideas and support.

Declaration

The work in this thesis has been carried out by the author between March 2008 to August 2012 at the Department of Computer Science, University of York. Apart from work whose authors are clearly acknowledged, all other work presented in this thesis was carried out by the author. The results of this work have been previously published by the author. A complete list of refereed publications is as follows:

1. N. F. M. Azmi, J. Timmis, and F. Polack, "Profile adaptation in adaptive information filtering: An immune inspired approach," in IEEE Int. Conf. on Soft Computing and Pattern Recognition (SoCPaR), 2009, pp. 414-419.

2. N. F. M. Azmi, J. Timmis, and F. Polack, "Towards a principled design of bio-inspired solutions to adaptive information filtering," in 15th IEEE Int. Conf. on Engineering of Complex Computer Systems (ICECCS), 2010, pp. 315-316.

3. N. F. M. Azmi, F. Polack, and J. Timmis, "Immune inspired adaptive information filtering: Focusing on profile adaptation," in Bio-Inspired Models of Networks, Information and Computing Systems, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, 2011, vol. 103, pp. 242-247.

CHAPTER

ONE

INTRODUCTION

This chapter provides an overview of the research reported in this thesis. Section 1.1 presents the background of the related topics which motivated this work. This is followed by Section 1.2 which presents the challenges that encouraged the author to choose the topic of immune-inspired adaptive information filtering. Then in Section 1.3 the research question is formally stated. To investigate the research question, the research objectives are presented in Section 1.4. The chapter ends with an overview of the contents of this thesis presented in Section 1.5.

1.1 Motivation and Background

Information filtering (IF) is the process of filtering incoming data streams on the basis of a description (profile) of a single user, or a group of users, where the user reacts to the stream of information in such a way that only relevant data (information) is preserved. Typical examples include the automatic generation of a digital library (such as CiteSeer), web-based news filtering (such as NetNews articles), email filtering and advanced recommendation systems.

The process of identifying relevant information poses many difficulties, such as the uncertain nature of information requirements and the dynamics of data streams. Moreover, an IF system depends on its ability to recognise an individual's interests and adapt to changes in them; it depends on *profile*, a model of an individual's preferences. On the basis of an individual profile, an IF system needs to rely on the ability of the user profile to maintain satisfactory filtering accuracy for as long as it is being used. There are, however, issues of uncertainty due to rapid or gradual changes from the user and the document stream, so an IF system

requires a high level of adaptivity. The role of adaptive information processing systems in IF is known as adaptive information filtering (AIF). Tauritz [13] defined an AIF system as "a system that is capable of adapting to changes in both the data stream and the information needs". An AIF system has three main components: the data stream component, the filtering component, and the learning component [14]. In AIF, the relevance of the data is determined in accordance with the changing (adaptive) needs of a particular user [13, 15].

Nanas et al. [11, 16] argued that AIF is a characteristic example of a Multimodal Dynamic Optimisation (MDO) problem, in which the users' interests correspond to regions in a multi-dimensional information space. Studies by Nanas et al. [11, 16] argued that Evolutionary Algorithms (EAs) suffer from a lack of diversity when applied to radical changes, such as a new topic of interest. The issue of profile adaptation to information filtering poses an interesting and challenging research area. Clearly, the system must be both adaptable and robust in the sense that first, the content of the web is forever changing, second, the user's expectations need to be met despite the uncertain nature of interest, and finally, the system is robust to huge amounts of noise.

The natural immune system exhibits properties that are of interest to the area of information filtering [11, 17, 18]. Of particular interest is the ability of the natural immune system to maintain sufficient diversity, adaptability and self-organisation. The natural immune system provides the inspiration for a range of computer algorithms and applications; this is the field of artificial immune systems, AIS. Artificial Immune Systems (AIS) have been defined as "adaptive systems inspired by theoretical immunology and observed immune functions, principles and models" [4] and have the potential to support the criteria of adaptability and diversity. AIS encapsulate a number of desirable properties of the natural immune system including recognition, memory and self-organisation. Thus, given such interesting properties, the ability of AIS to solve problems in AIF is worthy of investigation.

The preceding paragraphs have expanded the title of this thesis; 'Artificial immune systems for information filtering: focusing on profile adaptation'. In IF, the ambiguity in representing a document, the uncertain nature of a user's information needs and the associated formulation process [19] contribute to the difficulties encountered in creating a user profile. A user profile should dynamically adapt to drifts in user interest and 'learn' with the changing interests. Inaccuracies in user profile will affect the quality of recommendation. The profile has to be able to represent the complete range of a user's interests and to continuously adapt, in response to user feedback to any changes in them, and the method by

which this goal may be achieved is in the use of AIS. This thesis is concerned with the design, implementation and evaluation of two AIS algorithms over sets of documents where each set has a notion of dynamics associated with it. In the first case, a passive filtering task is performed in which the system attempts to classify email into two classes: those which will be of interest to a user and those which will not, based on previous experience. This scenario is referred as 'binary classification' that discriminate between interesting and non-interesting emails. In the real-world situation, users are typically interested in more than one topic in parallel and both their interests and the information environment change over time. Therefore, users may read one or more email according to their interests. This scenario is referred to as 'multi-topic classification'. Therefore, to more accurately simulate a real situation, an extended experiment is performed in which each user profile has to represent more than one topic in parallel and adapt to both modest and radical variations in the topics. Thus it is necessary not only to investigate the performance of the algorithm in representing multiple topics in parallel, but also to demonstrate the ability of the algorithm to forget a previous topic when it starts to process a new topic.

The second task is to discover the ability of dynamic clonal selection to maintain a representation of the developing user interest in a changing information environment, particularly in a wider range of documents, for instance, the web content. While at first glance it would appear that both e-mail and web document are quite different, it could be argued that both are simply text documents but with specific meta data associated with them. E-mail has headers, subject and body while a web page contains hyperlinks. Both of these associate with an incoming information item with a dynamic information source. Thus, the main experimental part of the thesis takes the form of two case studies. Both serve to investigate AIS in domains are currently under explored reported in the literature, whilst also contributing to the IF field and particularly the adaptive document filtering domain. Both email and web pages often contain images, however, the image processing in the IF is out of the scope of the thesis.

Now that the subject of this thesis has been introduced, it is possible to discuss some of the challenges that need to be tackled in completing the thesis and state clearly the aim of this work (our research hypothesis and the research objectives), and then to outline the structure of this thesis.

1.2 Challenges

With regard to the aim of this thesis, there exist two particularly challenging areas. These are: 1) the challenges in adapting the user profile; and 2) the ability of a solution to perform well in a dynamic domain.

The state of previous research on using user profiling particularly in the fields of information retrieval (IR) and IF is not something new to this domain. Work on user profiling has fundamentally tackled the problem of Personalized Information Delivery (PID) in the field of IR and text categorisation (TC). Subsequently, IF has added a new dimension to the problem and pointed towards an alternative scientific direction, known as adaptive information filtering (AIF). A user profile or user model can be loosely described as a collection of assumptions or beliefs that the system holds about the user [3]. A user profile is not built once and applied unaltered thereafter. It is a long-term construct that has to continuously adapt to temporal changes in the user's interest. User interests change over time, driven by changes in the user's environment and knowledge. The user profile appears to adapt to a variety of changes ranging from frequent variations in a user's short-term needs, to occasional radical changes such as the emergence of a new topic of interest and the loss of interest in a particular topic. This results in the profile constantly changing structurally in response to changes in the stream of feedback on a user's document. Hence, the viability of an IF system relies on the ability of the user profile to maintain satisfactory filtering accuracy for as long as it is being used. The user profile has to be able to represent the complete range of a user's interests and to continuously adapt, in response to user feedback, to any changes in them. Inaccuracies in user profile affect the quality of recommendation. The ambiguity in representing a document and the uncertain nature of a user's information needs [15] contribute to the difficulties encountered in creating a user profile. These challenges of profile adaptation to information filtering raise a challenging research area.

Finding the information on the basis of a user's interest within a document or text in a data stream poses another challenge. A page of text does not fit neatly into a template, unlike rule-based classification or association, in which all rules have the same form: "IF(x) AND(y)....THEN(z)" [20]. Text is much more unstructured and tends to contain much more noise and irrelevant information than a structured dataset created by a conventional algorithm.

Classification in a dynamic scenario is also a challenge. In the scenario investigated, the web document source is dynamic. In this dynamic scenario, the previously unseen data should be assigned changes. This is a challenge because

it requires classification algorithms to have extra layers of complexity. The algorithm must constantly update its internal representation of the class distribution and it must do this in a robust way so that it does not start making mistakes. However, it is believed that an AIS may already offer hope in this area. Its internal representation is already dynamic, for instance the clonal selection algorithm and the immune network algorithm, therefore, it is thought that it may naturally lend itself to this scenario.

1.3 Research Questions

The context of this thesis is artificial immune systems for information filtering with a particular focus on the problem of profile adaptation. We developed an immune-inspired IF system that uses a single profile to recognize a user's interests and adapt to changes in them. As well as addressing this problem from the perspective of a single topic of interest, this research will also approach it from the perspective of many topics of interest. Therefore, the research question is defined as follows:

> *Can an AIS algorithm be developed that derives and maintains diversity to manage adaptation of profile on both short-term and long-term changes?*

From this main research question, some subsidiary questions were identified in order to answer the main research question. These are as follows:

1. What are the immunological properties that can address the problem of profile adaptation?
2. What is an effective mechanism to use for the representation of profile and information items?
3. How can we build an AIS that has the ability to adapt to changes in a user profile given multiple interests and radical changes in interests?
4. What is the baseline on which the performance of the AIS should be evaluated?

1.4 Research Objectives

As stated in Section 1.3, this research study will investigate the research question in the context of AIS in information filtering for profile adaptation. Therefore, the research objectives of the work in this thesis are:

- **RO1:** To identify an immunological principle and the AIF properties based on the principled approach and to work out how to instantiate those properties in the context of the profile adaptation problem.
- **RO2:** To establish an experimental testbed in the email environment in order to investigate the performance of AIS algorithms in classifying emails on the basis of the user's interest with regard to single and multiple email topics.
- **RO3:** To develop an AIS algorithm which incorporates adaptivity to adapt to changes in a user profile given multiple interests and radical changes in interest on adaptive document filtering.

1.5 Thesis Structure

The structure of this thesis is as follows. Chapter 1 sets the scene for the topic of artificial immune system for information filtering. It illustrates current knowledge and relevant understanding of the field. In this chapter, the motivation and the challenges which encouraged the choice of the topic and the development of the study are presented. The research question and the research objectives for this work are also presented in this chapter. This chapter ends by describing the structure of the thesis and the most significant contents of the succeeding chapters.

Chapter 2 and Chapter 3 primarily explore and explain the background of the research. There is a literature review which serves two purposes. First, it serves to impart technical knowledge to the reader to allow full comprehension of the later chapters. Second, it serves to provide evidence to justify the use of AIS and the chosen problem domain in the context presented in this thesis. As this thesis covers two topics, information filtering and artificial immune systems, the subjects are separated for the sake of clarity. For this reason, Chapter 3 does not follow in terms of content from Chapter 2.

Chapter 2 begins with an introduction of IF concepts including the basic architecture which underpins IF systems, and the principle view of IF. Readers will be introduced to the concept of adaptation in IF, known as AIF. Some of the challenges in AIF will be discussed in this chapter. An IF system deals with the problem of adapting the user profile to changes in user interest. There is an explanation of the inspiration of profile adaptation and why it is the major concern of the thesis. This includes the challenges of user profiling. To provide a better overview of adaptive user profiles, some of the existing research is reviewed in

this chapter. The purpose is to determine how these approaches have tackled the problem of adaptation. Having noted the existing applications which emphasize adaptive user profiling, the chapter also gives an overview of various existing algorithms for a learning profile. A review of the existing profile adaptation algorithm is undertaken based on the approaches through a learning algorithm, an evolutionary algorithm and a connectionist algorithm.

Chapter 3 presents a literature review covering the areas of AIS. In this chapter, a wider appreciation of the context of AIS is given, including the biological context of the immune system, the immune principle and the computational perspective of AIS, including its framework and the AIS application. From the principled approach introduced by Stepney et al. [5] known as ODISS (Openness, Diversity, Interaction, Structure, Scale), this chapter presents our principled design of high-level abstraction which is used as part of the basis for building a biologically-inspired application: in this case, an immune-inspired IF application. An analysis based on ODISS is used to identify aspects which can address the desirable characteristics of the immune system and the application area of profile adaptation in IF. This principled design leads to a comprehensive set of requirements which makes it possible to identify an appropriate property of the immune system and the problem domain which is studied here. This addresses **RO1**. From the principled design, it is suggested that profile adaptation can be developed by incorporating ideas from aspects of dynamic clonal selection (DCS) with the evolution of gene libraries to maintain sufficient diversity. Therefore, in this chapter the concepts of clonal selection, gene libraries and DCS are also presented. Finally, this chapter ends with a consideration of some issues which must be taken into account in the design of the dynamic clonal selection algorithm for profile adaptation.

From the intensive literature reviews in Chapter 2 and Chapter 3, readers will not only become familiar with the background area of the immune system as well as the IF domain but will also see how the principled design can be helpful in the construction of a bio-inspired algorithm. Readers will be shown how the principled design of bio-inspired solutions to complex problems based on ODISS can help to analyse and evaluate the target application area, in the same terms as the investigation into the natural systems from which the inspiration for the study came.

Chapter 4 describes AIS algorithms in text-mining scenarios, particularly in the classification of emails. In the first part of the chapter, an existing immune-inspired e-mail classification algorithm known as AISEC [9] will be introduced. This includes the process and the working extension of the algorithm. After this

review, an extended version of AISEC is developed which focuses on increasing the diversity of words in the gene library with a large library of words and the ability to detect synonyms, and this is discussed in detail. To show that the extended AISEC is capable of continuous learning, and potentially of tracking changes in email topics, an experiment was conducted to verify whether explicit changes in a user's interests could be tracked, and this experiment is described. The email experiment was carried out in two types of scenario; binary classification (discriminating between interesting or non-interesting email) and multiple-topic classification. The protocol of the experiment and its analysis is discussed in this chapter.

To further evaluate the extended version of AISEC, Chapter 5 presents a sensitivity analysis of its parameters. This is because the dynamic behaviour of an algorithm can be controlled by the algorithm's parameters, of which there are many. Therefore, there is a need to examine the influence of the algorithm's parameters on the performance of the algorithm. This chapter focuses on the analysis of the extended AISEC parameters by investigating the effects on the *FPR* (False Positive Rate), *FNR* (False Negative Rate), *predictive accuracy* and the *A* value from the Vargha-Delaney *A* statistics. The implementation of the sensitivity analysis discussed in this chapter is divided into two scenarios; namely, the single-topic classification (more precisely refer to binary classification problem) and the multiple-topic classification. Chapter 4 and Chapter 5 address **RO2** of this research.

Chapter 6 presents the work for **RO3**. This chapter is concerned with turning AIS towards web-content mining with a specification to discover an adaptation for profile on adaptive document filtering. The algorithm is inspired by the dynamic clonal selection with gene library to maintain a sufficient diversity presented in Chapter 2. In the first part of Chapter 6, some motivational work is presented including the inspiration of dynamic clonal selection and its potential for adapting a user profile of interest. Later, the algorithm and the process of DCS known as profile adaptation through dynamic clonal selection (ProAd-DCS) is presented, in order to give a better context of the developed system. The goal is to adapt our multi-topic profile both to short-term variations in the user's need and to progressive, but potentially radical changes in long-term interests. A more formal pseudo code and a lower-level description of the algorithm will be explained to aid implementation of the study. An evaluation of the algorithm which is based on a simulated approach is also presented in this chapter. This work addresses the research objective **RO3**. At the end of the chapter, a comparative experiment will also be discussed. This involves a similar IF system which

focused on adapting a user profile, for example the Nootropia system, a user profiling model based on a self-organising term network influenced by the Autopoetic theory [21] and Rocchio's learning algorithm — an algorithm for learning user interests that has been well studied in information retrieval (IR) [22, 23]. Comparative performance is important in order to assess whether the works are incremental improvements on the state of the art or evolutions of existing work.

Finally, Chapter 7 concludes the thesis. A summary of the work presented in this thesis and its limitations, as well as suggestions for future work, are presented.

CHAPTER

TWO

INFORMATION FILTERING

This chapter gives an overview of information filtering (IF) topics. The chapter begins with the general context of IF which includes its history – when and why IF emerged. Later, it covers IF characteristics, techniques and processes. This chapter also focuses on adaptive information filtering (AIF) as the key to dynamic filtering. Several factors that contribute challenges of adaptive filtering are briefly discussed, including the user profile representation and its implication in AIF. A detailed review of current profile adaptation work which led to this research is also presented in this chapter.

2.1 The general context of Information Filtering Systems

A tremendous amount of information is created and delivered over electronic media. This explosive growth in information has fed the growth in the number of information resources available over the networks. The number of networked users has increased rapidly, with the widespread proliferation of computers and networks. As more and more users are getting on-line, it is increasingly difficult to find information unless one knows exactly where to get it and how to get it. Users risk becoming overwhelmed by the flow of information. In dealing with web information overload, classical methodologies or techniques from IF have been applied with various degree of success [24,25]. An IF system is an approach that delivers the relevant information to the user, omitting the non-relevant information. The basic architecture underpinning the IF systems relies on a represen-

tation, called the *profile*, of a user's interest. Some researchers in IF have referred to an IF system as a 'Personalized System' [26] or 'Personalized Information Delivery' [27]. Although different names are given, both systems tackle the problem of information overload, and the user profile is fundamental for automating the information processes.

A universally-accepted definition of IF, unfortunately, is still lacking.[1]. However, various definitions of IF have been proposed in the literature. Belkin and Croft [28] defined IF as:

> "the process of determining which profiles have a high probability of being satisfied by a particular object from the incoming stream".

Oard[1] defined IF as:

> "a system that sorts information through large volumes of dynamically generated information and presents to the user those which are likely to satisfy his or her information requirement".

A similar definition was proposed by Tauritz [13] who defined IF as:

> "a process of filtering data streams in such a way that only particular data are preserved, depending on certain information needs".

Höfferer et al. [29] believed that IF systems are not just restricted to assisting users by filtering the data stream and delivering the relevant information to the user, but are also used to target information to potentially interested user. Further, Hanani et al. [3] classified IF according to four parameters, namely:

1. Initiative of operation — concerned with how the filtering operation is initiated;
2. Location of operation — describing the possible location of the filtering process;
3. Filtering approach — distinguishing types of filtering techniques;
4. Acquiring knowledge of users — describing methods to acquire knowledge about users

The parameters described above do not reflect a classification of IF but present a 'point of view' of IF in general (see Figure 2.1). The description of each parameter is discussed in the following section.

[1] http://terpconnect.umd.edu/~dlrg/filter/

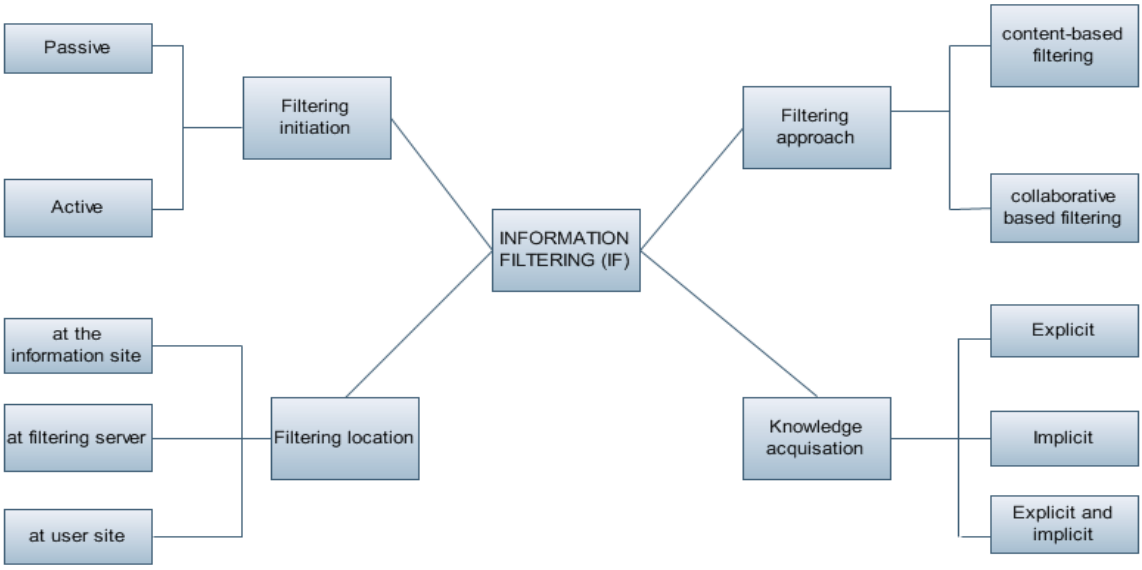


Figure 2.1: Classification of Information Filtering, [3]

2.1.1 Who initiates the Information Filtering Operation?

From Figure 2.1, it can be seen that generally IF operations are initiated either passively or actively. Hanani et al. [3] defined passive IF as a system that omits the irrelevant information from incoming streams of data items, with no effort to collect the data items for the users, while active IF systems are systems that actively seek relevant information for users. Examples of passive filtering are e-mail filtering (for example, spamming), automatic generators for digital libraries such as CiteSeer[2] and Usenet news, which is a world-wide distributed Internet discussion system. Examples of commercial active IF systems are Amazon, Musicmatch, eBay and so on . The distinction between passive and active filtering might be sensible as some of the filtering systems filter out irrelevant data items, while others provide the user with all available data items ranked and ordered according to their relevance. However, some IF systems, for example, the Cognitive Information Filtering System (CIFS) [29] operates both passively and actively at the same time in order to enhance the filtering effectiveness.

2.1.2 Where Does Information Filtering Operate?

The filtering process is generally performed in one of three locations:

1. at the information source — a distinction can be made between *dynamic* and *static* information sources, based on the lifetime of the information

[2]http://citeseer.ist.psu.edu/citeseer.html

items [30]. A dynamic information source can be thought of as a broadcaster while a static information source contains information with a longer lifetime, and maintains the information items for future access, as in, for example, a digital library or a news archive. At this type of location, a user posts a profile to an information provider and in return, the user is supplied with the information that matches the profile.

2. at the filtering server — some filtering system are implemented at a server, for example SIFT (Stanford University Information Filtering System) [31], and SIFTER (Smart Information Filtering for Electronic Resources) [15]. The filtering server implements active filtering, as the user posts the profile to servers. On the other hand, the information provider sends data items to these servers and the servers filter the data items and distribute relevant items to respective users.

3. at the user site — each incoming stream of data items at the user site is evaluated by a local filtering system, which removes the irrelevant items or rank-orders items according to their relevance [3]. Filtering at the user site is often integrated within an e-mail reader. Another example is the Usenet. This kind of approach implements passive filtering.

2.1.3 What are the techniques for Information Filtering?

The distinction between filtering approaches is broadly cited [3], [24, 29, 32] and widely used in IF research. The term 'content-based' was introduced by [32]. These authors referred content-based filtering as *cognitive filtering* which works by "characterizing the contents of the message and the information needs of potential message recipients and then using these representations to intelligently match messages to recipients" [32]. Some researchers are in agreement with this definition [3, 26, 33]. Other researchers, for example H*ö*fferer et al. [29] defined cognitive filtering from a physiological point of view, where the user model represents the complete user's cognitive style and personality factors, goals and plans, capabilities and preferences, and beliefs and knowledge.

'Sociology filtering' was defined by Malone et al. [32] as filtering that works by supporting the personal and organizational interrelationship of individuals in a community. Some researchers [24, 26, 33] defined sociology filtering as the same as 'collaborative filtering', in which people collaborate to help one another to perform filtering by recording their reactions to documents they read and con-

tribute to human recommendation. Other researchers, for example Hanani et al. [3] believed that sociology and collaborative filtering share characteristics with *properties-based filtering* which they defined as, 'filtering that is based on individual properties that include more than one areas of interest.

From these terms, in general, two types of IF approach can be identified: content-based filtering and collaborative filtering. These approaches differ in terms of user profile and how information items are represented and compared. For content-based filtering, user profile and information items are both represented using features that are extracted from the actual content of information items. Typically, content-based filtering is mainly concentrated on documents rather than audio or visual information. This is because the extraction of features such as 'term' is straightforward in the case of document text, but far less obvious for audio and visual information [11].

In collaborative filtering, information items are characterized by the ratings which they receive from users in a community. The user profile comprises the user's rating on information items. The goal of collaborative filtering is to recommend new items for the user (who is usually referred to as an *active user*). Recommendation of items is done by finding a set of users (usually referred as *neighbours*), which share the same interest as active user. In order to determine similarity, a measure of similarity between users is required. Once the 'set of neighbours' is established, recommendation can be made based on the neighbours' ratings.

Unlike content-based filtering, collaborative filtering does not require access to the actual content of information items [24], so, it has been successfully applied to various types of media for example, images, movies and audio. Recently, Yanga and Kin Fun Li [34] classified the collaborative filtering application as memory-based and model-based. Memory-based filtering can be further categorized either as user-based or item-based. Hybrid approach of model-based and memory-based filtering, called the personality diagnosis model, appears in [35]. Further explanations of the memory-based and model-based approaches of collaborative filtering can be found in [34] and [35]. Hanani et al. [3] argued that collaborative filtering is unlikely to provide a complete solution to filtering needs because each user's area of interest always plays a major role in determining the relevance of information. Therefore, they suggested that a combination of collaborative filtering with content-based filtering can boost the filtering result [3].

2.1.4 How to Acquire Knowledge about a User?

In general, the methods for acquiring knowledge about users can be categorized as an *explicit* approach, an *implicit* approach, and a mixed approach of explicit and implicit [3].

1. Explicit: User interrogation is a popular explicit technique for acquiring knowledge [3]. Examples of user interrogation are:

   - users are required to fill out a form describing their areas of interest or other relevant parameters;
   - users are provided with a predefined set of profiles from which the user may choose the most suitable profile;
   - users are provided with a set of terms that represent each domain, from which they can construct a personal profile;
   - users are allowed to determine terms and the weights of importance;
   - users are asked to specify keywords to create their initial profile;
   - rules are provided in order to guide the user in the task of rules definition.

   The approaches described above are similar to those defined in [36], referred to as *pre-encoding* the contents of the user model.

2. Implicit: This approach does not require active user involvement, instead, the users reaction to each incoming data item is recorded (to learn from the actual relevance of the data item to the user), for example:

   - user spending time on reviewing data items to determine their relevance
   - user behaviour on information items, for example; user saves, discards, prints or forwards the data items
   - observing on the hyperlinks clicked and those passed over
   - users' past navigation history (browsing history)

3. Mixing of explicit and implicit: This approach involves a mixing of the explicit and implicit approaches. Examples of this method are [3]:

   - Document Space — This method creates a field of documents that the user has previously judged as relevant. Any new incoming document is tested for its similarity to the documents existing in that space. If

it is similar, it is considered relevant. This method considers that the user evaluates each documents relevance without a need to define the profile;

- Stereotype Inferences — In this method, users are asked to provide explicit information about themselves to enable the system to relate them to user stereotypes (it captures default information about groups of people).

Acquiring knowledge about a user involves extracting features which are representative of the user's interest; this is known as a user profile and can distinguish between interesting and non-interesting information items for that particular user. The user profile is not built once and applied unaltered thereafter. It is a long-term construct that has to continuously adapt to temporal changes in the user's interests in response to user feedback, and be able to represent the complete range of a user's interests.

2.2 An Adaptive Information Filtering (AIF)

Reviews of the state of the IF parameters which have been presented in [3, 15, 25] show some similarities, such as (a) data representation component; (b) user-model component; (c) filtering component and (d) learning component. Figure 2.2 illustrates a generic work-flow of an IF system which involves these four components. Based on the diagram, the data representation component needs to obtain or collect data items from an information provider. The data items are indexed in an appropriate format and the represented data items are the input of filtering components. On the other hand, the user-model component presents an acquisition function (explicitly or implicitly) of the information need of the users interests or details. The knowledge acquired on a user is usually kept in the form of a user profile or rules, and it is also the input of the filtering component. A comparison function, interpreted as a binary judgment, is then used to determine whether the document presented satisfies the user profile or not. Users should always have the option to enter or modify values in their profiles, such as deleting interests or updating the profile by adding new interests, and the data items also might undergo changes such as new topics arising or an existing topic being deleted. Therefore, a learning component in a filtering system is essential. This component improves further filtering and enhances filtering effectiveness, as a result of the difficulties of the user model and of shifts detected in the changes of information needs. Otherwise, inaccuracies occur in profiles that

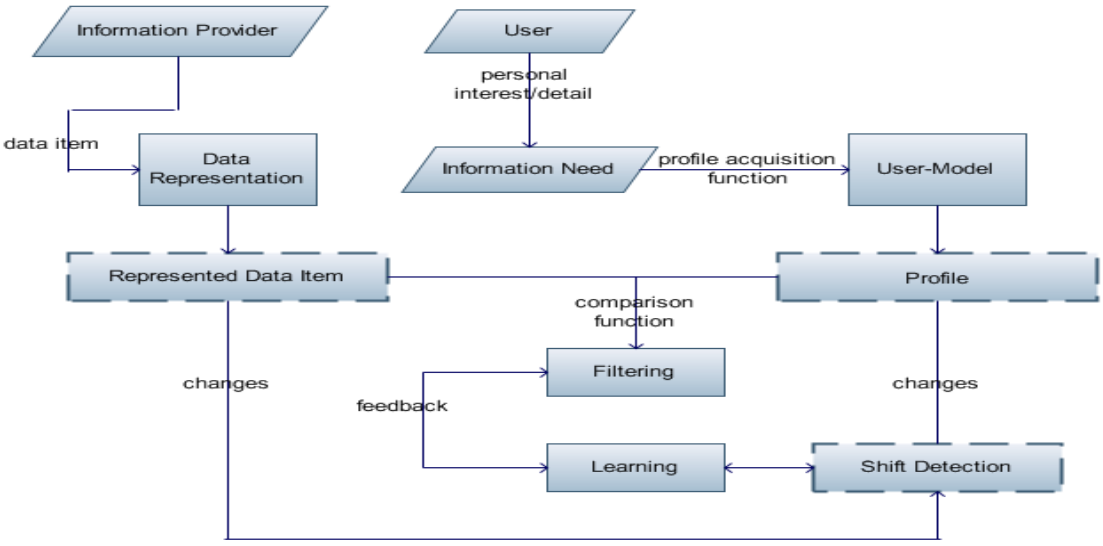


Figure 2.2: Generic Work-flow of Information Filtering

affect the filtering result.

According to Kjersti [26] a filtering system should satisfy three requirements:

- **Specialisation**: A system should be highly responsive to the needs of the user. Since filtering involves repeated interaction with user, the system should be able to identify patterns in user behaviour.
- **Adaptation**: Since interaction with users occurs over a period of time, it cannot be assumed that the user's interest is constant. When interest changes, the system must notice the changes and be able to adapt its behaviour to the changes.
- **Exploration**: A system should be also capable of 'information seeking', that is, exploring new information domains to find information that is potentially of interest to the user. In terms of information input, an IF system primarily handles unformatted textual data such as documents, semi-structured items such as electronic messages (*e-mail*), NetNews articles and NewsWire stories, or more complex structures such as hypertext documents containing voice, graphics and pictures.

Based on the criteria highlighted by Kjersti [26], therefore, IF features properties of adaptability of changing user profile interests (either of a particular person or a group of persons with a shared interest) and deals with the heterogeneous nature of information of an incoming data stream. Thus, these characteristics are likely to need a role of adaptive information filtering (AIF) systems. Tauritz [13]

defined an AIF system as "a system that is capable of adapting to changes in both the data stream and the information needs". Adaptation in information filtering involves the process of filtering incoming data streams in such a way that only relevant data (information) is preserved. The relevance of the data is dependent on the changing (adaptive) needs of a particular person or group of persons with a shared interest. In addition, the users' interest cannot be assumed to be constant, meaning, it cannot be static. Therefore, a filtering system must be responsive to dynamic user interests, to users with multiple topic interests and to users with changing interests. Changes in the user interest may be caused by changes in the user's environment (for example a job environment) and knowledge (for example new knowledge acquired through interaction). The combination of parameters causes a variety of changes (dynamic) and renders the profile adaptation a challenging research area.

As well as learning a user profile, the adaptivity must also cope with changes in data stream (for example the text stream). The data stream consists of textual documents with a specific classification [14]. To deal with the data stream, it is necessary to be able to compare the documents with the interests of the user at a given time. Looking at a stream of incoming texts, [14] consider the following types of change:

- new topics arise,
- existing topics disappear or
- existing topics change (example: the content change)

A changing topic can be interpreted as the superposition of two similar topics, and one of these topics disappears while the other arises. Lanquillon and Renz [14] argued that changes due to a disappearing topic do not have any serious effect on the performance unless this topic is similar to an existing topic but belongs to the other relevance class, or too many obsolete topics make modeling the information filter difficult. Concept drift is also related to the adaptation of the data stream. The field of concept drift aims to notice changes within a given dataset, and then adapt to these changes [37]. However, the drift field is not restricted to the dataset, but is also focused on the changes of individual objects [38] which are examined multiple times over a given time period, where they might drift from one resultant class into another.

Adaptivity not only constitutes a major concern in the IF community but has also recently, come to the attention of the IR community. The issues were dis-

cussed in the International Workshop on Adaptive Information Retrieval (AIR)[3]. Papers presented in that workshop tackled the adaptation issues from various perspectives in IR, such as what to adapt, how to adapt, and how to evaluate.

2.3 User Profile

In Section 2.2, it was emphasized that Adaptive Information Filtering (AIF) is the research domain that seeks to provide a solution to the problem of information overload by continuously providing a user with information that is relevant to the user's long-term interests. This is accomplished with a tailored representation of the user's interest, called the user profile.

A user profile or user model can be loosely described as a collection of assumptions or beliefs the system holds about the user [3]. As stated above, a user profile is not built once and applied unaltered thereafter. It is a long-term construct that has to continuously adapt to temporal changes in the user's interest. User interests change over time, driven by changes in the user's environment and knowledge. The user profile appears to adapt to a variety of changes ranging from frequent variations in a user's short-term needs, to occasional radical changes like the emergence of a new topic of interest and the loss of interest in a particular topic. This results in the profile constantly changing structurally in response to changes in the stream of feedback. Hence, the viability of an AIF system relies on the ability of the user profile to maintain a satisfactory filtering accuracy for as long as it is being used. The user profile has to be able to represent the complete range of a user's interests and to continuously adapt, in response to user feedback, to any changes in them.

2.3.1 The Challenges of Profile Adaptation

The ambiguity in representing a document, the uncertain nature of a user's information needs and the associated formulation process [15] all contribute to the difficulties encountered in creating a user profile. A user profile should dynamically adapt to drifts in users' interest and 'learn' with the changing interests. Inaccuracies in a user profile affect the quality of recommendation. These challenges of profile adaptation to information filtering pose an interesting and challenging research area. Some additional features of profile adaptation are as follows.

1. A user is interested in many topics at once, and a topic of interest may consist of related subtopics. This means that a dynamic user profile needs to

[3]http://www.dcs.gla.ac.uk/workshops/air/

represent multiple topics of interest.

2. A system supporting dynamic user profiling needs to maintain and adapt a diverse population of profiles in parallel.

3. The user profile must be capable of continuous learning and forgetting. A profile that only learns and does not forget will eventually become saturated with irrelevant features. Moreover, forgetting is necessary for maintaining an up to date representation of the user's interest.

4. User involvement is crucial because continual relevance feedback is the basis for the updating and fitness evaluation of profiles. However, user feedback may be unstable for many reasons. A user may not be very discriminating, or may have wide, shallow interests.

Dynamic profile adaptation is an example of multi modal dynamic optimization, (MDO) [16]. Through profile adaptation, the profile becomes open to its environment with the addition and removal of 'topics'. The profile constantly changes its structure in response to changes in the stream of user feedback to documents.

2.3.2 Related Work on Adaptive User Profile

There are two main variations on building a profile of user interest: indirect (also known as implicit) (that is, watching the behaviour of the user) and direct (also known as explicit) (that is, asking the user for feedback). Even when gathered implicitly, a user profile is prone become to go out-of-date as users interests change over time. Dynamic user profiles are those which update as the user task is changed. There are some existing applications which emphasize adaptive user profiling. This section will review these applications.

Letizia [39] is a system which tracks users' browsing behaviour, as the user follows links, makes search queries, or asks for help, and then uses this data to predict which pages would be of more interest to the user on the next click. Letizia automates a strategy of recommendation based on a best-first search augmented by inferred user interests based on previous browsing behaviour. The user can follow or ignore the recommendations. This system is not using the content of the pages to build the strategy but the behaviour of the user. To date no user study has been carried out to show whether this strategy is effective for the user. For the profiling systems that recommend a set of web pages as a guided tour or trail, a system such as WebWatcher [40] watches the users actions and

gains expertise on that part of the web already visited by the user. The system then recommends sequences or paths of sites to the user based on knowledge of the users previous interests and knowledge of that part of the web. The results of web searches based on known topic of interest can also be ranked according to closeness to a users preferences. Systems such as that of Syskill and Webert [41] keep a separate profile for each topic for each user. The profile is used to rate pages returned from a web search. In the Syskill and Webert system, two users rated the returned pages on a three-point scale and this feedback was used to adapt the profiles. Various learning algorithms were evaluated and preliminary results showed some improvement as the profiles adapted.

The CLEVER[4] system is a profiling application which focuses on hyperlink structures of the web and is developed on the basis of known topics. This system starts with a canonical topic taxonomy with example web pages. The user selects and/or refines specific topic nodes in the taxonomy and may provide additional example URLs which serve as starting points for the web crawl. The user may inspect the system regularly to provide direct feedback by marking pages as useful or not useful. This system has achieved good results based on known topics, taxonomies of topics, and the incorporation of the link structures into the process. A more intrusive use of profiles for users is found in systems such as Avanti [42] where the profile is used to change the content and appearance of Web pages for the user using a set of adaptation rules. This has been used for elderly or handicapped users. Community profiles are also used to provide a shared community level of feedback that can then be used by members of that community. Footprints [43] is an example of this type of system. Visitors can see common paths through a website as an aid to navigation at that site. Finally, Nootropia [44] is a user-profiling model for content-based document filtering which uses a non-linear term network to represent a user's multiple interests and which self-organises in order to adapt to both short-term variations and substantial changes in them. In Nootropia, user-profile adaptation is achieved using a deterministic process that calibrates the weight of profile terms, removes incompetent terms and recruits new candidate terms. In the process, the profile becomes open to its environment and operates far from equilibrium, constantly adjusting in response to changes in relevance feedback. As a result, new structures (hierarchies) and new modes of behaviour (document evaluation) are generated.

We have reviewed some of the existing applications for adaptive user profiling. To develop an adaptive user profiling there needs to be an algorithm to learn

[4]http://www.almaden.ibm.com/cs/k53/clever.html

the profile. The details of the algorithms which have been used in the study of profile adaptation domain are reviewed in the next section.

2.4 Algorithms for Learning Profile

Having noted the existing applications which emphasize adaptive user profiling in Section 2.3.2, this section gives an overview of some existing algorithms for learning profiles. A review of the existing profile adaptation algorithms is carried out based on machine learning algorithms (Section 2.4.1), evolutionary algorithms (Section 2.4.2) and connectionist algorithms (Section 2.4.3). Due to the large number of studies in information filtering, this section is necessarily incomplete.

2.4.1 Profile Adaptation through Learning

There has been a tendency to seek an adequate solution to the problem of profile adaptation in machine learning algorithms. In this section, some of the existing approaches for profile adaptation using machine learning algorithms are discussed. The algorithms are as follows:

Rocchio's Learning Algorithm

Rocchio's relevance feedback is an algorithm for learning user interests that has been well studied in information retrieval (IR) [22, 23] cited by [27]. This algorithm is an example of adaptation in an IR system. Systems employing the Rocchio's algorithm typically assume the stability of user interests and apply the algorithm as a batch process. Given an initial query vector, Q a new vector, $\acute{Q}$ is generated using Equation 2.1 where D_i^R and D_j^N are the vector representations of the i^{th} and j^{th} relevant and non-relevant documents respectively. Parameter $\mid \eta_R \mid$ and $\mid \eta_N \mid$ refers to the set of related and non-related documents respectively. For parameter α, β and γ it determine respectively how much the initial query and the relevant and non-relevant documents contribute to the formulation of the updated query. The original Rocchio's algorithm instantiates the parameter as $\alpha = 1$, $\beta = 2$ and $\gamma = 0.5$. Rocchio's algorithm updates the query weights linearly at a rate that depends on a feedback parameter [23].

$$\acute{Q} = \alpha Q + \frac{1}{\mid \eta_R \mid}\beta \sum_{i=1}^{\eta_R} D_i^R - \frac{1}{\mid \eta_N \mid}\gamma \sum_{j=1}^{\eta_N} D_j^N \tag{2.1}$$

The adaptability to react to changing interests can be controlled from the weights assigned to a positive and a negative feedback on a document. However, the linearity in updating the user interest representation makes it difficult to quickly remove a long-standing interest [45]. Another problem which is encountered is that Rocchio's algorithm is a batch algorithm [27] in which, a set of relevant and preferably non-relevant documents is required for the algorithm to be effective. This is not the case for dynamic information sources where adaptation should be achievable on a per document basis [27].

Reinforcement Learning Algorithm

Reinforcement learning (RL) is about learning from interaction how to behave in order to achieve a goal based on interactions with the environment [46] cited by [47, 48]. The learner receives a scalar-valued feedback called a *reward* when it chooses and takes an action at a given time and a given state. The objective is to maximize the expected value of the cumulative *reward* it receives in the long run from the environment [46]. The pace of the learning profile in the RL algorithm is defined by appropriately adjusting the learning coefficient over time. This approach has drawbacks for both large and small learning coefficients. A large learning coefficient causes the profile to be adapted rapidly to short-term needs, which may lead to over-specialization to the most recent documents, while a small learning coefficient can cause high profile inertia, which hinders the profile's responsiveness [27]. In learning the user interest, the retrieval agent seeks the relevant documents, directed by the value of reinforcement learning [40]:

$$Q_{n+1}(s,a) = R(s') + \gamma_{a'\varepsilon actions-in-s'} max[Q_n(s',a')] \tag{2.2}$$

In Equation 2.2, Q - value is the discounted sum of the future rewards that will be obtained when the agent follows a hyperlink in an HTML document and subsequently chooses the optimal hyperlink. The application of reinforcement learning to profile adaptation has been employed by [15, 40, 47].

2.4.2 Profile Adaptation through Evolution

The profile adaptations that have been reviewed so far concentrate on learning the profile through learning algorithms. There has been a study of profile adaptation which uses the approach of an evolutionary algorithm. In evolutionary IF, a population of profiles is maintained which collectively represent the user interests. The population evolves according to user feedback. Individual profiles

that better represent the user interests become fitter, reproduce and proliferate, while those that do not receive positive feedback are eventually removed from the population.

Genetic Algorithms (GAs)

Profile adaptation in GAs relies on evolving a population of user profiles in response to user feedback [49]. The fitness evaluation function rewards those profiles that have received positive feedback and vice versa. Successful profiles mate to produce offspring which can represent the current user interests more accurately and which replace profiles in the population that are no longer successful. Thus profiles that represent topics of interest proliferate and those that do not are removed from the population eventually. Random mutation of profiles allows for further exploration of the information space for areas of interest. Algorithm 1 provides a basic GA for information filtering.

```
t ← 0;
initialise P(t);
use P(t) to evaluate documents;
while user provides feedback do
    evaluate P(t);
    update p_i ε P(t) based on feedback;
    select F(t) ⊂ P(t) based on fitness;
    for p_i and p_j ε F(t) do
        crossover p_i ⊗ p_j → o_i,o_j;
        occasionally mutate o_i,o_j;
        replace less fit individuals in P(t) with o_i and o_j
    end
end
```

Algorithm 1: Basic GA for information filtering [11]

GAs are well suited to the problem of profile adaptation by combining a global search with a local search of the information space, thus, each individual may improve through modifications in its chromosome, which the individual's offspring will inherit [13, 49]. The population's evolution was guided by user feedback. As a result, this approach was able to adapt the profile in both long-term changes and short-term variations in the user's interests. However, Nanas et. al in [11] argued that GAs suffer from the following factors:

1. Multimodal Dynamic Optimization, (MDO): In MDO there is no single and static optimum, but instead, a varied number of optima that continuously

change their position and shape in the solution space [16]. Profile adaptation is an example of MDO, where a user may be interested in multiple topics in parallel and interest changes are time dependent. Thus, there is no single and static optimum, but rather various and changing circumstances. When dealing with MDO, GAs face diversity problems due to the combined effect of selecting parents for reproduction based on their relative fitness and fixed population size, which implies the offspring replace existing individuals and typically the less 'fit' [16] .

2. Maintaining diversity: In time-dependent problems, such as profile adaptation, user interests change over time. GAs suffer because of their tendency to converge and therefore lose diversity progressively because of radical changes in user interest.

3. Fitness bottleneck problem: Evolutionary IF systems are user dependent and therefore, there is an inherent fitness bottleneck problem. User involvement is crucial because relevance feedback is the basis for the fitness evaluation of profiles. There is no objective fitness function that can be used at any time for chromosome evaluation. The evolutionary process continues as long as user feedback exists.

4. High computational cost: A diverse population of profiles has to be maintained and adapted in parallel. In addition, the relative importance of topics represented by individual profiles is reflected by their fitness but not by the representation itself [11].

2.4.3 Profile Adaptation through Connectionist Architecture

The concept of connectionism has attracted the attention of researchers over the past decades. The approach is not new in the domain of text processing since it includes the information retrieval, text categorization and information filtering. In the connectionist approach to an IF system, the IF system may be viewed as operating in two modes. Initially, there is a learning phase, in which a collection of documents of interest to the user is presented to the system. Then, there is the comparison phase, in which documents arriving via an information stream (particularly the web) are filtered to the user. If these documents are considered relevant by the user, then the system enters learning mode so as to improve its filtering ability.

The system thus consists of a two-layer network. The purpose of the first (or bottom) layer is to receive an article as input and build a representation of

it. This layer is then analysed, resulting in the formation of a second (or upper) layer of supervisor nodes which monitor activity in the layer beneath when in the comparison mode.

For profile adaptation, the connectionist approach has been achieved by using either a Self-Organisation Map or the Hebbian Learning Network of linked weights. This section will review these approaches.

Self-Organizing Map (SOM)

A Self-Organizing Map (SOM) is a type of neural network which combines nonlinear projection, vector quantization (VQ) and data-clustering functions [50,51]. A SOM can map the originally high-dimensional document space onto a two-dimensional map grid which expresses content similarity between documents in an intuitive graphical fashion [51]. For profile adaptation, the SOM algorithm is used to learn and update the user profile. The formation of a user profile in SOM is typically based on the user search history based on Vector Space Model (VSM) of inverted index for all the documents in the search history.

The SOM model consists of a set of neurons organized into a two-dimensional regular structure which is composed of two layers of neurons, the input layer and the output layer (which is also called the competition layer). Let us denote the input vector as $X = (x_1, x_2, ..., x_n)^T$, then the connecting weight between the neurons in the input layer and those in the competing layer is $W_j = (w_{1j}, w_{2j}, ..., w_{nj})^T$, $j = 1, 2, ..., h$, and the output competition layer neurons are shown in Equation 2.3. During the training process, a SOM can change the disordered input set into an ordered topology connection in the competition layer.

$$y_j = \sum_{i=1}^{n} w_{ij} x_i = W_j^T X, j = 1, 2, ..., h \tag{2.3}$$

The self-organizing process is applied to see the connection weights that best match the input vector based on the following criterion:

$$win(X) = argmin||X - W_j||, j = 1, 2, ..., h \tag{2.4}$$

where $|| * ||$ is the Euclidean norm of the argument vector, and $win(X)$ is the corresponding neuron win called the winning neuron for the input vector X. Finally, the neuron is updated by the following rule:

$$m_i(t+1) = m_i(t) + h_{c_i}(t)[x(t) - m_i(t)] \forall_i \in N_c(t) \tag{2.5}$$

where $m_i(t+1)$ is the node's weight vector with t as the index for the recursive

steps. The scalar multiplier $h_{c_i}(t)$ is called the neighborhood function and it is like a smoothing kernel over the grid [51] and the $N_c(t)$ is the total number of nodes used in the SOM.

The SOM approach to user profiling has been applied in [52] for personalization in web search and in [53] for dynamic user modeling. Although SOMs have been shown to be successful at modeling the user profile, they do not cover multiple interests. Single topic profiles are described in both cases.

Hebbian Learning Network

The Hebbian learning network is an unsupervised learning model in which the basic idea is that, "the synaptic weight is increased if both an input and output are activated" [54] cited by [55]. The Hebbian learning network in IF consists of two layers, the input layer and the output layer. For the input layer (or first layer), each node has a name (each individual word in an article is assigned a node). 'Frequency' is the number of times that the word has occurred in the present article. Each node has a number of associates (outgoing links) and supporter (incoming links). The script formation process in the input layer will form two children, a left (A) and a right (B). In the output layer (or second layer), for each suitable script formed in this layer, there exists a supervisor node whose purpose is to monitor for the presence of the script in each article under examination. If and when the script is detected (i.e. the two words appear successively in an article), the appropriate supervisor is activated and is permitted to make a contribution (via its output value) to the interestingness rating of the article [55]. Links are formed between supervisors when it is found that two supervisors occur in sequence in the learning phase. For example, one supervisor may have been created to monitor for the term 'object oriented' while another may have been created to monitor for 'oriented programming'.

Each node in the network is initially provided with S-entity points to capture the preference of one node's link with another relative to its other outgoing links [55]. Generally, the Hebbian learning network employs two separate weighting schemes for S-entities [56], one based on *weight* to estimate the likelihood that one term will appear after another (i.e. for some word A, it measures the likelihood that other words B, C, D and so on will appear with it in interesting articles). The second measure is *strength* to measure the relative importance of one S-entity as compared with all others appearing in the profile. The weight and strength of each S-entity in a profile are adjusted using unconstrained, constrained or both of Hebbian Learning respectively. This adjustment occurs during the initial profile

construction phase and as a result of ongoing relevance feedback. In [56], the authors simplified a rule of anti-Hebbian learning to construct the initial profile. The rule is as follows:

> "The strength of the link between one word and another depends on how often these words occur in sequence in interesting articles. This link is strengthened each time the sequence occurs and possibly weakened when it does not. Each word has a maximum strength which the sum of its link strengths cannot exceed".

To our knowledge, there is a limited number of works on Hebbian learning networks to profile adaptation. Although work in [56, 57] and [55] is not solely on profile adaptation (these works concentrate much more on adaptive linking to document collection), even so however, they emphasize an adaptive user profile for continuous revisions of the user profile based on feedback from the user rating of the retrieved pages. The user profiles consist of weighted keywords and the adaptation is based on the Hebbian Learning Model with direct user feedback. In [58], the authors have implemented a collaborative system that develops linked structures based on user browsing patterns and Hebbian learning. Users were asked to perform associate browsing, that is they were not given topics, but were given English nouns and asked to browse for associated terms. The browsing patterns were used to build and rank associative links among pages. This work may have a place in adaptive structuring of the web, but has to be tested on large scale sites. Not all of these works however, cover multiple topics of interest.

2.5 Summary

This chapter presents the underlying concept of Information Filtering (IF), covering its parameters, components and processes. IF that ranks and presents incoming documents according to a particular user's interest is a user-oriented service. The IF problem by its nature can be seen as a classification problem; all documents can be classified as belonging to either a positive class (relevant to the user) or a negative class (not relevant to the user). IF environments feature properties of adaptability of changing user profile interest (either of a particular person or group of persons with a shared interest) and deal with the dynamic nature of a data stream. These characteristics are likely components of adaptive information filtering (AIF) systems. One of the goals of AIF research is to develop a filter system that can cope with changes in user interest. This AIF system is based on a tailored representation of the user's interests, called a 'profile', which assesses

the relevance of information items, which are then appropriately presented to the user. The user expresses satisfaction or dissatisfaction with the results of the filtering by means of relevance feedback which can be either implicit, explicit or both. Over time, a user may develop interest in more than one topic in parallel and interest changes inevitably occur. As the user's interests change, the user profile has to be able to trace, represent and constantly track these interest regions over time. The system has to be able to adapt based on user feedback to various changes in a user's multiple interests. These challenges present a fascinating research area for profile adaptation in AIF. Various algorithms for profile adaptation domains have been developed and are reviewed in this chapter. The review of the existing approaches is typically based on three main approaches; learning, evolutionary and connectionist approaches. Due to the large number of studies of information filtering, this review is necessarily incomplete.

CHAPTER

THREE

THE POTENTIAL OF ARTIFICIAL IMMUNE SYSTEMS (AIS) TO INFORMATION FILTERING (IF)

In this chapter, the natural Immune System (IS) and Artificial Immune Systems (AIS) are discussed. This chapter begins with the motivational description of the natural immune system which includes an explanation of innate and adaptive immunity. The chapter then goes on to discuss AIS (Section 3.2) and two AIS frameworks will be described, in some detail, the conceptual framework and the engineering framework. Having noted the potential of AIS for information filtering (IF), Section 3.3 presents a comprehensive review of AIS in the IF domain. A review of work on the relation of AIS to IF will identify the immune inspiration used in the IF applications and discuss how the AIS approach is adapted to the application of IF. In Section 3.4 there will be a discussion about the principled meta-probes for identifying an appropriate characteristics of immune-inspiration and the application domain of IF. These principled meta-probes are based on the conceptual framework which address notions such as Openness, Diversity, Interaction, Structure and Scale, otherwise known as the *ODISS*. As an outcome of this high level abstraction, we summarize the result of applying these principled meta-probes both to the natural immune system and to the target IF domain, namely the adaptive information filtering (AIF). From the principled meta-probes, the discussion then focuses on the AIS algorithm called the Dynamic Clonal Selection Algorithm (DCSA), which was specifically developed for a range of adaptivity functions in dynamic environment problems, that is,

adaptive information filtering (AIF). This chapter ends with a discussion about the relevant biological inspiration and algorithm development of DCSA, which is presented in Section 3.5.

3.1 The Immune System in Context of the Biological Perspective

The immune system's job is to detect foreign invaders, primarily microbes, tiny organisms such as bacteria, parasites, fungi and viruses, which can cause infections [59]. The IS is a complex system which works in a network of cells, tissues and organs to defend the body. The immune system has the ability to 'remember' enemies that it has fought in the past. If the IS detects a 'registered' invader, it will strike much more quickly against it. As a result, an invader which tries to attack the body for a second time will most likely be wiped out before there are any symptoms of disease. When this happens, the body has become immune. Ishida [60] characterized IS as:

- a *self-defining* system that creates, organizes and maintains the identity of the self; and
- an *adaptive system* that implements an adaptive mechanism based on 'selection' to realize the self-defining system.

However, he was not the first to characterize IS as a self-defining system; Cohen in [61, 62] noted that the IS is about body maintenance, in which, to keep the body fit, the IS not only depends on the right type of inflammatory response in concert with the needs of the situation, but, at the same time has need to orchestrate a spectrum of responses dynamically over time according to the shifting needs of the tissue. The mechanisms to achieve this are *immune dialogue* [63] and *immune correspondence* [62]. Immune dialogue arises because the IS continuously exchanges molecular signals with its interlocutor, the body and additionally, both adjust their behaviour in the light of the signals (one-way signal such as antibodies and two-way signals such as cytokines) which each receive and send to the other [61]. The immune correspondence arises because different classes of immune cells respond to different aspects of any single immune object, self and non-self [61, 62]. A detailed review of the natural immune system and its functionality can be found in [4, 59, 60, 62, 64, 65].

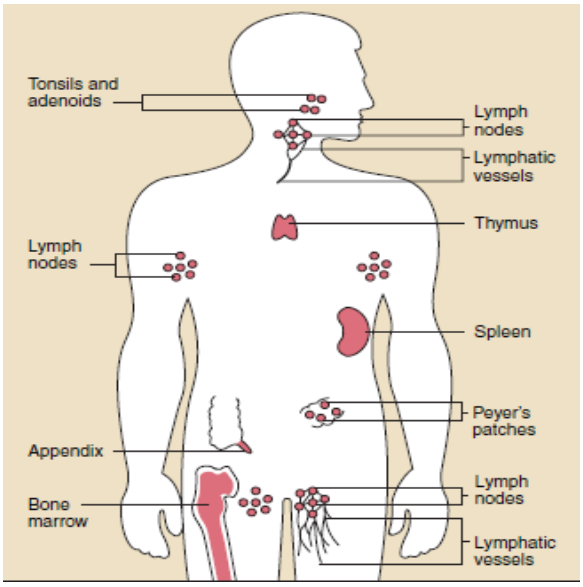


Figure 3.1: Immune organs are positioned throughout the body [4]

3.1.1 Structure of the Immune System

There are many immune organs that make up the immune system and they are stationed throughout the body, as shown in Figure 3.1. These organs are called *lymphoid organs* because they are home to lymphocytes, the white blood cells that are the key players in the immune system.

The organs of the immune system either create the cells that participate in the immune response or act as sites for the immune function [59]. These organs of the immune system are connected with one another and with other organs of the body by a network of lymphatic vessels which are similar to blood vessels. In [4], lymphoid organs are divided into *primary* (or *central*), organs which are responsible for the production and maturation of lymphocytes, and *secondary* (or *peripheral*) organs where the lymphocytes interact with the antigenic stimuli, thus initiating an adaptive immune system.

3.1.2 The Defence Layer

The immune system protects organisms from infection with multi-layered defences [4, 66]. Figure 3.2, which is adapted from [4], illustrates these layers of the defence system. The three main layers include the anatomic barrier [67], innate immunity and adaptive immunity, described as follows:

- *Anatomic barrier*: This barrier acts as the first layer of the defence system. It is composed of the skin and the surface of the mucous membranes.

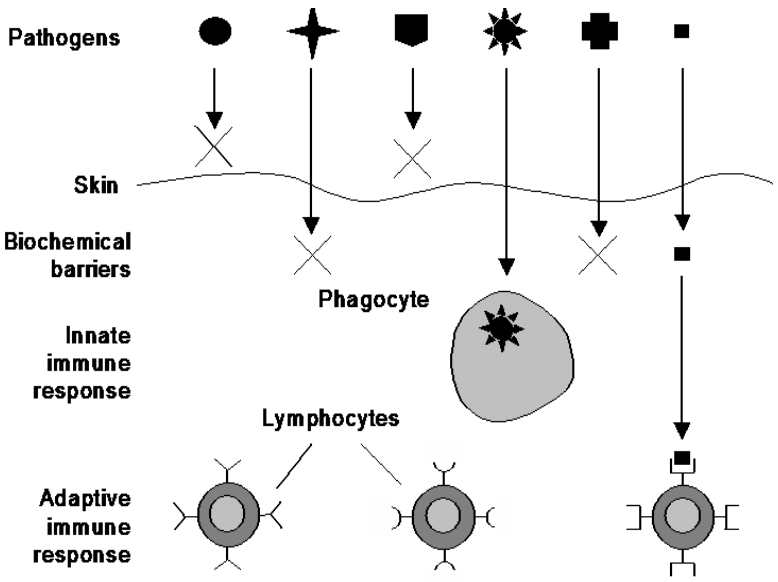


Figure 3.2: The Immune System Defence Layer, [4]

- *Innate immunity*: Once pathogens have entered the body, they are handled by the innate immune system and the adaptive immune response. The innate immune system provides an immediate, but non-specific response. The term *innate* refers to the part of immune system that individuals are born with and which does not adapt over a person's lifetime [4].
- *Adaptive immunity*: The term *adaptive* is so-called because it adapts or learn to recognize specific kinds of pathogen and retains a memory of them [66]. The improved response allows the adaptive immune system to mount faster and stronger attacks each time a known pathogen is encountered.

Both adaptive and innate immunity make distinctions between the ability to distinguish self from non-self reliably most of the time. In [68, 69] it is stated that receptors for these two types of immunity are encoded in fundamentally different ways. The receptors of the innate immune system are encoded in the *germline*, and are expressed without rearrangement, and by most or all cells of a given type, while by contrast, the receptors of the adaptive immune system are encoded in rearranged gene segments (*somatically encoded*). Table 3.1, adapted from [69], indicates the differences between recognition of self and non-self in the innate and adaptive immune systems. A detail explanation of this can be found in [68, 69].

Property	Innate IS	Adaptive IS
Receptor	Germline encoded.	Somatically generated.
Response	All the cells can express the same receptor, no need for clonal expansion. The response is immediate.	Each specificity is expressed on a single cell, thus, the effector functions can only be performed after clonal expansion.
Self/ Non-Self Discrimination	Perfect: selected over evolutionary time	Imperfect: selected in individual somatic cells.
Action Time	Immediate activation.	Delayed (lag time) activation.

Table 3.1: The differences between the Innate and Adaptive Immune Systems [68, 69]

3.1.3 Immune Cells

The defensive cells are more commonly known as *immune* cells. The cells of the immune system work together with different proteins to seek out and destroy any foreign or dangerous entities that enter the body. Immune cells are white blood cells, the *leukocytes*, produced in huge quantities in the bone marrow. There are a wide variety of immune cells; some seek out and devour invading organisms, while others destroy infected or mutated body cells. Another type has the ability to release special proteins called *antibodies* that mark intruders for destruction by other cells [59]. Figure 3.3 illustrates the distinct pathway of the immune cells. A detailed review of immune cells and their functionality can be found in [4,59,64].

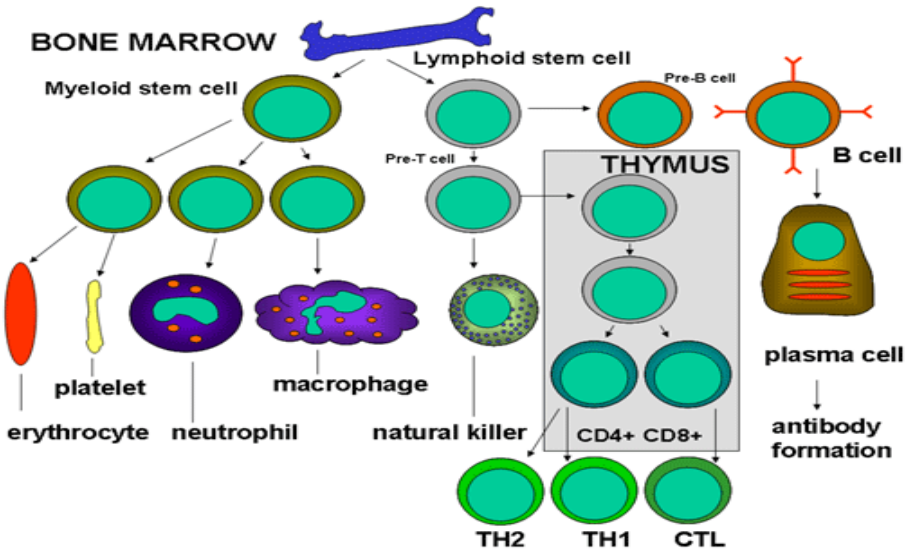


Figure 3.3: The Development of Immune Cells[1]

3.1.4 The Immune Response

The immune response is concerned with how the body recognizes and defends itself against bacteria, viruses, and substances that appear foreign and harmful to the body [65]. When the body is injured, an inflammatory response [59, 64] is triggered and a reaction, pain, serves to alert the individual to the injury. Cells that have been damaged by injury or invasion release a number of chemical signals, such as *histamine* and *cytokines*. Some of these chemical signals act to attract specific types of white blood cell to the site of damage or injury. Then *phagocytic* cells such as *macrophages* migrate to the area of infection to attack the bacteria and destroy them by engulfing and digesting them and then displaying parts of the bacteria on their surface. This signal attracts other immune cells such as T cells and B cells to help in the fight. The host antigen-presenting cell (APC) expresses on its surface co-stimulatory molecules. These molecules, working together, can both attract naive T cells through the secretion of *chemokines* and activate naive T cells to respond to specific antigens of the pathogen. A helper T cell which binds to a bacterial antigen sends out signals to T cells and other immune cells to participate in the immune response. Once T cells are activated, the adaptive immune response takes over.

[1] http://pathmicro.med.sc.edu

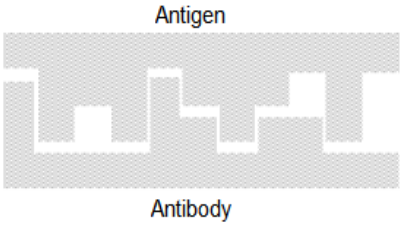


Figure 3.4: The antigen antibody binding via regions of complementary [4]

An adaptive immune response comprises two broad classes of response [59]; *antibody responses* and *cell-mediated immune responses*, which are carried out by different classes of lymphocytes, called B cells and T cells.

- *antibody responses*; B cells bind to bacterial antigens directly (or through macrophage presentation). In response to this binding and chemical signals from helper T cells, B cells multiply and transform into memory B cells and plasma cells. Plasma cells make antibodies that are specific to the bacteria that triggered the response. The antibodies circulate in the bloodstream where they bind specifically to the foreign antigen that stimulated their production. De Castro and Timmis [4] pointed out that, in order for an antigen to be recognized, the antigen, *Ag* and antibody, *Ab* must bind with each other over *extensive regions of complementary*, see Figure 3.4. This interaction of binding is determined by the set of the antigens' and antibodies' properties which are called the *generalized shape* of a molecule. Antibody binding also marks invading pathogens for destruction, mainly by making it easier for *phagocytic* cells of the innate immune system to ingest them.

- *cell-mediated immune responses;* Helper T cells activate other types of T cell in the body to participate in the immune response. The natural killer T cell, for example, kills a virus-infected host cell that has viral antigens on its surface, thereby eliminating the infected cell before the virus has had a chance to replicate. Killer T cells bind to a virus-infected cell with assistance from helper T cell signals. Once the killer T cell binds to virus infected cells, the killer T cell will destroy the infected cell, killing the virus and limiting viral infection [59]. In other cases, the T cell produces signal molecules that activate macrophages to destroy the invading microbes that they have *phagocytosed*.

3.2 AIS: Artificial Immune Systems

The Immune System (IS) has been explored in terms of its nature, which is adaptive, self-organized and diverse. This has motivated computer scientists to develop a new computational paradigm inspired by the natural immune systems, known as an Artificial Immune System or AIS. Some researchers have defined AIS according to their own understanding and belief, for example in [4, 70, 71]. However, this present study adopts the definition from [4] which defined the AIS as

> ".... the adaptive systems inspired by theoretical immunology and observed immune functions, principles and models, which are applied to complex systems"

The definition of AIS in [4] covers two important aspects; first, AIS is *inspired by* but not *constrained by* the biological processes of the immune system. This has the implication that the developed AIS does not have to be an exact equivalent of the immunological processes on which it is based. Rather, it is an abstraction of relevant immunological properties that can be utilised for problem solving. Second, the primary motivation for developing an AIS is to solve engineering problems.

Early studies on AIS began in the 1980s as a new computational research area. Work by Farmer et al. [70] is considered to be the pioneer work related to the artificial immune systems. Their work described a dynamic model for the immune system based on the immune network hypothesis. Later on, work described in [72, 73] built computer security systems which discriminated between self and non-self. Further, a long-term research project has been established in order to build a computer immune system [66, 74, 75]. Recently, AIS research has become an interdisciplinary study with specialist researchers focusing on the biological area [76], the mathematical aspect [77] and the engineering [78]. Others have focused on designing the AIS as engineering-oriented with less emphasis on understanding and extracting key biological properties [79–81].

Over the years, AIS has been successfully applied to a number of problem domains such as data mining, pattern recognition, anomaly detection, optimisation, adaptive control and computer security. However, as stated above, there are instances of AIS drifting away from biological models and attention to biological detail, and putting the focus more on engineering-oriented approach [82] (later extended in [5]). These AIS studies have been criticised as suffering from what is described as *the reasoning by metaphor* approach, in which the AIS algorithms

Immune Inspiration	AIS Algorithm
Self-NonSelf Discrimination	Negative Selection Algorithm (NSA) [72]
Clonal Selection	CLONALG [8], Dynamic Clonal Selection Algorithm (DCSA) [91]
Immune Network	aiNET [92], AINE algorithm [93], RAIN [94]
Danger Theory	Dendritic Cell Algorithm (DCA) [95]
Tunable Activation Threshold	Receptor Density Algorithm (RDA) [96]

Table 3.2: Example of AIS Algorithms and their Corresponding Immune Inspiration

were developed directly from a naive biological model without much analytical framing of the representation's properties [82]. As a response to this criticism, the attention of researchers has been attracted towards paying more attention to the underlying immunological system which serves as the inspiration, and developing an abstract computational model of the underlying immunology in order to help understand the computational properties of the immune system and working more closely with immunologists to better understand the biological aspects of the system. Example of such works are those of Stepney et al. [5], Twycross and Aickelin [83], Andrews and Timmis [79] and Bersini [84]. This in fact has been mentioned in [85], where the author urged computational scientists to embrace working with the immunological community to aid the understanding of the nature of immune computation which, will then lead to the development of richer and more effective immune inspired engineered systems.

AIS have been inspired by many different aspects of the immune system. Much of the development of AIS algorithms is basically inspired by immune theories such as *self/non-self discrimination* [72], *clonal selection* [86], *immune network theory* [87], *danger theory* [88, 89] and *tunable activation threshold* [90]. Examples of different types of AIS algorithm are depicted in Table 3.2

3.2.1 Framework for Artificial Immune Systems

As explained in Section 3.2, there is a need to pay attention to the underlying biological system and develop an abstract model (to understand the computational properties of the biological inspiration, that is the immune system) in order to build effective immune-inspired engineered systems. In this section, the frameworks for AIS will be described which will act as a guideline in the process of

going from immunology to engineered systems. The frameworks covered in this work are the conceptual framework, the layared framework and the immuno-engineering framework. Although there are other existing AIS frameworks, for example, the ARTIST framework [66], framework suggested by Dasgupta [74], those frameworks are specific to a particular application domain, computer security.

The Conceptual Framework

In an attempt to build the systems that resemble the intelligence found in natural systems, Stepney et. al [82] (later extended in [5]) argue the need for a well-formed framework for the development in bio-inspired computing systems. They proposed a *Conceptual Framework*, to enable the development of bio-inspired algorithms in a more principled way. Although it was done in the context of AIS, it can be generalized. Indeed, the conceptual framework can be seen as a methodology for the development of bio-inspired systems. Moreover, the framework promotes the use of an interdisciplinary approach to develop and analyze the algorithm [5]. This is summed up by Andrews & Timmis, who state that:

> "The framework aims to stop the designer from making naive assumptions about biological processes that are providing the inspiration, and thus preventing the development of algorithms that are just a weak analogy of the process on which they are based."([79],p.133)

The conceptual framework proposed by Stepney et al. [5] gives rise to a principled approach that attempts to capture immunological knowledge which will lead to a better understanding of the natural immune systems. This is important in order to decide which aspects of the immune system must be studied to generate the required behaviour and which aspects are surplus to requirements in the implementation of a solution to a particular problem being studied [6]. The first stage of this framework, as outlined in Figure 3.5, is to *probe*, observe and experiment with biological immunology. With this in mind, an abstract model of the immune system can be built and validated through mathematical or computational techniques. The execution and validation of a model provide the principles for designing and analysing *an immune-inspired algorithm*, possibly tailored to a range of problem domains.

To identify the appropriate characteristics of the immune-inspiration and the application domain, Stepney et al. [5] suggested that the underlying properties of classes of model could be analysed at a higher level known as 'meta-questions'.

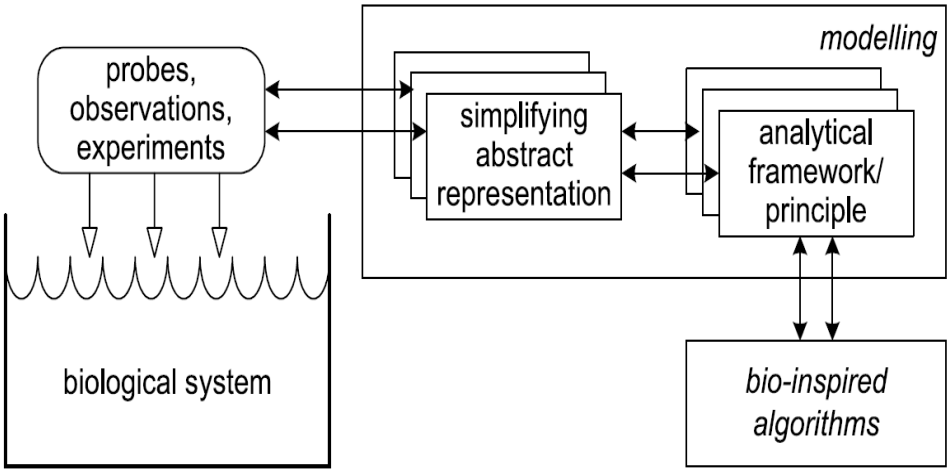


Figure 3.5: The Conceptual Framework [5]

These meta-questions address notions such as Openness, Diversity, Interaction, Structure and Scale, otherwise known as the *ODISS* meta-probes. The ODISS meta-probes are used to challenge the biological system (as in [5,83]) and the application domain, and to identify matching characteristics and ODISS properties. The probes are not independent. For example, interaction supports and underpins openness and assists in maintaining diversity; diversity is, in part, a factor of scale and structure. Scale and structure are both concerned with the layering of complexity in systems. In this work, we used the meta-probes to analyse and compare the characteristics of an application domain (the adaptive information filtering) and aspects of the mammalian immune system, in order to extract natural idioms for adaptation into a user profile. This is explained in Section 3.4.

The Layered Framework

If an AIS is to be used in engineering, rather than simulation of principles and processes, it needs suitable design guidelines. For this reason de Castro and Timmis in [4], propose a layered framework for engineering AIS. This framework succinctly demonstrates the general structure of most AIS, and so is used here as a template for this description of the main AIS types. The layered framework takes the application domain of the AIS as its starting point, followed by three layers to be considered before the required AIS is engineered. These layers are:

- *Component Representation*: how the components of the system are to be represented.
- *Affinity Measures*: how the interactions between the components of the sys-

tem are to be quantified.

- *Immune Algorithms*: how the components of the systems are going to interact to determine the system dynamics.

The most influential concept to affect the representation of components in AIS was introduced by Perelson and Oster [97], who defined the notion of *shape space*. Their study viewed the immune system as a molecular recognition device designed to identify 'foreign shape'. They state that the antibody, Ab and antigen, Ag bind perfectly when $Ab = Ag$ if antibody combining regions and antigenic determinants are complementary . Even though work of Perelson and Oster [97] used antibodies and antigens, de Castro and Timmis [4] point out that this shape space representation can be applied to any type of receptor and molecule that binds it and, typically, an AIS component (e.g.an antibody) can be represented as an attribute string (set of coordinates) of length L-dimensional that might be composed as real values, integers, bits and symbols. The choice of these coordinates (string) is driven by the problem domain of the AIS and important in the definition of which measure(s) will be used to quantify the interaction.

Recently work by McEwan [98] introduced the concept of 'boosting' in the immune system as an alternative approach of common AIS abstraction of shape-space. 'Boosting' has known to be a general method in machine learning community for improving the accuracy of any given learning algorithm [99]. AIS shape space notions of affinity may be a poor abstraction by the following reasons: "they do not scale to large computational intelligence domain; biological aspects are implausible and they cannot make an operational distinction between context and signals necessary to realizing constructive problem representations in an on line setting" [98]. Their work on immune-inspired augmentation to boosting, however, does not show how these properties can be aggregated to each other.

Once the suitable representation has been selected, one or more sets of functions are determined to quantify the interaction between the elements of the system. This measurement is termed *affinity* measure or distance measure. There are many possible affinity measures, such as the Euclidean, Hamming and Manhattan distance. Like the shape-space, the choice of the distance measure is also depending on the problem domain, and on the type of shape-space. Detailed explanation of shape-space and affinity measure can be found in [4]. The immune algorithms are typically inspired by the immune processes covered in Table 3.2, and falls into one of four groups: negative selection, clonal selection, immune networks and danger theory. Detail on these types of algorithms and their appli-

cation can be directed to [4].

Note that, the choice of representation, the affinity measure and the immune algorithm depend on the application domain. This approach leads to the search for solutions with derivation of the AIS components that are oriented to the problem studied. This kind of approach has been proposed by Freitas and Timmis [17,100], who outlined the need to consider carefully the AIS component most suitable to the application domain when developing AIS. Many other studies for example, [79,82,101] and [102] agreed with the approach.

The Immuno-Engineering Framework

It has been emphasized that AIS is a diverse area between immunology and engineering, developed through the application of techniques such as mathematical and computational modeling of immunology. Consequently, an initiative called immune-engineering [6] which is in line with the conceptual framework [5] has been proposed. The framework comprises four disciplines; biology, computer science, mathematics and engineering, which enables the development of a biologically grounded and theoretical understanding of AIS, thereby, enabling the engineering of robust systems. Figure 3.6 depicted the framework for immuno-engineering. Timmis et al. [6] claimed that the immuno-engineering framework not only allows for the potential development of engineering AIS, but also allows feedback to biology from computation. For this work, the immuno-engineering approach is followed as a principled way to the engineering of an immune-inspired system which needs a combination of the approach to conceptual framework [5] to instantiate the immunological properties and problem-oriented perspective [17] (in our case the profile adaptation for adaptive information filtering). The problem-oriented perspective provides a guideline to recognize specifically the problem domain because specific applications or domains have specific requirements. In our work, we adopt the ODISS meta-probes to identify the appropriate characteristics of the immune-inspiration that are suitable to the application domain which later will be explained in Section 3.4. More precisely, there is a need to careful consider the representation issues, similarity distance measure (affinity) and the immune process that are tailored for the data and the application domain of profile adaptation.

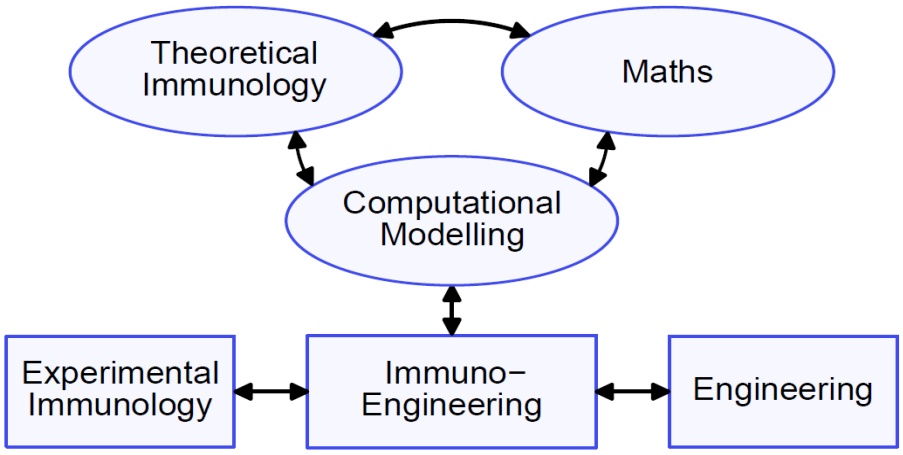


Figure 3.6: The Immuno-Engineering Framework from [6]

3.3 The Potential of Artificial Immune Systems in Information Filtering: Application Review

In this section, we review some of the existing work on immune inspired IF. The purpose of the review is to identify the immune inspiration adapted in the study, the representation and affinity issue applied in the existing immune inspired IF applications, and the limitations on the usefulness of existing immune inspired applications to the IF application domain. The discussion is categorized into two types of immune approach (as highlighted in [4]), namely a network-based approach and a population-based approach. For each of these two approaches, the discussion highlights the following characteristics:

- reference and problem solved: the kind of IF application task addressed by that work;
- algorithm aspect: a review of the aspect of the proposed AIS algorithm including modifications to the algorithm;
- representation: including the scheme for antibody representation and antibody recognition area;
- affinity: comparing the distance or affinity functions of an existing immune inspired IF and
- limitations of the work.

Tabular summaries of population-based immune inspired IF systems are given in Appendices A.1 to A.4. For network-based immune inspired IF, readers are

directed to Appendices B.1 to B.4. Due to the large number of studies of immune inspired IF, this review is necessarily incomplete, but we hope that it will be a good resource for the application of AIS to the IF domain.

The next issue for attention in this section is to identify the limitations of the existing immune-inspired IF in the problem of profile adaptation. Based on the existing literature, most of the immune inspired IF is mainly concentrated on e-mail filtering (including SPAM filtering), document classification, and recommendations on collaborative filtering. The role of AIS in regard to the problem of profile adaptation has not been fully explored, except in [44]. The limitations relating to profile adaptation in the existing immune inspired IF studies is identified below.

1. User profiling follows the metaphor of an immune network, with the terms that represent a user's multiple interests constructed as a hierarchical network. The ability of immune properties to maintain sufficient diversity and adaptability is not fully explored.

2. Profile adaptation is treated implicitly through iterative learning of user interest.

3. In collaborative filtering, the existing immune inspired approaches focus on solving the sparsity problem. Profile adaptation on collaborative filtering has not been a focal point of study.

The viability of AIF system relies on the ability of the user profile to maintain a satisfactory filtering accuracy for as long as it being used. The user profile has to be able to represent the complete range of a user's interests and to continuously adapt, in response to user feedback, to any changes in them. As the user interests and the information environment change, new terms are required to cover new topics of interest, while it is at least impractical to maintain in the profile terms that no longer reflect the user's interest. Existing approaches to IF do not fully comply with these mentioned requirements, mainly because they typically rely on the Vector Space Model (VSM). Both documents and the user profile are represented as keyword vectors in a space with as many dimensions as the unique words in a predefined vocabulary. The problem is that keyword vector representation inherently ignores correlation between words in text [11]. For the immune inspired IF, vector-based representation has been generally the *de facto* choice in the domain of AIS [103]; for example, binary keyword vectors used to represent document and immune receptors [9], representation of antibodies as

weighted keyword vectors [104] and weighted keyword networks for profile representation [44]. In this work, we are inspired by the biological immune systems to build an adaptive user profile that can continuously maintain a representation of the developing user interests within a changing information environment. In contrast, AIS have the inherent ability to boost and maintain the diversity of the immune repertoire achieved through the preservation of diversity (*heterostasis*) and the introduction of diversity (*heterogenesis*). To further identify characteristics or properties lying in the biological system (that is the immune system) and the application domain, we follow the principled meta-probes suggested in [5]. A discussion of this principled abstraction follows in the next section.

3.4 Principled Meta-Probes Applied to Adaptive Information Filtering (AIF) as a Source of Immune Inspiration

As a first step towards identifying appropriate characteristics of the immune inspiration and the application domain, we follow a proposal of Stepney et al. [5], that the underlying properties of classes of model can be analysed at a higher level known as 'meta-questions'. We follow the questions that address notions such as Openness, Diversity, Interaction, Structure and Scale, otherwise known as the *ODISS* meta-probes. The ODISS meta-probes are used to challenge the biological system (as in [5, 83]) and the application domain, to identify matching characteristics and ODISS properties. The results of this high-level abstraction are used as part of the basis for building a biologically-inspired application – here, an immune inspired AIF application. In Section 3.4.1 and Section 3.4.2, we summarize the results of applying these principled meta-probes to both the natural immune system and the target AIF domain.

3.4.1 ODISS Meta-Probes applied to Immune Systems

Applying the ODISS meta-probe analysis to the immune system is not only challenging, but will most certainly be incomplete. Therefore, in this section, we merely seek to demonstrate the principle characterization of the immune system and to shed some light on how the immune system can be considered. Each of the five meta-probes is considered in turn.

Openness

The mammalian immune system is open in the sense that it is continually evolving (on multiple time scales) and replenishing itself through the continual production of immune cells. Continual evolution is needed for the immune system to remain effective in the face of changing pathogen exposure and the evolution of viruses. A widely-accepted characteristic is the two levels of evolution – the innate immune system that evolves across mammal generations, and the adaptive immune system that evolves within the host individual [4,105]. Furthermore, the immune system is maintained in a dynamic equilibrium (or *homeostatic* state), in which it must respond to a diverse array of microbes, despite its constant exposure to self-antigens [106]. The influential *clonal selection theory* postulates a primary immune response mechanism that is then optimised through increases in antibody affinity, to the secondary response – the immune system supports continual replenishing of resources in terms of reproduction of new cells, self-maintenance of the system [62, 63] and cells death (apoptosis) [107].

In the *immune network theory*, the immune network is a basis for the regulatory mechanism that maintains homeostasis, through the continuous production and recruitment of immune cells and molecules [87]. The meta-dynamic of the immune network (or immune recruitment mechanism) allows the addition of new elements into the network to extend or adapt coverage of the space of antigens [4].

Diversity

Diversity is a hallmark of the immune system, and a key feature in the ability to recognise and react to a continuously changing environment. Lymphocyte repertoires with millions of different specificities function in concert with diverse cytokines, chemokines and different types of antigen-presenting cells [108]. Different pathogens are handled by qualitatively different immune responses such as cellular and humoral responses. At the same time, most immune responses against self peptides and antigens are avoided. The polymorphism of major histocompatibility molecules involved in antigen presentation means that different individuals in a population may respond differently to identical antigens [108].

The diversity of the adaptive immune system is maintained through *heterostasis* that is the preservation of diversity in which cells are selected to clone or to become memory cells according to antigen affinity. Furthermore, diversity can be changed or extended through *heterogenesis* that is the introduction of diversity, either through somatic hypermutation or through the recruitment of new cells and the suppression of similar antibodies [4]. In the innate immune system, the innate

repertoire is naturally diverse, comprising different cells with different functions (for example natural killer cells, macrophages, dendritic cells) [108].

In addition to the maintenance of component diversity, a range of mechanisms is involved in developing specificity of response: these include degeneracy and pleiotropism. Cohen [109] described antigen receptor degeneracy as the "capacity of any single antigen receptor to bind and respond to (recognize) many different ligands", whilst Edelman and Gally described how degeneracy supports adaptability, in terms of structurally different elements that yield the same or a different function depending on the context in which they are expressed [110]. Pleiotropism relates to the ability of, for instance, effector molecules (any cytokine, chemokine or other cell-interaction molecule) of the immune system to produce different functional effects in many different cell types, or sometimes even in the same cell type [109]. Thus, in brief, diversity is underpinned by different types of immune system component that have a similar role, and similar types of immune system component that have different roles.

Interaction

As in other natural complex systems, there is a wide range of interactions in the immune system, forming dialogues among 'agents' and with the host environment. A few of the typical interactions are summarised as follows.

- Intercommunicating tissue cells are a feature of the clonal selection theory, which postulates an immune system composed of discrete sets of elements that are compared with the environment. If there is an explicit antigenic population to be recognized, all or some antigens can be presented. This scheme of direct interaction with the environment supports a reinforcement learning strategy [111].

- Multiple interacting immune agents such as macrophages, T and B cells are involved, for instance, in immune correspondence [63, 109, 111].

- Networks of signaling molecules support immune agent communications in an immune dialogue [62, 109].

- Interaction between the innate and adaptive immune systems allows the initial pathogen attack to be handled by the innate immune system, in a response that alerts the adaptive immune system to the pathogen invasion [4].

Structure

The structure (architecture) of the immune system is multi-layered, with defences on many levels (anatomic barriers, innate and adaptive immunity, and so on). Some structures are well-understood, but the full structure and its purpose are still the subject of debate. However, what is of interest is the way in which structures are necessary for host protection. Why is it that, in some organisms, only an innate immune system is required rather than a combined innate and adaptive system? What are the implications of such structural differences on the overall system? Furthermore, the immune system has a *double plastic structure*; plasticity means that the immune system can adapt basic components and structure. Double plasticity means that both individual cells and connections are constantly being added to and removed from the network [4].

Scale

Scale is partly a function of the structural layering of a complex system. It is also a factor of quantities, and is included in the ODISS meta-probes to remind the principled designer of bio-inspired algorithms that nature counts in billions, not tens, of components, interactions, and so on. It follows from the above sections that an immune system contains a considerable number of different types and functional variants of component. It is also the case that the natural immune system has large (and variable) populations of each type of component. The adaptivity and maintenance of a dynamic equilibrium, are hypothesised to depend on the quantity and diversity of components and component behaviours, as much as on simple actions of the immune system. Another important aspect of scale is the ability of the immune system to react to small amounts of new antigen, using cloning to amplify the effectiveness of the response.

3.4.2 ODISS Meta-Probes applied to Adaptive Information Filtering (AIF)

The ODISS meta-probes have been applied to various biological subjects, but are not normally applied to application domains. Here we apply the ODISS approach to the AIF domain in order to highlight the features that would be necessary for an ideal AIF system.

Openness

AIF systems are open systems in that they must be always acting and changing, maintaining themselves through a continuous interchange with their environment. The openness of an AIF is arguably more limited than that of the immune system: in AIF systems, there are two main inputs – user profiles and the data stream. New or changed profile features or new data themes can arise at any time, but completely new forms of input are typically not within the remit of these systems. For an AIF system, typical consequences of openness are the ability to adopt new goals at run time, to self-reconfigure and to self-reorganize components. Profile adaptation due to changes of interest and multiple interests of users in relation to the data stream involves a flow of information to and from the system environment, as well as between the components of the system. In terms of openness, this information flow is used to cause the system to adapt (evolve) during computation. An ideal AIF system would be permanently evolving, permitting changes to profiles and data flows while the filtering activities continue. In practice, the extent of this evolutionary openness depends on the speed of adaptation which is appropriate to a particular AIF application.

Diversity

For an AIF system, the content and semantics of data streams, and the relevance of data items to users, changes over time. To handle this, the AIF system needs diversity. The data stream input to an AIF system is processed into some suitable representation that identifies terms (keywords, ontologically-similar phrases and so on); the terms are the basis for matching to user interest. In AIF, a good representation would have to handle a significant diversity of data items, within a data stream and over time. Representation schemes need to be appropriate and adaptable, to maximize the retrieval or the filtering result.

The AIF system selects data items for output to a user by relating the terms in the representation of the input data stream to the user's needs, typically expressed either as direct requests (user queries) or as a (dynamically-updated) user profile. The representation used should permit the AIF system to determine and return a set of data items which gives optimal coverage of the information space of the user's request or profile. Good coverage of the information space improves the chance of a user being satisfied, but also improves the quality of feedback. User feedback (direct, or through monitoring of what users do with the output data) is the basis for adaptation of the user profile, which is how the system can adapt its response to changing user needs. This aspect is particularly important if

the user is not satisfied with the recommended data items, or if there is a change in the interests, and thus the reactions, of a user.

Interaction

Interaction of an AIF system includes interaction with environment (the user and the data) and interaction between system components (for example, an agent). Adaptation in the AIF system is the result of dialogues between the user and the AIF system. In some cases, dialogues are explicit (user queries, direct editing of profile elements), but a more responsive approach is through implicit dialogues, using deduction from interaction monitoring to revise user profiles. For an effective AIF system, the users have to perceive that they can rely on the filtering result and can hand over control to the system. Furthermore, the user has to keep using the system, so that the system can learn and adapt, even when the result is not optimal. This raises challenges relating to the management of user perception, achievement of appropriate user control and trust, and the design of user interaction and response monitoring.

For interaction among components, adaptivity is closely related to the capacity of agents to interact with their environment and with each other, both directly and indirectly. Keil and Goldin [112] defined direct interaction as the exchange of data over time between computing agents or between an agent and its environment, for instance interaction via messages, where the destination agent is specified in the message. Indirect forms of interaction rely on *the persistence* and *observability* of changes in the environment. Keil and Goldin [112] argued that indirect interaction involves a time delay (decoupling) and does not rely on agents and observables sharing a location (in space or time).

Structure

A typical AIF system is an on-line system, whose durability and environment are not pre-determined; adaptivity requires the system to have a dynamic structure, for instance within the filtering and learning components. Such flexibility again supports the use of an agent-based approach for these components [113]. Continual interaction between learning and filtering is essential. This enhances filtering effectiveness as information needs and input data streams evolve over time.

Scale

An AIF system that maintains itself through continuous interaction with its environment needs a significant number of 'agents'. It is an open question of complex

systems as to how many agents are necessary to achieve the behaviour of the system, and how quantity interrelates with diversity. Furthermore, an AIF system faces problems of sparsity and scale. For instance, a recommender system may have to cope with a significant part of the world-wide web (and its phenomenal rates of growth and change). In addition, users are many and varied in their profile characteristics, interests and interaction habits. User preference may have to be deduced from only a small number of instances of feedback for each user interest or each data item, and there is much variety of behaviour across users [114]. Scale thus raises issues of the quantity and quality of data, and of the quality and scope of representation for both data streams and user characteristics. The problems would seem to bear comparison with immune systems, facing continuous exposure to many diverse new antigens, many of which appear first in small numbers.

3.4.3 ODISS Meta-Probes Mapped to Immune Inspired Adaptive Information Filtering

Table 3.3 is a concise comparison of our ODISS meta-questions on the AIF domain and shows comparable features of the immune system and immune theories.

From the mapping, we can summarize that both AIF and the immune systems are open, and whilst an AIF system is arguably less open (a software system has less scope for receiving entirely novel inputs or generating entirely novel responses), the AIF shares the need for continual evolution to produce effective adaptation to small changes in a very large range of inputs. Again, the AIF and the immune systems have common characteristics of diversity, and we can take inspiration from immune system properties such as degeneracy and pleiotropism in finding effective ways for filtering and learning to adapt to changes in inputs. The immune systems and the AIF shared similar characteristics in terms of interaction, whereby for an immune system it forms a dialogues among the 'agents' and the host environment while, for an AIF system the interactions can occur internally (among systems component) and externally (among users and the system). From the mapping, both AIF and the immune systems shares common characteristics of structure. For the immune systems, it has a *double plastic structure* where both individual cells and connections are constantly being added to and removed from the network. An effective AIF system requires the structure to be dynamic within the filtering and learning components as information needs and input data streams evolve over time. Finally, for scale, the immune system has a large (variable) population of each type of component and it has the ability

Table 3.3: ODISS characteristics of AIF as a source of immune inspiration [1]

ODISS	**Adaptive Information Filtering**	**Potential immune system inspiration**
Openness	Adaptation to changing data and user profile	General features of continual evolution, replenishment, and addition of resources
Diversity	Need to respond to diverse and non-specific changes; diversity of user profiles across the user population, and of data items across the input data stream	Immune system's ability to boost and maintain diversity achieved through the preservation of diversity (*heterostasis*), and the introduction of diversity (*heterogenesis*); properties of degeneracy and pleiotropism
Interaction	Need for flexible interactions both between user and system and among systems components	General features of immune dialogue; context-dependent interaction
Structure	Structure of data, thematic connectivity	*Double plastic structure* – the immune system can adapt basic components and structure; both individual cells and connections are constantly added and removed
Scale	Need to respond efficiently in the face of very large and changing data streams, and sparse evidence of changing data and user characteristics	Responsiveness to small amounts of new antigen; amplification through clonal selection

to react to small amounts of new antigen based on cloning to amplify the effectiveness of the response. For an AIF system, it needs to response effectively in the problems of large and changing data stream in the world-wide web and sparsity of changing user characteristics.

3.5 The DCS: Dynamic Clonal Selection

Section 3.4.1 and Section 3.4.2 contained discussions of how the ODISS can be used to identify common features of the immune system and the AIF domain. The ODISS approach leads to a comprehensive set of requirements which allow the identification of an appropriate property of the immune system and the problem domain studied. This is the start of a principled abstraction of how adapting

a user profile in IF might take inspiration from aspects of the immune system. From the studies reviewed, this work suggested that profile adaptation can be developed by incorporating ideas from aspects of dynamic clonal selection (DCS) with the use of gene libraries to maintain sufficient diversity. DCS has been identified as an AIS algorithm that supports learning in a dynamically changing environment [91, 115–118]. The following section presents the principle of clonal selection, including the gene libraries, in order to provide an understanding of this immune inspiration.

3.5.1 The Biological Inspiration of Clonal Selection

The clonal selection principle was introduced by Burnett [86] and described the basic features of an immune response to an antigenic stimulus. It establishes the idea that only those cells that recognize the antigen proliferate, thus being selected against those that do not. Figure 3.7 provides an overview of the clonal selection process. The main features of the clonal selection theory are that [86]:

1. The new cells are copies of their parents (clones) subjected to a mutation mechanism with high rates (somatic hypermutation)

2. There is elimination of newly differentiated lymphocytes which carry self-reactive receptors.

3. There is proliferation and differentiation on contact of mature cells with antigens.

The development of the clonal selection algorithm was proposed by Castro & Zuben [8]. The main immune aspects considered in the development of the algorithm are: maintenance of the memory cells functionally disconnected from the repertoire, selection and cloning of the most stimulated cells (number of clones proportional to affinity), death of non-stimulated cells, affinity maturation and re-selection of the clones with higher affinity, generation and maintenance of diversity and the hyper-mutation inversely proportional to the cell affinity. Later on, the CLONALG algorithm [119] was developed to solve multimodal function optimization problems. This algorithm has several interesting features [119]:

1. the population size is dynamically adjustable;

2. there is exploitation and exploration of the search space;

3. it has the capability of maintaining local optima solutions;

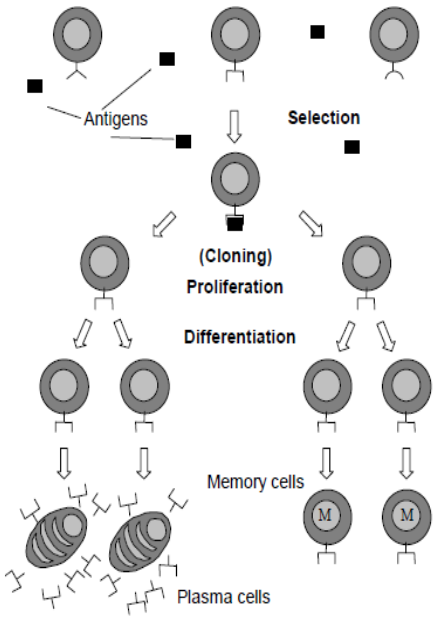


Figure 3.7: The Clonal Selection Principle [7,8]

Another variation of clonal selection was introduced by Kim and Bentley [120] and is known as 'DynamiCS'. This algorithm was designed for use in a computer security scenario, where the threat to computers on a network will continuously change. In particular, *DynamiCS* is based on Hofmeyr's [66] idea of the 'dynamics' of three different populations: immature, mature and memory detector populations. Initial immature detectors are generated with random genotypes. Using negative selection, new immature detectors are added to keep the total number of detectors constant after a predefined number of generations (polarization period, T). If a detector is within its predefined life span L, and the match counts are larger than a predefined activation threshold A, it becomes a memory detector. Mature detectors are used to identify unknown attacks. In this way, the DynamiCS learns normal behaviour by observing only a small set of self-antigens at any one time. Detector cells will be replaced whenever normal behaviours change. However, the DynamiCS was found to be slow to react to changes, and a sharp change in self behaviour resulted in a high false positive rate. This outcome was due to the memory detectors not being exposed to the entire set of self-patterns during toleration, a situation also present in the natural system. Kim and Bentley [12] then introduced an extended DynamiCS which had the added mechanism of removing memory detectors when they showed a poor degree of self-tolerance. This was shown to reduce the high false positive rate, but at the expense of requiring a larger amount of co-stimulation (user interven-

tion) to achieve this. This work was further augmented by [91] and [116] by the addition of a hyper-mutation operator to produce the effect of gene library evolution. Rather than new detectors being generated randomly, new detectors were produced by mutating deleted detectors. Thus, a 'virtual gene library' made from mutations of deleted memory detectors was maintained. The test results showed that this scheme produced immature detectors that were better suited to cover existing non-self antigens. The pseudo code Algorithm 2 provides an overview of DynamiCS. Further explanation of the algorithm can be found it [12].

From the explanation on clonal selection and an example of existing application based on dynamic clonal selection, we believed that the use of gene libraries could produce reasonable coverage of detectors to detect changes of profile in varying incoming documents. Gene libraries are a "biological mechanism for generating a combinatorial diversity in the immune system" [117]; they are shaped by evolution to create detectors that preserve the ability to respond to novel threats [117,118]. In fact, gene libraries are often thought of as a biological mechanism for generating combinatorial diversity of antibodies. Furthermore, through dynamic clonal selection, it can inherently maintain and boost diversity and can dynamically control the size of the immune repertoire by means of selection, cloning and mutation procedures. Moreover, diversity in the population is enabled by means of the receptor editing process. Further implementation of the proposed approach which include the process, issues that arise in the design of the algorithm and the experiment is presented in Chapter 6.

3.6 Summary

Bio-inspired algorithms have been commonly applied to complex problems including those that deal with adaptivity to dynamic environments. Within the domain of bio-inspired algorithms, artificial immune systems (AIS) have been actively explored in the problem of information filtering (IF). This chapter reviewed the potential of AIS applied in the domain of IF. To provide some background on AIS, the first part of the chapter reviewed the biological perspective of the immune system which included the immune system components, the structure of the immune system and the immune response. Some definitions of AIS and the chronology of AIS development is presented in Section 3.2 to provide a basic understanding of AIS. To build an effective immune-inspired engineered system, there is a need for attention to the underlying immune system and the development of an abstract model to understand the computational properties of the

immunological inspiration. Therefore, Section 3.2.1 focused on the existing AIS frameworks namely, the conceptual framework, the layered framework and the immuno-engineering framework. These frameworks will be used as a guideline in the process of moving from immunology to engineered systems. As part of the ongoing work on immune-inspired IF, this thesis focuses on profile adaptation in IF. Therefore, Section 3.3 reviewed the current state of immune inspired IF. From the review, it was identified that most of the immune inspired IF is much more concentrated on e-mail filtering (including SPAM filtering), document classification and recommendations on collaborative filtering. From the literature on immune inspired IF, we have identified some of the limitations relating to profile adaptation in the existing literature and it is discussed in this section. Before we can decide on suitable immune inspiration for our problem domain, we need a principled guideline to identify appropriate characteristics of the immune inspiration and the application domain. Therefore, we followed a proposal of Stepney et al. [5] whose work suggested meta-questions that address notions such as Openness, Diversity, Interaction, Structure and Scale, otherwise known as the *ODISS* meta-probes. A discussion of these meta-probes is presented in Section 3.4. This ODISS approach leads to a comprehensive set of requirements which allow to identify an appropriate property of the immune system and the problem domain studied. From the principled meta-probes, the discussion focused on the AIS algorithm called the Dynamic Clonal Selection Algorithm (DCSA), which was specifically developed for a range of adaptivity in dynamic environment problems that is, adaptive information filtering (AIF). Therefore, relevant biological inspiration are also presented in Section 3.5. To continue the investigation of approaches to AIF in the context of changing user interests, the next chapter will present the platform and context in which the experiments will be carried out.

```
begin
    Initialise Dynamic Clonal Selection Algorithm
        Create an initial immature detector population with random
        detectors ;
    Generation Number = 1 ;
    while (Generation Number < max Generation) do
        if (Generation Number = N) then
            Select a new antigen cluster;
        end
        Select 80% of self and non self antigens from chosen antigen
        cluster;
        Reset Parameters:
            Generation Number++;
            Memory Detector Age++;
            Mature Detector Age++;
            Immature Detector Age++;
        Monitor Antigens:
        {
            Monitor Antigens by Memory Detectors:
                Check any memory detector detects any non-self antigen ;
                Check any memory detector detects any self antigen ;
            Monitor Antigens by Mature Detectors:
                Check any mature detector detects any non-self antigen ;
                Check any mature detector detects any self antigen ;
                Create new memory detectors ;
                Old mature detectors are killed ;
            Monitor Antigens by Immature Detectors:
                Check any immature detector detects any self antigen ;
                Delete any immature detector matching any self antigen ;
                Create new mature detectors ;
                }
            if (immature detector population size + mature detector population
            size < non memory detector pop size) then
                Do
                    { Generate a random detector ;
                      Add a random detector to an immature detector
                      population ;
                    } Until (immature detector population size + mature
                    detector population size = non memory detector pop
                    size);
            end
    end
end
```

Algorithm 2: The Algorithm for DynamiCS [12]

CHAPTER

FOUR

EXPERIMENTING WITH ARTIFICIAL IMMUNE SYSTEMS FOR INTEREST CLASSIFICATION

AIS have been applied to email classification for some years based on the dynamic nature of the immune system. E-mail classification is chosen because of its characteristic of having properties of dynamism and diversity in identifying user interests in e-mails. In the e-mail environment, the topics which a user may be interested in are liable to drift over time. The ability of an algorithm to keep track of these changes in the application domain is very important in such a filter. In addition, the immune system operates in an ever-changing environment. The immune system constantly has to tackle antigenic signatures which it has never seen before while those it has seen may change and adapt over time. The dynamic nature of the immune system can be capitalised on the domain of e-mail classification.

This chapter describes the development of Artificial Immune System (AIS) algorithms in a text-mining scenario. The purpose is to investigate the AIS algorithms in classifying emails based on user interest with regard to both single and multiple email topics. The task of email classification is widely used as an experimental platform for exploring the effect of changing user interests. The experiment on AIS in email classification is based on Secker's algorithm [9], which is an artificial immune system for email classification (AISEC) which classifies emails as interesting or uninteresting according to the subject and sender of the email. A detailed description of AISEC is presented in Section 4.1. Our inter-

est is in experimenting with AIS in a changing-interest scenario and classifying multi-topic emails. An extension of the AISEC algorithm was developed and is described in Section 4.2. The remaining sections of this chapter present an analysis of the experiment: Section 4.3 presents the results of the experiment on email classification based on a binary classification problem (discriminating between interesting email or not interesting email) and Section 4.4 presents the results on email classification when there are multiple topics, for instance, two-topics, three-topics and four-topics emails. This chapter ends with a summary in Section 4.5.

4.1 An Overview of the Artificial Immune Systems For Email Classification (AISEC) Algorithm

The task of email classification is widely used as an experimental platform for exploring the effect of changing user interests. Secker et al. [9] developed an artificial immune system for email classification (AISEC) which classifies emails as interesting or uninteresting according to the subject and sender of the email. The AISEC algorithm removes uninteresting email from a user's inbox. The system has been shown to be capable of continuous learning, adapting to changes in a user's interests. The AISEC system [9] uses inspiration from the behaviour of B cells in the immune system to remove uninteresting emails from the system. In the immune system, there is a set of naive B cells and a set of memory B cells; when a naive B cell meets an antigen, it is stimulated, and becomes a memory B cell or become a plasma cell. In the algorithm, B cells have a feature vector. The feature vector is populated with words from the email subject and sender fields (see Figure 4.1) – the training phase of the algorithm populates the feature vector of naive B cells. In the task of email classification, both naive and memory B cells represent examples of words from uninteresting e-mails.

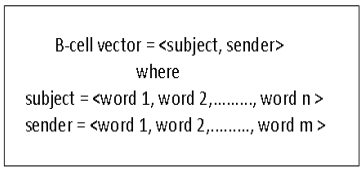


Figure 4.1: Structure of the B cells vector

After training, the algorithm processes emails as they arrive, treating the email as an antigen. The words extracted from the email subject and sender fields are

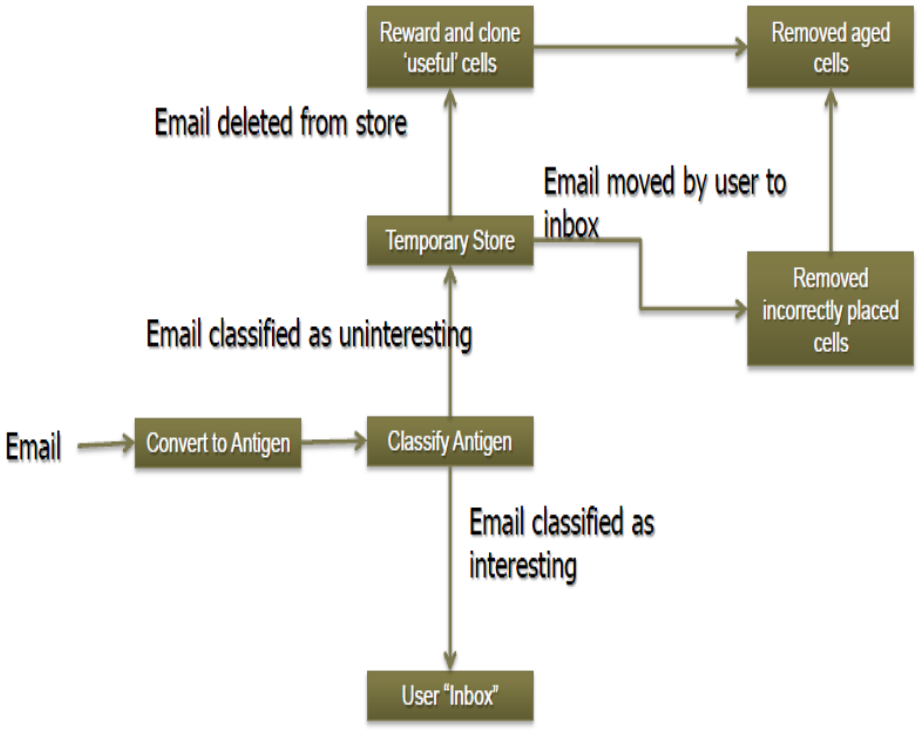


Figure 4.2: High Level View of the AISEC system after Initialisation, [9, 10]

compared with the feature vectors of existing naive and memory B cells. The degree of matching is calculated as an *affinity measure*, which is compared with a preset *affinity threshold*. If the B cell's affinity exceeds the threshold, then this B cell is said to recognise that email antigen, which is a candidate uninteresting email. At this point, user feedback is needed to determine whether the email is in fact uninteresting. A confirmation results in a reward to the B cell, and this may result in cloning of this B cell. Cloning produces another B cell which may be mutated, according to a predefined mutation rate which is inversely proportional to the affinity with the antigen. Mutation involves the replacement of one word from the feature vector. Finally, in order to prevent unlimited growth in the population of memory B cells, a cell-death process is implemented in which cells which have not received sufficient stimulation over a period of time are purged from the system. Because the algorithm handles a stream of emails, and regular user feedback, the algorithm is dynamic. Figure 4.2 shows a high-level view of the AISEC system for classifying an email.

The immune inspiration and the process involved in the AISEC algorithm have been explained. Our attention turns next to describing in greater detail the algorithm, including its parameters. The discussion which follows is a summarisation based on [9]. The pseudo codes in Algorithm 3 until Algorithm 7 were

also taken from [9], with some minor variations in their text.

4.1.1 Algorithms and Processes

AISEC is a population-based algorithm consisting of two distinct stages: a training initialisation phase followed by a running phase. This running phase is further divided into two tasks, that of classifying new data and that of intercepting user feedback to drive evolution. The entire algorithm is shown in Algorithm 3, where TE denotes the set of initialisation (training) examples and ag denotes an antigen (a processed e-mail of unknown class). During the initialisation stage (Algorithm 4), the goal is to populate the gene libraries, produce an initial set of memory cells from initialisation examples, and produce naive B cells based on mutated initialisation examples. In the initialisation stage (Algorithm 4), Ksm denotes the initial of stimulation count for memory B cells, while Ksb denotes the initial stimulation count for naive B cell and Ka denotes the affinity threshold. Once the system has been initialised, it is available to begin to perform two distinct functions; the classification of unknown e-mails and the population update processes based on user feedback on the correctness (or error) of classification attempts. During this phase, the algorithm will wait for either a new e-mail to enter the system and be classified, or an action from the user indicating feedback. Upon receipt of either of these, the system will invoke the necessary procedure as in either Algorithm 5 or Algorithm 6. When feedback from the user is received, a co-stimulation signal for a B cells is activated. At this stage, the useful B cells are stimulated (in the sense of correctly classifying an email), and unstimulated B cells are removed from the system. ag is the antigen (e-mail) on which feedback has been given. Algorithm 6 shows this process in detail.

Naive cells with the highest affinity to the e-mail are cloned. The affinities of the clones with the e-mail are then determined and if one of the naive cells (clones or pre-existing cells) is found to have an affinity with the e-mail greater than a pre-existing memory cell, then that naive cell is promoted to a memory cell. On the other hand, for an incorrect classification, the opposite happens. All cells with affinities with the misclassified e-mail over a certain threshold are detected and removed. The process of cloning and mutation is detailed in Algorithm 7.

In Algorithm 7, Kl and Km are constants used to control the rate of cloning and mutation. The symbol "$\lfloor x \rfloor$" denotes the "floor" of x. That is, the greatest integer smaller than or equal to the real-valued number x. This operator is

```
PROGRAM aisec
begin
    train (training set)
    Wait until (an e-mail arrives or a user action is intercepted)
    ag ← convert e-mail into antigen
    if ag requires classification then
        classify(ag)
        if ag classified as uninteresting then
            move ag into user accessible storage
        else
            allow e-mail to pass through
        end
    end
    if user has given feedback on ag then
        update_population(ag)
    end
end
```

Algorithm 3: AISEC overview

```
PROCEDURE train(TE)
begin
    foreach te ∈ TE do
        Process e-mail into a B cells
        Add subject words and sender words to appropriate library
    end
    Insert Kt processed e-mails into MC, selected at random from TE
    foreach mc ∈ MC do
        mc's stimulation count ← Ksm
    end
    foreach te ∈ TE do
        te's stimulation count ← Ksb
        foreach mc ∈ MC do
            if affinity(mc,te) > Ka then
                clones ← clone_mutate(mc,te)
                foreach clo ∈ clones do
                    if affinity(clo,bc) ≥ affinity(mc,te) then
                        BC ← BC ∪ clo
                    end
                end
            end
        end
    end
end
```

Algorithm 4: Initialisation

```
PROCEDURE Classify(ag)
begin
    foreach bc ∈ (BC ∪ MC) do
        if affinity(ag,bc) > Kc then
            classify ag as "uninteresting"
            RETURN
        end
    end
    classify ag as "interesting"
end
```

Algorithm 5: Classification

necessary because num_clones and $num\_mutates$ must both be integers. In the Algorithm 7 procedure, the input $bc1$ is first cloned num_clones times then each clone is mutated $num\_mutates$ times by picking a point in the clone's feature vector and replacing the word found in that point with another suitable word pulled from the required gene library.

4.1.2 Parameters of the AISEC Algorithm

In previous section, the process and algorithm description of the AISEC were discussed. The dynamic behaviour of the algorithm can be controlled by the algorithm's parameters. Therefore, in this section the list of the AISEC's parameters and their context are discussed. These parameters will also be examined in terms of the algorithm's behaviour in sensitivity analysis. The experiment and the results of the sensitivity analysis are discussed in Chapter 5. The following are the major parameters used in the AISEC and the extended AISEC version. The following description is a summarisation based on [9]:

1. Kc **(classification threshold)**. The classification threshold influences the decision on the class of the incoming emails. If an antigen (an email) shows affinity for any immune cell higher than this threshold, the email is classified as uninteresting. Therefore, the classification threshold defines the algorithm's tolerance level to identify uninteresting email. A low Kc level may allow a low affinity antigen to be classified as an uninteresting email (a negative classification). However, if the tolerance level is set too high, there might be a danger that the immune cells might negatively class antigens (false positive classification).

2. Ka **(affinity threshold)**. The affinity threshold influences three different

```
PROCEDURE update_population(ag)
begin
    if classification was correct then
        foreach bc ∈ BC do
            if affinity(ag,bc) > Ka then
                Increment bc's stimulation count
            end
        end
        bc_best ← element of BC with highest affinity to ag
        BC ← BC ∪ clone_mutate(bc_best,ag)
        bc_best ← element of BC with highest affinity to ag
        mc_best ← element of MC with highest affinity to ag
        if affinity(bc_best,ag) > affinity(mc_best,ag) then
            BC ← BC \ bc_best
            bc_best's stimulation count ← Ksm
            MC ← MC ∪ (bc_best)
            foreach mc ∈ MC do
                if affinity(bc_best, mc) > Ka then
                    decrement mc stimulation count
                end
            end
        end
        add words from ag's feature vector to gene libraries
    end
    else
        foreach bc ∈ (MC ∪ BC) do
            if affinity(bc,ag) > Ka then
                remove all words in bc feature vector from gene libraries
                delete bc from system
            end
        end
    end
    foreach bc ∈ BC do
        decrement bc's stimulation count
    end
    foreach bc ∈(MC ∪ BC) do
        if bc's stimulation count = 0 then
            delete bc from system
        end
    end
end
```

Algorithm 6: Update B cells population

```
PROCEDURE Clone_mutate(bc1,bc2)
begin
    aff ← affinity(bc1,bc2)
    clones ← ∅
    num_clones ← ⌊aff × Kl⌋
    num_mutate ← ⌊(1 − aff)× bcx's feature vector length ×Km⌋
    DO num_clones TIMES
        bcx ← a copy of bc1
        DO num_mutate TIMES
            p ← a random point in bcx's feature vector
            w ← a random word from the appropriate gene library
            replace word in bcx's feature vector at location p with w
        end
    end
    bcx's stimulation level ← Ksb
    clones ← clones ∪ bcx
    Return clones
end
```

Algorithm 7: Cloning and mutation

processes: selecting B cells that will receive rewards, selecting memory cells to lower their stimulation level and removing cells when false positive classification occurs. A low Ka value increases the chance of generating more memory B cells in the initialisation phase, which may increase the population size. However, as more classifications of emails as uninteresting occur, a low Ka value may allow an increased number of cells to be removed from the population, because more of these classifications will be wrong, thus leading to a reduction in cell population. In contrast, high Ka values allow only high affinity cells for the misclassified antigens to be removed from the algorithm. This might encourage the population to grow in the long run, resulting in an increase in the false positive rate.

3. Kl **(a constant which controls the rate of cloning**). This parameter determines the maximum number of clones a cell may produce. It applies only once a true positive classification has occurred and applies to the cell showing highest affinity amongst all B cells and memory cells combined. High Kl may result in an increase in naive B cells population sizes as more clones may be generated. This might increase redundancy because clones are mutated duplicates of the most competent cell, and the mutation applied in some cases may be insufficient to move the clone outside the range of affinity threshold required to classify an antigen.

4. Km **(a constant which controls the rate of mutation**). The Km parameter is responsible for the rate of the mutation. High Km might result in an increase of diversity in B cells by generating more variants from selected cells. Diversity of B cells affords the algorithm the ability to recognise unknown e-mails. However, if Km is too high, the mutated clones will tend to have a low affinity for the antigen.

5. Ksb **(an initial stimulation count for naïve B cells**). This parameter influences the potential life-span of a naive B cell. High Ksb can allow the non-stimulated B cells to survive longer, which results in an increase in the overall cell population size. In this case, allowing the non-stimulated cells to live too long will also allow the false positive rate to increase. Upon a false positive classification, all B cells are suppressed by reducing their stimulation level. However, in addition to this, the cells matching a misclassified antigen will be removed. Therefore, a high rate of positive classification and a low false positive rate will be the ideal combination for drastic cells death.

6. Ksm **(an initial stimulation count for memory B cells**). This parameter influences the potential life span of memory B cells. High Ksm allows memory cells to survive longer, but low Ksm can lead to an increase in cell population. But in this case, it is possible that allowing the memory cells to live longer may allow the false positive rate to increase because more e-mails may be recognised and classified as uninteresting messages by old memory cells. Therefore, the influence of Ksm is considered to be more prominent with a low Kc than with a high Kc.

7. Kt **(an initial number of memory cells generated during initialisation)**. The Kt parameter determines how many memory cells will be selected from the initialisation data set in the initialisation phase. If the Kt is a low value, then it is more likely that the affinity function will return a value less than Ka for all cells in the initial MC population. In this case, no clones will be produced for the current antigen and there will be no B cell recognizing that specific antigen. Therefore, a high Kt can guarantee that more B cells are produced during the initialisation, compared with a low Kt value.

4.2 The Extended Version of the AISEC Algorithm

Work on the AISEC algorithm has attracted researchers to carry out further experiments on email classification. For example, Prattipati and Hart [10] modified the

AISEC algorithm to improve the speed of adaptation and the overall accuracy of the classification algorithm. Their work experimented using the original AISEC to explore whether explicit changes in a user's interests could be tracked. In their experiment, the AISEC adapted quickly to a change in interest from Junk to Inbox. However, an obvious decrease in classification accuracy was found when the reverse process occurred: when the user lost interest in a topic, the AISEC failed to react. In addition, they also modified the original algorithm by replacing the mutation vector originally used in [9] with a position-biased mutation operator proposed by Kelsey and Timmis [121].

In the previous AISEC [9, 10] the algorithm extracted words only from the email subject and sender fields. This limits the potential diversity of the *gene library* (the set of all words from feature vectors of B cells that recognise uninteresting emails). This in turn reduces the diversity generated by cloning and mutation, since, when mutation is performed, a word from this library replaces a word from a cell's feature vector. In order to recognise new topics of interest and enable the removal of existing topics of interest, we need a large library of words, and we need to be able to detect synonyms. There are several ways to increase the diversity of words in the gene library. Our first modification to the AISEC was to consider the body of the email in addition to the email subject and sender field, as this provides a richer set of words. Furthermore, we used the WordNet corpus as a source of different types of relationship. This allowed more accurate classification of emails, and improved the capability to identify new and potentially uninteresting e-mails. A fuller description of the use of the WordNet hypernym-hyponym relationship is presented in Section 4.2.1.

Furthermore, to show that the extended AISEC is capable of continuous learning, and of potentially tracking changes in email topic, we carried out an experiment to verify whether explicit changes in a user's interests could be tracked. The experiment was conducted in two scenarios; binary classification (discriminating between interesting or not interesting email) and multiple email topics. Full descriptions of these experiments and discussions of the results are presented in Section 4.3 and Section 4.4 respectively.

4.2.1 Extracting Semantic Concept From WordNet

WordNet contains semantic relationships in synset, a set of synonyms representing a distinct concept. WordNet[1] is an online lexical database whose design was inspired by current psycholinguistic theories of human lexical memory. WordNet

[1]available online (http://wordnet.princeton.edu/)

can be described as an attempt to map the human understanding of words and the relationships between them. In WordNet, all words (and phrases) are tagged with their part of speech (POS); noun, adjective, verb, and so on, thus allowing an efficient lookup mechanism for a word's formal parts of speech. The most important relationship between words in WordNet is hypernym-hyponym, a semantic generalisation-specialisation relation that holds between two words that can (in a given context) express a related meaning; this is referred to as a *synset* [2]. Words having more than a single meaning appear in more than one synset, each representing a different concept. Table 4.1 presents relationships defined between synsets.

Table 4.1: WordNet Relationship of Synset [2]

Synset Relationship	Example
Hyponymy (words that describes things more specifically)	Apple is a hyponymy of fruit. Daffodil is a hyponymy of flower.
Hyperonymy (words that refer to broad categories or general concepts)	Car is a hyperonymy of "Toyota Camry", "Honda Civic" and "Ford Fiesta".
Holonymy (the whole of)	HAS PART COMPONENT: tree is a holonym of branch HAS PART MEMBER: office is a holonym of clerk IS MADE FROM OBJECT: tyre is a holonym of rubber.
Meronymy (part of a whole)	PART OF: leg is a meronym of table A MEMBER OF: sheep is a meronym of flock
Antonymy(Opposite of)	fast is an antonym of slow

A vocabulary problem exists when a term is present in several synsets as Figure 4.3 shows.

Determining the correct concept for an ambiguous word from several synsets is difficult, as is deciding the concept of a document containing several ambiguous terms. In this work, we exploited WordNet's hypernym-hyponym relationship as well as the antonym to determine whether we could obtain fewer but more general concepts and thus further improve the classification ability. Generated hypernyms will produce generalisations of the word, thus, searching in the email's body for more generic topics. In contrast, the hyponyms generated may guide the search in a similar way, except this time a specialisation of the word

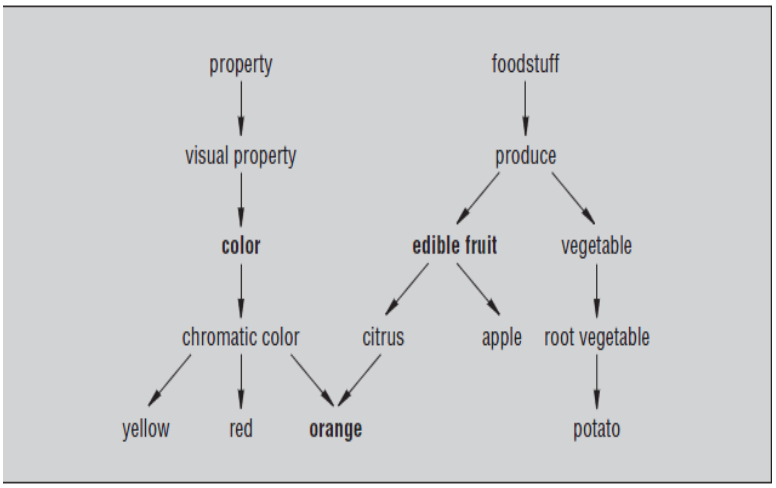


Figure 4.3: Two hypernym trees for the term "orange". The two-level hypernym for orange with the color concept is "color" but with the fruit concept is "edible fruit".

is performed. Moreover, by taking the relationship of antonym, it will generate a word which contradict the user's expectation of interest. Generating words that mean the opposite of these may be interesting because identification of these words in the email's body may be very useful as it may automatically fill in gaps in the user's knowledge.

The synset relationships allowed us to generate diverse terms (keywords) to improve searching for relevant documents. The relationships will generate words that are related to a noun or relevant word but contain somewhat different meanings and each may be useful to the searching mechanism. The WordNet relations in the extended version of AISEC involved searching up to two levels in the hierarchy of WordNet operations. Its representation is encode as list of transform word which create sets of user interest. This vector is referred as *Interest Word Vector* (IWV) with each position containing one of the five unique WordNet operations mentioned in Table 4.1.

The calculation of affinity of the artificial immune cell for WordNet in the extended AISEC version is modified using the combination of the words found in the cell's IWV and words generated by the cell's vector of interesting topics in user profile. To calculate the affinity, the mean of these two scores is taken. The score value will return a real number in the range of $\leq aff \leq 1$ and is compared with the classification threshold (Kc). The procedure may be outlined as follow in Algorithm 8. In Algorithm 8, ($count_{bc_{IWV},ag}$) is refer to the count of relevant words found in both ag while ($count_{INT,ag}$) is the count of features found in both ag and the set INT of word using wordNet operation. For cloning and mutation

procedure, the number of clones is proportional to the affinity of the cell and is calculated based on $num_{clones} \longleftarrow \lfloor aff \times Kl \rfloor$ and the number of position of IWV to be mutated in each clone cell is inversely proportional to the affinity of the cell and is calculated based on $num_{mutate} \leftarrow \lfloor (1 - aff) \times \mid bc_{IWV} \mid \times Km \rfloor$, where kl denotes a constant value which controls the rate of cloning and km denotes a constant value which controls the rate of mutation. Both parameter are in the range of [0 - 1]. Algorithm 9 shows the pseudo code for clone and mutation procedure. Note that Algorithm 9 is very similar to Algorithm 7.

PROCEDURE Affinity(ag,bc)
INT $\longleftarrow \oslash$;
foreach *(location i in bc_{IWV})* **do**
  $w \longleftarrow$ word in location i of bc_{IWV} ;
  $int\_word \longleftarrow$ generate set of words using wordNet operation in location i of bc_{IWV} ;
  INT $\longleftarrow$ INT $\cup$ int_word
end
aff $\longleftarrow \frac{1}{2} \times (\frac{count_{bc_{IWV},ag}}{|bc_{IWV}|} + \frac{count_{INT,ag}}{|INT|})$;
RETURN aff

Algorithm 8: The Affinity Function Procedure

PROCEDURE $Clone\_mutate$(bc, aff)
begin
  aff $\leftarrow$ Affinity(ag,bc)
  clones $\leftarrow \oslash$
  $num_{clones} \longleftarrow \lfloor aff \times Kl \rfloor$
  $num_{mutate} \leftarrow \lfloor (1 - aff) \times \mid bc_{IWV} \mid \times Km \rfloor$
  DO num_clones TIMES
    bcx $\leftarrow$ a copy of bc
    DO num_mutate TIMES
      p $\leftarrow$ a random point in bcxs feature vector
      w $\leftarrow$ a random word from the appropriate gene library
      replace word in bcx's feature vector at location p with w
    end
  end
  bcx's stimulation level $\leftarrow Ksb$
  clones $\leftarrow$ clones $\cup$ bcx
  Return clones
end

Algorithm 9: Procedure for Cloning and Mutating a Cell

4.3 Comparing the classification performance of the AISEC versions for a Single Class of Email

Experiment Objectives: To investigate the relative classification performance of the extended version of the AISEC algorithm against the original version of the AISEC algorithm for the classification of interesting email. The null hypothesis for this experiment is: "Two variations of one basic algorithm (Extended AISEC or Original AISEC) produce result that are not statistically significantly different (i.e, they don't have the same median)".

As mentioned in the experiment objectives, this experiment classified emails as interesting or uninteresting for the user and placed them in an appropriate folder. An interesting email refers to any email that can be regarded as PROJECT, for example, emails about meeting, presentation, reporting, resume and interview. When the user has read the email, feedback is taken from the user depending on his/her actions. There are two kinds of feedback that can be given to the system; positive, for the correct classification of the email, and negative, implying incorrect classification of the email. In the case of positive feedback, the user is not required to do anything except read the email that was previously classified as interesting. The AISEC system identifies such emails and assumes positive feedback. When the AISEC system has mis-classified an email, the user is required to move the email to its correct folder. The system recognizes the moved emails and interprets this as negative feedback. This process is repeated for every incoming email and the system adapts to the user's interests if there has been a change. Providing feedback for every email in a large test set during testing would be a time-consuming and tedious process. Therefore all the stages in classifying an email are automated. Simulated user feedback was given to both algorithms after the classification of each e-mail as the real class of each e-mail is known. Identifying a single email which has been read by the user in a folder requires iterating through the entire set of emails, which is both time and processor consuming. To avoid this, new folders are added and at each stage in the process of classifying the email and the email can be moved between the folders. At any time, a particular folder contains only the emails that need to be classified or to have feedback taken from them.

To compare the performances of the AISEC versions, we ran Secker's [9] original algorithm and our version (which used email bodies and the WordNet cor-

pus) on a large corpus of real-world email messages from the Enron Corporation[2]. The Enron email corpus has already been used as a case study for text classification, for example in [122–124]. The email data consist of over 600,000 emails from the email accounts of 158 employees, and the corpus was released in 2004. Although the dataset is large, many users' folders are sparsely distributed. For the comparison experiment, 3 sets of datasets were used where each of it consists of a subset of 100 folders, containing 2420 messages selected at random from the 600,000 emails. The system was trained with a training set of 200 emails. For the original AISEC version, the temporal ordering of emails within the test was preserved and only the words contained in the subject and sender fields of the e-mail were used. The sender information also included the return address, as these fields may differ. For the extended version, the same procedure was applied but using in addition the words from the body of the emails. The fields were tokenized using spaces and the characters $., , , (,), !, @, ,$ as delimiters and each token was inserted into a separate element of the correct feature vector. This pre-processing procedure was done for both original and the extended AISEC.

Since both algorithms are non-deterministic, their results depend on the random seed used for initialisation. The experiment was run multiple times with different random seeds and the median of the results is taken. During the reported runs of the AISEC algorithm, the same values for all parameters were used. These values (shown in Table 4.2) were arrived at by empirical testing during development using the same datasets used to evaluate AISEC's performance, and as a result tend to work well over this dataset. Therefore, the reported results evaluating AISEC's performance are over-optimistic. A legal range for each parameter is also indicated. We carried a statistical analysis based on the non-parametric Mann-Whitney-Wilcoxon or rank-sum test [125] to test whether the two algorithms' performances had different distributions (each having a different median), and the Vargha-Delaney A statistics [126] to measure the effect of size between these algorithms. A detailed explanation of these statistical tests is given in Section 4.3.1. To evaluate the classification performance, we calculated the true positive (TP), false positive (FP), true negative (TN) and false negative (FN) classification rates. We then used traditional *predictive classification accuracy* and Mathew's Correlation Coefficient (MCC) [127] measure. Section 4.3.2 contains a fuller explanation of this classification measurement.

[2]http://www.cs.cmu.edu/~enron/

Parameter	Optimized Value	Valid Range
Kc (Classification Threshold)	0.3	0 - 1
Ka (Affinity Threshold)	0.5	0 - 1
Kl (Clone Constant)	3.0	≥ 1
Km (Mutation Constant)	0.3	0 - 1
Ksb (Naive B Cells Stimulation Level)	175	0 - size of test set
Ksm (Memory B Cells Stimulation Level)	80	0 - size of test set
Kt (Initial Number of Memory cells)	20	0 - size of test set

Table 4.2: Parameters Used for Testing AISEC After Parameter Optimisation

4.3.1 The Non-parametric Statistics

Depending on whether an assumption is made on the underlying data distribution, statistical testing can be parametric or non-parametric. With parametric techniques, it is assumed that the observed variables are generated by some named probability distribution such as a Gaussian or a Normal distribution. The opposite of parametric statistics is non-parametric statistics in which the data are not assumed to belong to any particular distribution. If the distribution of data is known, parametric techniques can be more accurate and are generally advised. However, for many real world applications, it is often impossible to infer the underlying distribution of data and thus non-parametric techniques are more applicable. Moreover, non-parametric techniques are more powerful when a sufficiently large data set is available [128]. In our experiment, we employed non-parametric statistical analyses which included the Mann-Whitney-Wilcoxon or rank-sum test [125] to measure whether the two algorithms' performances had statistically significantly different distributions (with significantly different medians) and Vargha-Delaney A statistics [126] to measure the effect size between these algorithms' performances (whether or not they are scientifically significantly different).

Statistical Significance

We analysed statistical significance in order to determine whether any observed differences between the algorithms were likely to have occurred by chance. We applied the Mann-Whitney-Wilcoxon test [125]. This is a non-parametric test: it makes no particular assumptions as to the distribution of the response. An

Value of A statistics	Implication on Effect Size
A = 0.5	No Effect (no difference in algorithms' performances)
A = 0.56	Small Effect (low difference in algorithms' performances)
A = 0.64	Medium Effect (medium difference in algorithms' performances)
A = 0.71	Big Effect (big difference in algorithms' performances)

Table 4.3: The Vargha-Delaney A statistics Value and its Implication on Effect Size [126]

equivalent parametric test, such as the t-test, makes specific assumptions about the data such as a normal distribution, and without careful analysis of whether the assumptions have been met, the results of parametric tests can be unreliable [128]. The null hypothesis for the rank-sum test is that the predictive performance measures of the two algorithms have identical distributions with equal medians; the alternative hypothesis is that the distributions are different. We used two sided test with a 5% significance level whereby if the test returns a p-value of $<$ 5% , the null hypothesis is rejected, indicating that any observed difference in the number of evaluations is unlikely to have occurred by chance.

Scientific Significance

It is possible for an observed difference in algorithm performance to be statistically significant even though the actual magnitude of the difference is small. The effect size can be very small compared to the inherent variability in the results owing to the stochastic nature of the algorithms [126]. To guard against this situation (which can occur when the number of experimental trials is excessive), we also tested for scientific significance, i.e. that the effect size was sufficiently large to be scientifically important. We used the Vargha-Delaney A statistic [126], to assess the effect size. The A statistic value from the Vargha-Delaney A statistics lies between the value of 0 and 1. Table 4.3 presents a list of the A values and the implication on the tested sample size. We used this guideline to assess the effect size.

4.3.2 Matthew's Correlation Coefficient (MCC) and Confusion Matrix

The evaluation of IF systems has benefited from long experience in the evaluation of information retrieval (IR) systems. IR systems have been traditionally evaluated on the basis of the *predictive accuracy*, *precision* and *recall*. To define this terminology, given a class of documents, all documents in that class are referred to as 'positive documents', and all others as negative no matter how many different classes there are. The accuracy of a predictive system over a test set is given in Equation 4.1:

$$accuracy = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{4.1}$$

Furthermore, there is usually a much larger amount of negative data than the amount of positive data, and therefore a classifier may achieve high accuracy simply by predicting all records as belonging to that negative class. It is therefore desirable to introduce metrics that examine the errors made by the classifier. Precision and recall are such measures:

1. **Precision** is the proportion of the results returned that are actually relevant to the search query (given in Equation 4.2).

2. **Recall** is the proportion of relevant results returned with respect to the total number of relevant results (given in Equation 4.3).

$$precision = \frac{\text{Number of correct positive predictions}}{\text{Number of positive predictions}} \tag{4.2}$$

$$recall = \frac{\text{Number of correct positive predictions}}{\text{Number of positive documents in the set}} \tag{4.3}$$

The number of positive and negative documents in the dataset in typical classification scenarios are often unbalanced, so too are the costs associated with classifying or misclassifying positive or negative examples [129]. For this reason, a confusion matrix, shown in Figure 4.4, may be desirable for illustrating classifier performance. The confusion matrix consists of four cells:

- **TP (True positive)** count: The number of documents classified as positive that were positive.

- **FP (False Positive)** count: The number of documents classified as positive but were negative.

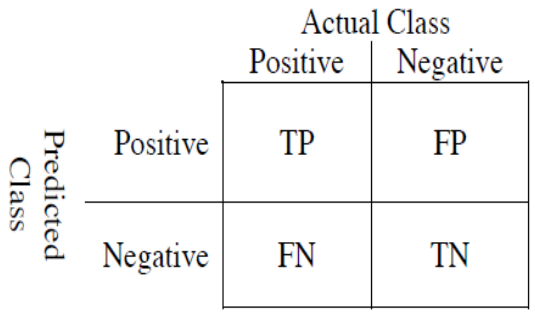


Figure 4.4: An Example of Confusion Matrix

- **TN (True negative)** count: The number of documents classified as negative that were negative
- **FN (False negative)** count: The number of documents classified as negative but were positive.

Given the TP count and the FP count of two classifiers, it is often difficult to judge which is superior if one classifier has a higher TP count whilst the other has a lower FP count. For these cases, the Matthew's correlation coefficient(MCC) [127] can be a good indicator. It has been pointed out in [129] that there is no single best metric for comparing the performance of classifiers but the MCC is often regarded as the most appropriate. In our experiment, to further evaluate the performance of the algorithms in terms of classification, we adopted the MCC measure. The MCC was chosen because it is unaffected by sampling biases, which may occur when the dimensions of the learning sets are very different [129]. The calculation of a MCC score is given in Equation 4.4. The MCC ranges from $-1 \leq C \leq 1$. A value of $C = 1$ indicates the best possible prediction, in that every interesting email was correctly predicted, and only true interesting emails were predicted. A value of $C = -1$ indicates the worst possible prediction (or anti-correlation), where not a single interesting email was correctly predicted and all the uninteresting emails were incorrectly predicted as interesting. Finally, a value of $C = 0$ would be expected for a random prediction scheme.

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{4.4}$$

Table 4.4: Statistical and Classification Performance Comparison of Original and Extended AISEC [131]

Algorithm	**Statistical Analysis**		**Classification Performance**	
	Two-tailed P-Value	**Vargha-Delaney A Statistics**	**Predictive Accuracy**	**MCC**
Original AISEC	0.0100	0.62099	86.67%	0.54
Extended AISEC			90.10%	0.60

4.3.3 Experimental Result

In order to compare the versions of AISEC, first, we tested the (alternative) hypothesis that the median accuracy of the two algorithms was not statistically (rank sum) significantly different. Next, we looked at the effect size between the two samples (using the Vargha-Delaney A-test). We used Matlab [130] as an exemplar of mathematical software to carry out the statistical testing as well as the graph presentation. Matlab is a mathematical computing platform extensively used by mathematicians, scientists and engineers to analyse and visualise data, implement algorithms, and build models. The core capability of Matlab is the ability to store data as matrices and efficiently manipulate the data using linear algebra. An extensive range of 'toolkits' provide additional functionality in specific application areas, such statistical analysis, optimisation, image processing, signal processing, symbolic mathematics and wavelet analysis.

The results of the statistical test are shown in Table 4.4. The critical p-value indicates that, with 95% confidence, there is a significance difference between the performances of the two algorithms. In summary, the two samples (the performances of the original version and the extended version) have different distributions which indicates that their medians are different. However, the A value from the Vargha-Delaney A statistics shows that the difference has only a small effect, meaning that the performance of the two algorithms showed a small effect when tested on the email corpus of Enron Corporation. The statistical analysis showed a small significant difference between the algorithms' performances, however, there was a difference in terms of the classification performance. Table 4.4, shows that the extended AISEC performed at least as well as Secker's original algorithm on the tested dataset.

The line chart in Figure 4.5 shows the changing predictive accuracy and MCC value after the classification of each mail by the number of e-mails classified. This uses the accuracy and MCC measures described above and therefore details the results for the test set. This chart was drawn using the median of the 740 runs as used to construct Table 4.4. Of interest are the areas 900 to 1,100 and 1,600 to 2,100 e-mails classified. In both situations, the original AISEC exhibited an increase in accuracy and MCC value. Although there was a slight decrease in accuracy and MCC value for the extended version, it was not obvious and the algorithm still performed a better classification.

The experiments have shown that using words from the email body and the WordNet for the AISEC has a positive result in identifying new and potentially uninteresting e-mails. The next stage was to carry out further analysis of the extended version of AISEC, to determine its ability to classify emails into multiple categories, whilst adapting to changing user interest. In Section 4.4 this second experiment is described and an analysis is undertaken of the multiple topic classification.

4.4 Experimenting with the Extended AISEC on the Classification of Multiple Email Topics

Experiment Objectives: To investigate the implementation of the algorithm to represent multiple topics in parallel and to demonstrate the ability of the algorithm to forget a previous topic when it starts to process a new topic. The null hypothesis for this experiment is: "The extended AISEC is not capable of classifying multiple email topics".

The ability of the extended AISEC to classify interesting emails based on a real data set from an email corpus has already been demonstrated. In a real world situation, however, users are typically interested in more than one topic in parallel and both their interests and the information environment change over time. Therefore, users may be interested in one or more email(s) topic according to their interests. To more accurately simulate a real situation, we initially performed experiments in which the system had to represent more than one topic in parallel and adapt changes on interest in them. We were particularly interested in these multiple topics experiments because they more accurately reflect a real situation. Moreover, the experiment not only had to investigate the performance

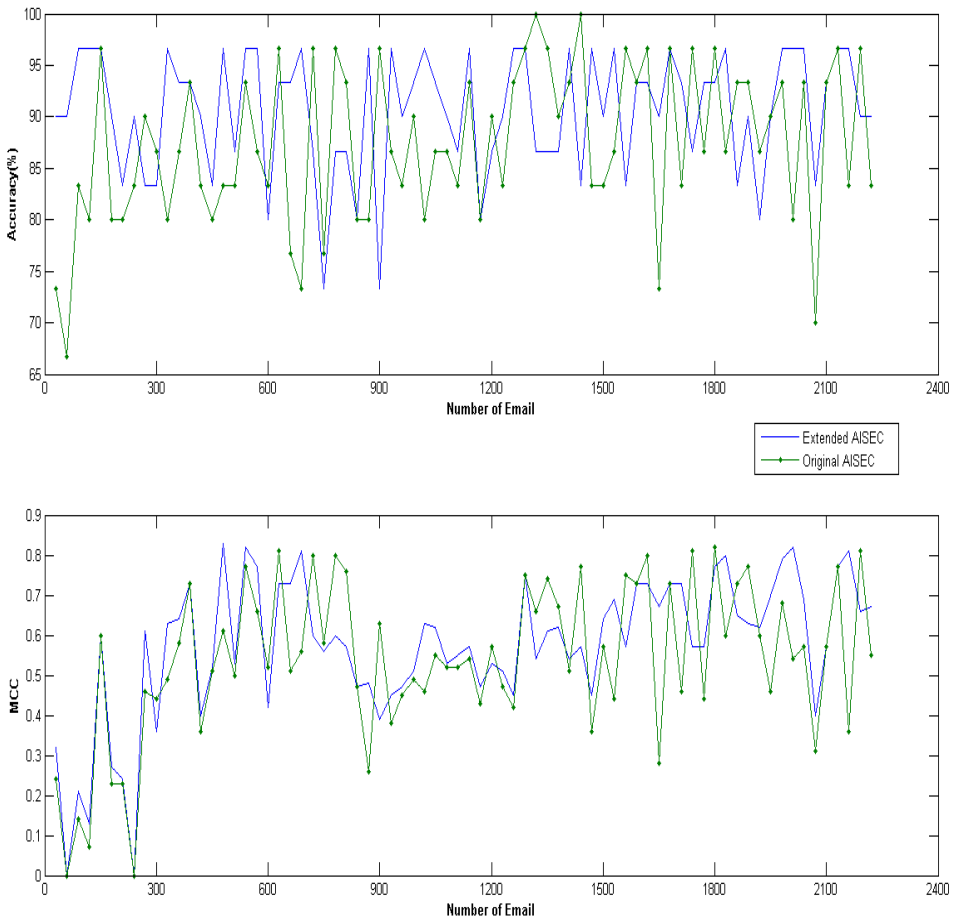


Figure 4.5: Changes in Predictive Accuracy and MCC Value by Email Classified

of the algorithm to represent multiple topics in parallel, but also to demonstrate the ability of the algorithm to forget a previous topic when it starts to process a new topic.

The experiments are based on the real-world email messages corpus from the Enron Corporation[3]. There are more than 200 folders for the email accounts of 158 employees and each folder represent more than 10 topics. Our experiments were based on 49 email topics with a maximum size of 379 emails (the email topic of *meeting*) and a minimum size of 3 emails (the email topic of *baseball trivia*). Figure 4.6 presents the list of the email topics and its corresponding size. In the case of multiple topics, both B cells receptor (the naive B cells (BC) and memory B cells (MC)) and the email message (the email's body including its title) are represented

[3]http://www.cs.cmu.edu/~enron/

as weighted keyword vectors. Thus, the vector space comprises of keywords extracted from the content (including title) of the email messages in the collection. For each of the 49 topics, the population of B cells is initialised by turning the weighted keyword vectors of the first 100 relevant emails into naive B cells. 25 random elements from BC are inserted into MC. The initial stimulation count of naive B cells and memory B cells is set to Ksb = 175 and Ksm = 80 respectively. The initialisation process proceeds with the cloning of memory cells that have a strong affinity to initialisation of relevant email. Mutated vectors of the original memory cells are thus introduced to BC. The affinity between two cells was measured as the proportion of common keywords in their vectors. The measure will return a value between 0 and 1. If the affinity between a B cells (bc) and an email message (antigen, ag), or another B cells is greater than Ka = 0.5, then the B cells is activated and cloned. The number of clones depends on the affinity between bc and ag and a cloning constant, Kl = 3.0. Each clone is mutated by randomly choosing a number of keywords in bc and replacing them with the corresponding keywords in ag. The number of mutated keywords is proportional to the length of the clone's keyword vector and a mutation constant (Km = 0.7) and is inversely proportional to the affinity between bc and ag. The initial stimulation count of clones is set equal to 20. Noted that, the extended AISEC parameter used in this experiment is identified after the parameter optimisation.

In each evaluation cycle, we initialise the population of B cells and then it sequentially evaluates each email message in the collection. An email message is assigned a relevance score with the highest affinity achieved among the B cells. Whenever an email message is relevant to the current topic of interest then the learning process takes place. The evaluation process is described further in the next section.

4.4.1 The Experimental Methodology

In [132], a methodology was proposed for evaluating the ability of a user profile for continuous adaptation in a dynamically changing environment. We followed this methodology to evaluate the profile and the process is summarised below:

1. Pre-process all the emails.
2. Start with empty profile.
3. The empty profile assigns a zero score to the email until it encounters an email relevant to the first of the evaluated topics (for example: topics on 'meeting') and then it is initialised.

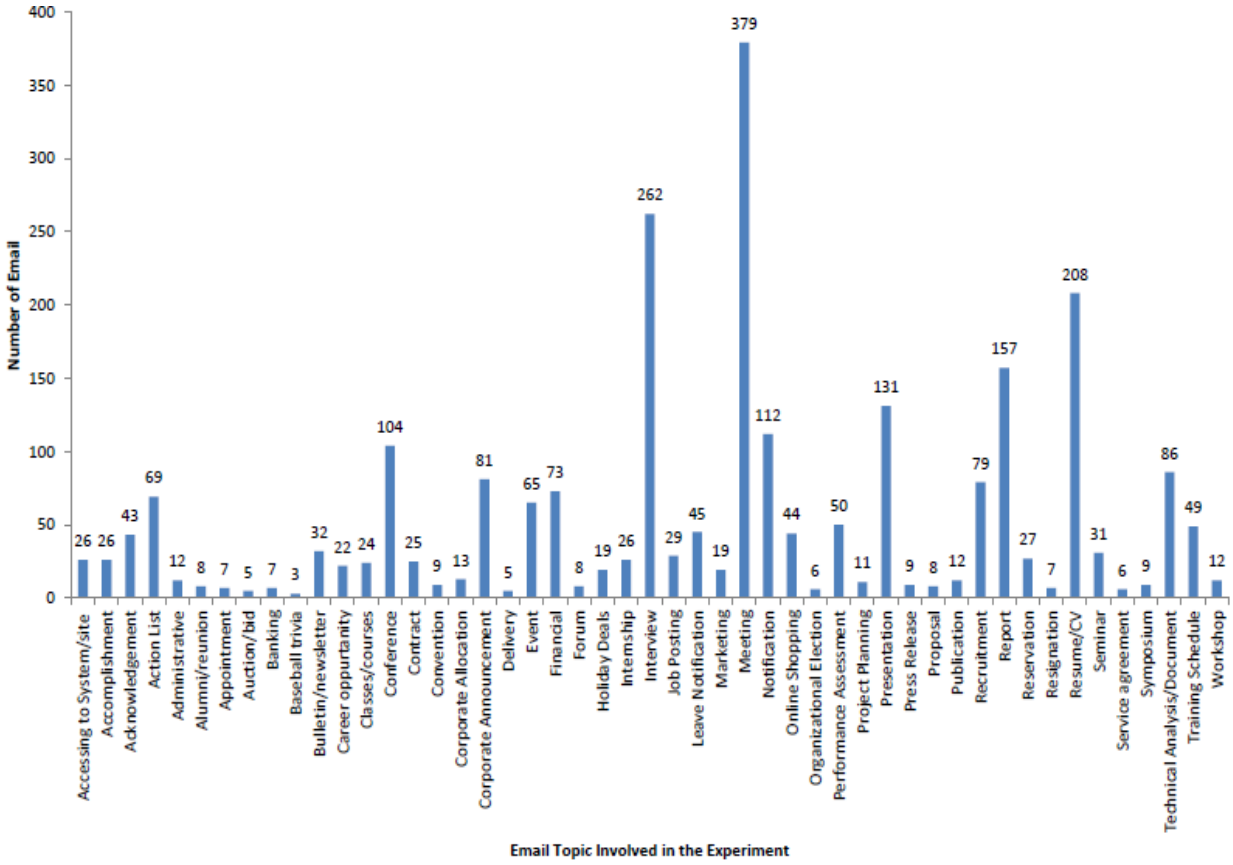


Figure 4.6: The Email Topics and their Corresponding Size

4. Initialize the 1st topic. Every time it encounters an email which belongs to topic "meeting", it evaluates the email and then uses it as positive feedback and adapts based on it.

5. Process all email sequentially;

   (a) The profile assigns a relevance score to each processed email.

   (b) Calculate the classification performance (i.e. MCC score) for the 1st topic.

   (c) New topic enter.

   (d) The profile is re-initiated but now it has to adapt with the new topic.

   (e) The classification performance for the 2nd topic is calculated when all emails in collection have been processed.

6. The process is continued with the rest of the topics and terminates when the last topic has been used as positive feedback.

In the multiple email topics experiment (for instance, two email topics experiment), the profile initially learns the first two topics in parallel. The system then forgets the first topic and continue to learn the second topic. Similarly, in the three-topic scenario, the profile has to be able to represent three topics in parallel, the first triple first, then the second triple and so on. The same condition applies for the four-topic scenario, where the profile has to be able represent four topics in parallel, learn it and be able to forget the previous set. We believe that this methodology can be used for testing the ability of adaptive systems, such as AIS, for online learning (and forgetting) in a complex, multi-dimensional and dynamic environment.

To evaluate the performance of the algorithm, we carried out a statistical analysis based on the non-parametric Mann-Whitney-Wilcoxon or rank-sum test [125] to test whether two algorithms' performance had different distributions (each having a different median), and the Vargha-Delaney A statistics [126] to measure the effect size between these algorithms. We then used Mathew's Correlation Coefficient (MCC) [127] measure to evaluate the performance of the algorithms in terms of classification. The end result for single email topic is 49 MCC score, one for each topic of interest. In the two-topic, three-topic and four-topic cases, the process mentioned above is applied for all consecutive pairs, triple pairs and quadruple pairs, respectively. At the end of each evaluation period, the MCC's score for each individual constituent topic is calculated, and also the combined MCC score, which is calculated based on the aggregate set of relevant email messages for all constituent topics. The experiment was conducted as a comparison between the extended AISEC version with two chosen baseline approaches, namely the Naive Bayes and the Recurrent Neural Network (RNN). The baseline approach is described in detail below.

4.4.2 The Baselines

To determine the relative classification performance of AISEC, it was necessary to test it against another continuous learning algorithm. Therefore we needed a baseline approach in order to evaluate the performance of our algorithm. In this study, we adopted two types of baseline approach: a statistical classifier (the Naive Bayes) and a connectionist network classifier (the recurrent neural network). A description of these baseline approaches is given next.

Naive Bayes The Naive bayes is a classification algorithm based on a Bayes's rule, which assumes that the attributes $X_1, ..., X_n$ are all conditionally indepen-

dent of one another, given the class, Y. The value of this assumption is that it dramatically simplifies the representation of $P(X|Y)$, and the problem of estimating it from the training data. More generally, when X contains n attributes which are conditionally independent of one another given Y, we have:

$$P(X_1, ...X_n \mid Y) = \Pi_{i=1}^{n} P(X_i \mid Y) \tag{4.5}$$

In this work, a variation of the Naive Bayes algorithm was implemented according to Equation 4.6 taken from [133].

$$v_{NB} = argmax_{v_j \varepsilon V} P(v_j) \Pi_i P(a_i \mid v_j) \tag{4.6}$$

where the set of class v = (uninteresting email, interesting email), $P(v_j)$ is the probability of email belonging to class v_j and calculated based on the frequency of occurrence of class v_j in the initialisation set. The term $P(a_i \mid v_j)$ is the probability of the email containing word a_i given the email belongs to class v_j. This probability is calculated using observed word frequencies over the initialisation data. In this modified algorithm, these observed word frequencies are updated based on the same user feedback mechanism as in AISEC.

When the size of the training set is small, the relative frequency estimates of probabilities, $P(a_i \mid v_j)$, will not be reasonable: if a word never appears in the given training data, its relative frequency estimate will be zero. This situation may happened due to people tend to use very different words. Therefore, to avoid the case of zero probabilities value, we applied the concept of Laplacian correction (Laplace Estimator) [134] to estimate $P(a_i \mid v_j)$. The estimate of the probability $P(a_i v_j)$ is given as:

$$P(a_i \mid v_j) = \frac{n_{ij} + 1}{n_j + k_j} \tag{4.7}$$

where n_j is the total number of words in class v_j, n_{ij} is the number of occurrences of word a_i in class v_j and k_j is the vocabulary size of class v_j. This is the result of the Bayes estimation with a uniform prior assumption, i.e. probabilities of the occurrences of words appearing in class, v_j are equally likely.

The widely-known Naive Bayes classifier was chosen as a suitable comparison algorithm. In [133] the author states:

> "....probabilistic approaches such as Naive Bayes are among most effective known to classify text documents...." (p. 180) [133]

In [9] it is stated that due to the ability to account for the unbalanced penalties and its computational tractability and competitive performance with easy imple-

mentation [135,136], the Naive Bayes classifier remains popular for the e-mail domain for example, spam filtering. Moreover, the learning process of Naive Bayes is extremely fast compared with current discriminative learners, which makes it practical for large real-world applications [136].

Recurrent Neural Network (RNN) The standard feedforward neural network, or multilayer perceptron (MLP), is a member of the *family* of neural networks. Feedforward neural networks have been applied in tasks of prediction and classification of data for many years. More recently, a new class of neural networks, based upon feedforward neural networks, has been introduced. These dynamic neural networks (or *neural networks for temporal processing*) extend the feedforward networks with the capability of dynamic operation, which means that the behaviour of the neural network depends not only on the current input (as in feedforward networks) but also on previous operations of the network [137]. Neural networks for temporal processing can be grouped in two classes. The first class, called *time-delay networks*, is based on feedforward neural networks. These structures perform some temporal pre-processing of the input data before the data is presented to neurons in the network. The second class consists of *recurrent neural networks (RNN)*, which have recurrent connections (neuron outputs are feed back into the network as additional inputs) as well as the structures of delay elements seen in time-delay networks. Classes of RNN architectures include Fully Recurrent Neural Networks (FRNN), Partially Recurrent Networks (PRN) and Simple Recurrent Networks (SRN).

In this study, we examined the use of SRN for email classification and compared its performance with our extended version of AISEC. The words in the email messages are represented based on semantic vector representation. These vector are determined based on the frequency of a word in different categories of email topics. Each word w is represented with vector (v(w,c_1), v(w,c_2),....,v(w,c_n)) where c_i represents a certain categories of email topics. A value $v(w, c_i)$ is computed for each dimension of the vector as the *normalized* frequency of occurrences of word w in category email topic c_i (the normalized category frequency), divided by the *normalized* frequency of occurrences of word, w in the email corpus (the normalized corpus frequency) that is:

$$v(w, c_i) = \frac{\text{Norm. freq. of } w \text{ in } c_i}{\Sigma_j \text{ Norm. freq. for } w \text{ in } c_j}, \text{for} j \varepsilon 1, ..., n \tag{4.8}$$

and where

$$\text{Norm. freq. of } w \text{ in } c_i = \frac{\text{Freq. of } w \text{ in } c_i}{\text{Number of emails in } c_i} \tag{4.9}$$

An example of word and their semantic vector representation is given in Table 4.5, however, not all the categories of an email topics are shown. For the case of a single topic, it consists of 49 categories, for the two-topic scenario it consists of 24 categories, for the three-topic scenario it consists of 18 categories and for the four-topic scenario consists of 12 categories of an email topics. As can be seen in the example, domain-dependent words such as 'can' have general distributions while domain-independent words such as 'meet' and 'order' have more specific preferences.

Table 4.5: Example of semantic vectors representation for SRN. For the purpose of this illustration, not all the categories of an email topics are shown.

Word	Email Topic						
	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic n
CAN	0.03	0.06	0.09	0.01	0.04	0.07	....
MEET	0.07	0.03	0.04	0.03	0.02	0.03	....
CHECK	0.05	0.05	0.05	0.04	0.02	0.01	....
ORDER	0.06	0.04	0.07	0.02	0.04	0.02	....
OUT	0.02	0.04	0.03	0.06	0.06	0.06	....
OF	0.01	0.04	0.03	0.06	0.08	0.05	....
SEE	0.03	0.05	0.04	0.08	0.03	0.04	....
IF	0.03	0.01	0.02	0.06	0.01	0.03	....

In terms of network architecture, the SRN network architecture in this experiment exploited a single hidden layer with one time step of recurrent connections. The vector representation for one word was shifted into this input layer at each time step. For instance, the first vector representing the first word initialized the input units with an initial activation. Basically, the activation of the input units was used to compute the activation of the hidden layer by summing the incoming weighted activation [138]. Then, the activations of the hidden layer were copied to the context layer. In our experiment, we tested context layers (hidden layers) with 320 units. The illustration of the network for the task of email topic classification is depicted in Figure 4.7.

In one epoch, or cycle of training through the training samples, the network is presented with the representation of a semantic vector from the training set and

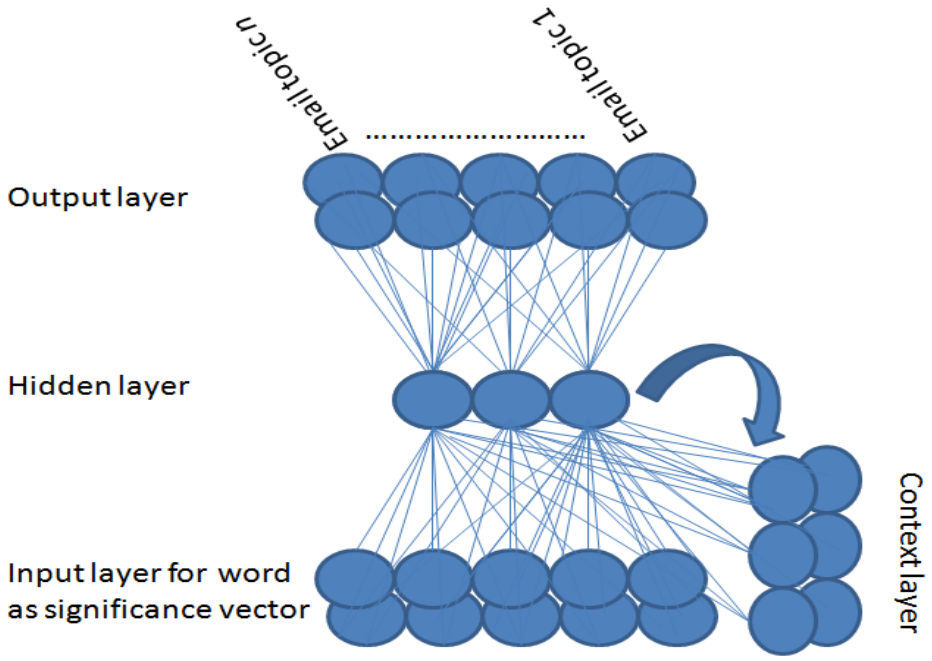


Figure 4.7: Recurrent Simple Network (SRN) for email topic Classification. The Large Arrow indicates the 1:1 Copy Connections from the Hidden Layer to the Context Layer

the weights are adjusted. Training was performed after each word of a phrase according to the supervised learning rule for SRN [138]. During training, the hidden layer develops a reduced representation of the incremental context in a phrase. Therefore, the values of the hidden layer at time $t-1$ can be used for the initialisation of the context layer for the subsequent word at time t. Each context layer unit is connected with each hidden layer unit via a weighted connection. The values of the output layer are computed in a similar manner to the hidden layer by thresholding the weighted activation coming from the hidden layer. The output layer represents the output of the desired email topic and there is one unit for each of the 49 topics (note: the two-topic scenario consisted of 24 topics, the three-topic scenario consisted of 18 topics and the four-topic scenario consisted of 12 topics) in the email corpus. The network was trained for 1000 epochs on the training samples using a fixed momentum term and a changing learning rate. The initial learning rate was 0.01, but this changed at 500 epochs to 0.006 and then again at 800 epochs to 0.001. The results for the SRN experiment are shown in Table 4.6 to Table 4.9 and are discussed in Section 4.4.3.

4.4.3 The Experiment Result and Analysis

We used the methodology described in Section 4.4.1 to compare our extended version of AISEC with the chosen baseline approach discussed in Section 4.4.2. The Naive Bayes is a popular choice in the task of classification [25,135,136,139] and a popular baseline for comparative experiments [25,140]. The RNN approach (specifically the SRN) was chosen because from the perspective of connectionist networks, it has been demonstrated that a connectionist network can be used under real-world constraints for text classification [51,137].

In this section, the results of the experiment are discussed. Table 4.6 to Table 4.9 present the complete comparative experiment results for the single-topic, two-topic, three-topic and four-topic experiments. Note that, the single-topic is referred to binary classification scenario. In particular, Table 4.6 presents the MCC scores achieved for the single-topic for the extended version of AISEC, Naive Bayes and the SRN (column two to column four). The final three columns ("diff(A)% ", "diff(B)% " and "diff(C)% ") present, respectively, the differences in percentage between the extended AISEC (algorithm 2) with Naive Bayes (algorithm 1), the SRN (algorithm 2) with Naive Bayes (algorithm 1) and the extended AISEC (algorithm 2) with SRN (algorithm 1). The negative sign indicate that the algorithm 1 produced better MCC score compare to algorithm 2. While non negative sign indicate that the algorithm 2 produced better MCC score compare to algorithm 1. The statistical analysis for the comparative experiment between the extended AISEC version with the baseline approach is presented in the last row of the list of email topics. The analysis includes the median, standard deviation and the non-parametric statistical analysis based on the Rank-Sum and the Vargha-Delaney A statistics. Similarly, Table 4.7 to Table 4.9 (column two to column four) shows the MCC score for the two-topic, three-topic and four-topic experiments respectively.

As shown in Table 4.6 for single-topic experiments, in overall, the extended AISEC and the baseline approaches showed a good performance where the median values for MCC scores were above 0.5. In this experiment, the extended AISEC shows the value of MCC which is higher compared to the result in Table 4.4 in Section 4.3.3. A particular reason is, a larger training set is used in this experiment. The classification measurement for single topic indicates that the algorithms gave a good prediction in single-topic classification. However, as already discussed, representing a single topic of interest is a relatively simple problem and does not accurately reflect a real situation. In reality, a user is typically interested in more than one topic in parallel. In contrast, in terms

Table 4.6: Results for single-topic experiments: Email Topic(first col.), MCC Score (two to fourth col.), differences in percentage between the extended AISEC version with Naive Bayes (fifth col.), the SRN with Naive Bayes (sixth col.) and the extended AISEC version with SRN (seventh col.).

Topics	MCC Score			Diff.(A)	Diff.(B)	Diff.(C)
	Extended AISEC	SRN	Naive Bayes			
Acc.Syst	0.784	0.711	0.558	22.542	15.318	7.224
Accomp.	0.700	0.638	0.357	34.367	28.181	6.186
Acknow.	0.747	0.697	0.514	23.247	18.243	5.004
Act.List	0.729	0.703	0.515	21.481	18.835	2.647
Admin	0.695	0.569	0.510	18.535	5.890	12.645
Alumni	0.745	0.695	0.520	22.444	17.426	5.018
Appoint.	0.797	0.734	0.504	29.349	22.977	6.372
Auction	0.737	0.771	0.537	19.955	23.398	-3.443
Banking	0.763	0.701	0.555	20.884	14.618	6.266
Baseb. triv.	0.850	0.745	0.557	29.317	18.869	10.448
Newslett.	0.800	0.616	0.561	23.940	5.547	18.393
Career opp.	0.802	0.709	0.564	23.786	14.523	9.263
Course	0.799	0.712	0.533	26.682	17.910	8.772
Conf.	0.731	0.734	0.585	14.601	14.945	-1.344
Contract	0.714	0.688	0.570	14.368	11.746	2.622
Convent.	0.810	0.746	0.568	24.199	17.832	6.366
Corp. Alloc.	0.741	0.707	0.527	21.419	17.971	3.448
Corp. Announce.	0.755	0.736	0.590	16.511	14.644	1.867
Delivery	0.817	0.756	0.591	22.554	16.450	6.104
Event	0.705	0.644	0.593	11.206	5.079	6.127
Financ.	0.721	0.676	0.601	12.008	7.582	4.426
Forum	0.739	0.734	0.604	13.489	12.930	0.558
Hol.Deals	0.696	0.687	0.554	14.220	13.288	0.931
Intern.	0.630	0.681	0.613	1.711	6.798	-5.087
Interview	0.630	0.656	0.564	6.643	9.269	-2.627
Job Post.	0.705	0.617	0.538	16.657	7.853	8.804
Leave Not.	0.739	0.600	0.531	20.818	6.963	13.856
Market	0.764	0.712	0.529	23.433	18.276	5.157
Meeting	0.735	0.710	0.617	11.844	9.332	2.512
Not.	0.733	0.729	0.630	10.354	9.918	1.437
Onl.Shop	0.750	0.734	0.632	11.875	10.266	1.610
Org.Elec.	0.759	0.631	0.600	15.937	3.136	12.802
Perf. Asses.	0.806	0.738	0.623	18.283	11.511	6.773
Pro.Plan.	0.753	0.718	0.641	11.275	7.780	3.495
Present	0.716	0.723	0.639	7.628	8.336	-1.708
Press	0.717	0.718	0.652	6.530	6.611	-0.180
Proposal	0.718	0.653	0.669	4.894	-1.571	6.465
Publica.	0.768	0.728	0.675	9.348	5.354	3.994
Recruit.	0.725	0.699	0.639	8.681	6.016	2.665
Report	0.733	0.700	0.679	5.373	2.139	3.234
Reserve.	0.684	0.626	0.647	3.740	-2.107	5.847
Resign.	0.753	0.647	0.612	14.116	3.535	10.581
Resume	0.738	0.717	0.656	8.146	6.116	2.030
Seminar	0.677	0.628	0.590	8.635	3.754	4.881
Ser.Agree.	0.708	0.687	0.604	10.420	8.324	2.095
Symp.	0.759	0.631	0.612	14.671	1.936	12.735
Tech.Doc.	0.733	0.728	0.620	11.209	10.734	0.475
Train.	0.690	0.676	0.596	9.481	8.021	1.461
Workshop	0.718	0.688	0.563	15.567	12.591	2.976
median	0.737	0.701	0.590			
std. dev.	0.044	0.045	0.057			
			rank sum	3604.000	3478.000	3090.000
			Z-val	8.370	7.475	4.718
			p-value	5.741E-17	7.703E-14	2.376E-06
			A value	0.991	0.938	0.777

Table 4.7: Results for two-topic experiments: Email Topic(first col.), MCC Score (two to fourth col.), differences in percentage between the extended AISEC version with Naive Bayes (fifth col.), the SRN with Naive Bayes (sixth col.) and the extended AISEC version with SRN (seventh col.).

Topics(1st:2nd)	Combined MCC Score			Diff.(A)	Diff.(B)	Diff.(C)
	Extended AISEC	SRN	Naive Bayes			
Acc.Syst.:Deliv.	0.764	0.749	0.678	8.556	7.138	1.418
Accomplish:Acknow	0.600	0.602	0.526	7.386	7.544	-0.158
Act.List:Not.	0.747	0.701	0.517	22.996	18.414	4.582
Admin.: Resig	0.702	0.608	0.517	18.574	9.162	9.412
Accomp.:Acknow	0.600	0.602	0.526	7.386	7.544	-0.158
Alumni:Recruit.	0.665	0.667	0.520	14.497	14.667	-0.170
Appoint.:Reserv.	0.715	0.672	0.561	15.371	11.056	4.315
Auction:Event	0.787	0.771	0.514	27.316	25.667	1.649
Bank.:Financial	0.737	0.708	0.513	22.339	19.505	2.834
Base.triv.:Newsletter	0.763	0.742	0.427	33.675	31.500	2.175
Car. oppurt.:Job Post.	0.750	0.657	0.467	28.317	19.048	9.269
Cours.:Train.	0.601	0.684	0.510	9.118	17.443	-8.325
Conf.:Symp.	0.801	0.788	0.588	21.255	19.966	1.289
Cont.:Serv.Agree.	0.789	0.709	0.578	21.155	13.070	8.085
Convent.:Sem.	0.711	0.720	0.532	17.919	18.825	-0.907
Corp. Alloc.:Press Rel.	0.714	0.680	0.556	15.825	12.449	3.376
Corp. Ann.:Newslet.	0.810	0.792	0.529	28.045	26.250	1.795
For.:Workshop	0.741	0.711	0.570	17.097	14.074	3.023
Hol.Deals:Online Shop.	0.755	0.736	0.519	23.594	21.734	1.860
Intern.:Recruit.	0.817	0.800	0.533	28.355	26.667	1.689
Interview:Res.	0.705	0.678	0.406	29.931	27.255	2.676
Leave Not.:Not.	0.701	0.695	0.445	27.616	-0.997	2.619
Market.:Corp.Announc	0.739	0.717	0.424	31.486	29.238	2.248
Meeting:Present.	0.696	0.665	0.468	22.836	19.725	3.110
Onl.Shop.:Delivery	0.630	0.585	0.449	18.082	13.604	4.478
Org.Elec.:Corp.Announc	0.630	0.642	0.467	16.314	17.499	-1.186
Pub.:Report	0.705	0.670	0.510	19.413	15.925	3.488
Perf. Asses:Report	0.739	0.715	0.513	22.546	-0.206	-2.340
Plan.:Proposal	0.764	0.739	0.420	34.416	31.917	2.499
Rep.:Tech.Doc.	0.735	0.746	0.431	30.475	31.564	-1.090
median	0.729	0.676	0.500			
std. dev.	0.052	0.057	0.057			
			rank sum	1275.000	1267.500	1074.500
			Z-val	6.470	6.400	1.439
			p-value	9.798E-11	1.553E-10	6.783E-04
			A value	0.994	0.990	0.760

Table 4.8: Results for three-topic experiments: Email Topic(first col.), MCC Score (two to fourth col.), differences in percentage between the extended AISEC version with Naive Bayes (fifth col.), the SRN with Naive Bayes (sixth col.) and the extended AISEC version with SRN (seventh col.).

Topics(1st:2nd:3rd)	Combined MCC Score			Diff.(A)	Diff.(B)	Diff.(C)
	Extended AISEC	SRN	Naive Bayes			
Acc.Syst:Deliv.:Act.List	0.724	0.649	0.578	14.556	7.138	7.418
Accomp.:Acknow.:Not.	0.601	0.602	0.526	7.486	7.544	-0.058
Admin.:Resign.:Leave Not.	0.647	0.670	0.517	12.996	15.313	-2.317
Alumni:Recruit.:Job Post.	0.729	0.708	0.517	21.274	19.162	2.112
Appoint.:Reserv.:Meeting	0.695	0.667	0.520	17.497	14.667	2.830
Auct.:Event:Newsletter	0.745	0.613	0.561	18.371	5.146	13.225
Bank.:Financial:corp. Alloc.	0.747	0.631	0.514	23.316	11.667	11.649
Base.triv.:Newslett.:Event	0.737	0.690	0.513	22.339	17.705	4.634
Car.oppurt.:Job Post.:Recruit.	0.763	0.642	0.527	23.675	11.500	12.175
Course:Train.:Workshop	0.650	0.657	0.513	13.717	14.448	-0.731
Conf.:Symp.:Seminar	0.610	0.584	0.510	10.018	7.443	2.575
Contract:Serv.Agreemt.:Corp.Alloc	0.762	0.688	0.518	24.355	16.966	7.389
Convention:Sem.:Forum	0.799	0.679	0.518	28.155	16.070	12.085
Corp.Alloc.:Press Rel.:Corp. Announce	0.721	0.610	0.512	20.919	9.825	11.093
Corp.Ann.:Newslett.:Press Rel.	0.714	0.680	0.506	20.825	17.449	3.376
Forum:Workshop:Training	0.761	0.692	0.509	25.145	18.250	6.895
Hol.Deals:Online Shop.:Delivery	0.741	0.711	0.470	27.097	24.074	3.023
Intern.:Recruit.:Career Oppurt.	0.755	0.736	0.419	33.594	31.734	1.860
Interview:Resum.:Meeting	0.807	0.710	0.433	37.355	27.660	9.695
Leave Not.:Not.:Admin	0.705	0.678	0.406	29.931	27.255	2.676
Market.:Corp.Announce.:Corp.Alloc	0.721	0.695	0.405	31.616	28.997	2.619
Meeting:Present:Planning	0.739	0.717	0.404	33.486	31.238	2.248
Org.Elect.:Corp.Announce:Admin	0.646	0.665	0.428	21.836	23.746	-1.911
Public.:Report:Proposal	0.630	0.585	0.429	20.082	15.604	4.478
Perf.Assess.:Report:Meeting	0.630	0.612	0.427	20.314	18.499	1.814
Report:Tech.Doc.:Publication	0.705	0.630	0.420	28.413	20.925	7.488
median	0.722	0.668	0.511			
std. dev.	0.057	0.043	0.052			
			rank sum	1027.000	1027.500	870.500
			Z-val	6.177	6.100	3.304
			p-value	6.515E-10	6.148E-10	9.534E-04
			A value	0.967	0.952	0.768

Table 4.9: Results for four-topic experiments: Email Topic(first col.), MCC Score (two to fourth col.), differences in percentage between the extended AISEC version with Naive Bayes (fifth col.), the SRN with Naive Bayes (sixth col.) and the extended AISEC version with SRN (seventh col.).

Topics(1st:2nd:3rd:4th)	Combined MCC Score			Diff.(A)	Diff.(B)	Diff.(C)
	Extended AISEC	SRN	Naive Bayes			
Acc.Syst:Accomp:Delivery:Act.List	0.624	0.549	0.578	4.556	-2.862	7.418
Admin.:Alumni:Appoin.:Worksh.	0.491	0.492	0.426	6.486	6.544	-0.058
Bank.:Baseb.:Bulle.:Car.Oppurt.	0.547	0.470	0.407	13.996	6.313	7.683
Course:Conf.:Contr.:Conv.	0.529	0.508	0.417	11.274	9.162	2.112
Corp.Alloc.:Corp.Announce.:Deliv.:Event	0.595	0.521	0.407	7.497	0.067	7.430
Financ.Forum:Hol.Deals:Intern.	0.645	0.513	0.412	13.321	0.096	13.225
Interv.:Job Post.:Leave Not.:Markt.	0.647	0.531	0.414	13.316	1.667	11.649
Mtg.:Not.:Onl.Shop:Org.Elect.	0.637	0.590	0.413	12.339	7.705	4.634
Perf.Assess.:Pro.Plan.:Present.: Press Rel.	0.663	0.602	0.427	13.675	7.500	6.175
Propos.:Publ.:Recrut.:Report	0.650	0.607	0.413	13.717	9.448	4.269
Reserv.:Resign.:Resume:Semin.	0.610	0.584	0.410	10.018	7.443	2.575
Ser.Agree.:Symp.:Tech.Doc.:Train. Sche.	0.662	0.588	0.418	14.355	6.966	7.389
median	0.630	0.540	0.414			
std. dev.	0.053	0.047	0.048			
			rank sum	211.000	185.000	196.000
			Z-val	3.494	1.994	2.627
			p-value	4.763E-04	4.620E-02	8.616E-03
			A value	0.924	0.743	0.819

of statistical non-parametric analysis, the p-value indicates with 95% confidence that there was a statistically significant difference between the MCC value of the extended AISEC version and the MCC value of the baseline approaches. In summary, these MCC value samples had a different distribution which indicates that their medians were different. The A value from the Vargha-Delaney A statistics shows that a comparison of the extended AISEC version with the baseline algorithms showed a large effect size with the A value above 0.71[4]. This indicates that the performance between a comparison of the extended AISEC version with the baseline approaches shows a large effect when tested on the email corpus of Enron Corporation for single-topic classification.

As shown in Table 4.7 for two-topic experiment, the extended AISEC produces better MCC score compared with Naive Bayes for overall topic combined. However when compared with SRN (in column"diff(C)% ") the extended AISEC produces better MCC score in 22 out of 30 topics combined. The SRN produces better MCC score compared with Naive Bayes in 28 out of 30 topics combined. In terms of classification measurement, the Naive Bayes shows a middle range of MCC value of 0.5 compared with the extended AISEC version and the SRN approach in which the median values for MCC scores were above 0.6. However, the extended AISEC version showed a good performance in the two-topic prediction compared with the SRN approach. Moreover, in all cases, the Mann-Whitney-Wilcoxon (or rank-sum) test and the Vargha-Delaney A statistic showed that the differences between each pair of these algorithms were statistically significant respectively. This can be summarized as showing that the tested algorithm's MCC value had different distributions and showed a large performance concerning effect size with an A value above 0.71.

In the results for three-topic and four-topic scenarios, as can be seen in Table 4.8 and Table 4.9, the difference was even larger. The extended AISEC version produces better MCC score compared with the Naive Bayes for overall topic combined. However, when compared with SRN, the extended AISEC produces better MCC score in 23 out of 26 topics (for three-topics) and 11 out of 12 topics for four-topics experiment. The Mann-Whitney-Wilcoxon and Vargha-Delaney A test values indicated further confidence in the comparison. To summarize, in the multiple-topic experiment, the extended AISEC version performed better compared with the SRN and Naive Bayes, while the SRN performed better compared with the Naive Bayes and, as Figure 4.8 ((second to bottom graph) reveals, this gradation was consistent throughout most of the evaluation periods.

Figure 4.8 summarizes the results of the single topic, two-topic, three-topic

[4]for a complete list of A value and its description, see Table 4.3 on Section 4.3.1

and four-topic experiments. In detail, the x-axis of the graph shows the topic of interest during each evaluation period and the y-axis shows the combined MCC value (for the two-topic to four-topic experiments). The graph is plotted as a continuous line because a single user profile adapts continuously throughout all the evaluation period. From Figure 4.8, it can be seen that after a first topic, the performance of the extended AISEC and the baseline approaches drop substantially. One reason for this is possibly the profile's high inertia. The populated profile with semantic word vectors in the email body may include thousands of terms and hence global competition has a minor effect on the weight of the existing terms. However, through the learning capability in the extended AISEC algorithm with the introduction of hypernym-hyponym relationship in WordNet, the algorithm is able to allow the profile to forget a no-longer-interesting topic and learn a new topic of interest effectively. This modification significantly improves the profile's adaptability.

4.5 Summary

This chapter has presented analyses of the ability of artificial immune systems (AIS) to achieve classification based on an email corpus. The experiment was divided into two main tasks: binary classification problem (discriminating between interesting or not interesting emails) and classification of multi-topic of emails categories. The task of email classification is widely used as an experimental platform for exploring the effect of changing user interests. The experiment for AIS in the classification of emails was based on Secker's algorithm [9] in which an artificial immune system was developed for email classification (AISEC) which classifies emails as interesting or uninteresting according to the subject and sender of the email. Section 4.1 explained the AISEC in detail. We developed an extension of AISEC which focused on enhancing the diversity of the *gene library* (the set of all words from feature vectors of B cells that recognise uninteresting emails) through the integration of WordNet in the algorithm. WordNet was used in the algorithm to be able to extend the gene library based on hypernym-hyponym relationships. In the original AISEC version, the algorithm extracted words only from the email subject and sender fields. This reduced the diversity generated by cloning and mutation, since, when mutation is performed, a word from this library replaces a word from a B cell's feature vector. In order to recognise new topics of interest and bring about the removal of existing topics of interest, a large library of words is needed, so we modified the algorithm to consider the body of the email in addition to the email subject and sender field, as this pro-

vides a richer set of words. This extension of the AISEC is described in Section 4.2. The remaining sections in this chapter presented analyses of the results of the experiment: Section 4.3 presented an analysis of the results of binary classification problem based on the concept of interesting emails. This experiment classified emails as interesting or uninteresting for the user and placed them in an appropriate folder. An interesting email in this context was one which was related to a PROJECT, for example emails on meetings, presentations, reports, resumes and interviews. Section 4.4 presented the results of the experiment on the classification of multiple email topics. For both experiments, a statistical analysis was carried out based on the non-parametric Mann-Whitney-Wilcoxon or rank-sum test [125] to test whether the studied algorithm's performance had different distributions (each having a different median), and the Vargha-Delaney A statistics [126] to measure the effect size between these algorithms. A detailed explanation of these tests is given in Section 4.3.1. To evaluate classification performance, we calculated the true positive (TP), false positive (FP), true negative (TN) and false negative (FN) count. We then used traditional predictive classification accuracy and Mathew's Correlation Coefficient (MCC) [127] measure. Section 4.3.2 offered a fuller description of these classification measurements.

In the experiment of binary classification problem (discriminating between interesting or not-interesting email), we evaluated the differences in performance between our extended AISEC and the original version of AISEC. The results showed that the extended AISEC exhibited an increase in accuracy and MCC value. Although there was a slight increase in accuracy and MCC value for the extended version, it was not obvious and the algorithm still performed a better classification. On the other hand, the A value from the Vargha-Delaney A statistics shows that there was a difference in the performance of the two algorithms when tested on a subset of the Enron Corporation email corpus. In this experiment, the extended AISEC performed at least as well as Secker's original algorithm on this dataset. The extended version was also capable of continuous learning and of potentially tracking changes in email. One explanation for this could be that the introduction of the WordNet corpus and consideration of the body of email helped to further improve the capability for identifying new and potentially uninteresting emails. The analysis of this part of the experiment only looked at the B cells' ability to reject uninteresting email. We then carried out a further analysis of the extended version of AISEC to determine its ability to classify emails into multiple categories, whilst adapting to changes in user interest. To more accurately simulate a real situation, we initially performed experiments in which each user profile had to represent more than one topic in parallel and

adapt to both modest and radical variations in these topics. We were particularly interested in these multiple-topic experiments because they more accurately reflect a real situation. Moreover, the experiment intended not only to investigate the performance of the algorithm to represent multiple topics in parallel, but was also to demonstrate the ability of the algorithm to forget a previous topic when it starts to process a new topic. The analysis of this part of the experiment is described in Section 4.4. The experiments were conducted as comparative experiments between the extended AISEC with two chosen baseline approaches, namely the Naïve Bayes and the Simple Recurrent Network (SRN). We adopted the same evaluation procedure which was based on non-parametric statistical analysis and predictive accuracy measurement based on Mathew's Correlation Coefficient (MCC). Since the experiment involved multiple email topics, we calculated the profile's MCC score as a combined MCC score which was calculated based on all documents relevant to the topics in a pair, or a triple. The results of this experiment confirm our hypothesis that: "The extended version of AISEC shows a good performance in the classification of multiple email topics". The extended AISEC version performed better compared with the SRN and Naive Bayes and this gradation was consistent throughout most of the evaluation periods. The results show that the antibody-antigen interaction of B cells and the introduction of the WordNet corpus and consideration of the body of email to help further improve the capability for identifying new and potentially uninteresting emails have a positive effect on the adaptability of the profile to changes in user interest and on the profile's response to the email corpus.

Figure 4.8: Comparative Experiments on Multiple Email Topic Classification: Single-topic (top), Two-topic (second), Three-topic (third), Four-topic (bottom) cases.

CHAPTER

FIVE

EXPERIMENTING WITH THE EXTENDED AISEC'S PARAMETERS USING SENSITIVITY ANALYSIS

In Chapter 4, the AISEC algorithm was described, tested and evaluated. The experiment was based on a version of the AISEC algorithm which had been extended from [9, 10]. The experiment was implemented in two main tasks. The first task of the experiment was to evaluate the performance of the AISEC version in the classification of 'interesting' and 'not interesting' emails. From the experiment, it can be summarised that the extended AISEC performed at least as well as the original AISEC in single-topic classification (the binary classification problem). To more accurately simulate a real situation, we initially performed experiments in which each user profile has to represent more than one topic simultaneously. Moreover, the experiment was not only to investigate the algorithm's performance in representing multiple topics simultaneously, it was also to demonstrate the ability of the algorithm to forget a previous topic that was no longer of interest when it starts to process a new topic. The results of the tests showed that a clonal selection based AIS algorithm could perform the classification of emails with a higher MCC score compared with the Naive Bayes algorithm and the SRN algorithm. Moreover, the gradation of the algorithm's performance was consistent throughout most of the evaluation periods, showing that the extended AISEC worked well at a continuous classification task and was

able to forget a previous topic that was no longer of interest. This can be summarised as that the antibody-antigen interaction of B cells in clonal selection with the introduction of the usage of WordNet had a positive effect both on the profile's adaptability to changes in interest and on the profile's response to the email corpus.

The dynamic behaviour of AISEC can be controlled by the algorithm's parameters, of which there are many. Therefore, there is some need to examine the influence of the algorithm's parameters on the performance of the algorithm. This chapter describes the potential usefulness of the sensitivity analysis technique in interest classification based on email. Moreover, the results from the sensitivity analysis give insights on how these parameters can be optimised to provide a better performance for each of the performance metrics. The implementation of sensitivity analysis discussed in this chapter is divided into two main tasks: single-topic classification and multiple-topic classification. The objectives of these experiments are:

1. To see the trend (effect of changes in threshold value) for each different parameter with on-line classification of changing topics.

2. To identify on what level the algorithm becomes reliable and acceptable within the scenario of extreme cases and the scenario of mild cases.

3. To identify whether there is a need to extend (add) a memory detector based on parallel experimentation with a group of specific thresholds (upper threshold, middle threshold and lower threshold).

The work described in this chapter focuses on the analysis of the extended AISEC's parameters by investigating the effect on the *false positive rate* (FPR), *false negative rate* (FNR) and the *predictive accuracy*. In addition to investigating the effect on AISEC's performance associated with the difference between the default parameter values and the optimised parameter values, a non-parametric statistical analysis was carried out based on the value of Vargha-Delaney A statistics [126]. An overall conclusion was reached on the basis of this analysis. In the next section, the implementation of sensitivity analysis on interest classification is described. This includes the concept of sensitivity analysis in general and the experiment protocol of the analysis.

5.1 The Implementation of Sensitivity Analysis for Interest Classification

Sensitivity analysis estimates the rate of change in the output of a model caused by changes in the model inputs. It is mainly used to determine which input parameter is more important or sensible to achieve accurate output values [141]. Sensitivity analysis is a good technique for:

1. Evaluating the applicability of a model,
2. Determining the rate of change in the output of a model with respect to changes in parameters
3. Understanding the behaviour of the system being modelled.

It has been applied in various fields, including complex engineering systems, economics, physics, social sciences, risk assessment and many others [142, 143]. Frey and Patil in [142] classified sensitivity analysis methods into three categories: mathematical, statistical and graphical. Mathematical methods assess the sensitivity of a model's output to the range of variation of an input. It typically involves calculating the output for a few values of an input that represent the possible range of the input. Examples of techniques for mathematical methods include nominal range sensitivity analysis, break-even analysis, difference in log-odds ratio and automatic differentiation. The statistical method involves running simulations in which inputs are assigned probability distributions and then assessing the effect of variance in inputs on the output distributions. This method allows the user to identify the effect of interaction among multiple inputs. Examples of the statistical method are regression analysis, analysis of variance, response surface methods and mutual information index. Finally, the graphical method gives representations of sensitivity in the form of graphs, charts or surfaces. Generally, this method is used to give a visual indication of how an output is affected by variation in inputs.

In this study, the sensitivity analysis on the parameters was based on the graphical method. The context of the parameters used in this experiment has been explained in Section 4.1.2. In the experiment, variations in parameter input will be given for each of the parameters tested. A chart will be presented showing the behaviour of the parameters and a summary will be given based on the description on the observed parameters.

5.2 Analysis of Extended AISEC parameters in Single Topic Classification

Experiment Objectives: The objective of this experiment is to examine the effect of changing the value of the parameters associated with on-line classification of emails in single-topic classification and to evaluate the corresponding changes of the performance metrics.

This section presents an analysis of the extended AISEC's parameters in single topic classification by investigating the effect on the *false positive rate* (FPR), *false negative rate* (FNR) and the *predictive accuracy*. A single topic classification means that the emails are classified into 'interesting email' or 'not interesting email'. It is known as the binary classification problem. In addition to investigate the effect on AISEC's performance of the difference between the default parameter values and the optimised parameter values, a non-parametric statistical analysis was carried out based on the value of Vargha-Delaney A statistics [126]. Results from this experiment show that the values of the parameters can be changed to achieve the desired performance.

As stated in Section 4.1.2, there are seven major parameters in the AISEC algorithm:

- Kc **(classification threshold)**
- Ka **(affinity threshold)**
- Kl **(a constant which controls the rate of cloning**)
- Km **(a constant which controls the rate of mutation**)
- Ksb **(an initial stimulation count for Naive B cells**)
- Ksm **(an initial stimulation count for memory B cells**)
- Kt **(an initial number of memory cells generated during initialisation)**

To summarise, the classification threshold (Kc) influences decisions on the class of incoming emails. If the antigen (email) shows affinity for any immune cell higher than this threshold, the message is classified as uninteresting. The affinity threshold (Ka) influences the selection of immune cells for reward or punishment, depending on the classification results. The cloning constant (Kl) determines the maximum number of clones a cell may produce. While, the mutation constant (Km) determines the number of times a mutation will occur to a cloned cell. The Naive B cell stimulation level (Ksb) and memory cell stimulation

Parameter	Default Value	Range
Kc (Classification Threshold)	0.3	0 - 1
Ka (Affinity Threshold)	0.5	0 - 1
Kl (Clone Constant)	3.0	≥ 1
Km (Mutation Constant)	0.3	0 - 1
Ksb (Naive B Cell Stimulation Level)	175	0 - size of test set
Ksm (Memory B Cell Stimulation Level)	80	0 - size of test set
Kt (Initial Number of Memory cells)	20	0 - size of test set

Table 5.1: Details of the Extended AISEC Parameters that will be Used in the Sensitivity Analysis

level (Ksm) influences the potential life span of a B cell and a memory cell respectively. Finally, the initial memory cell set size (Kt) is used to determine how many memory cells will be selected from the initialisation data set in the initialisation phase. Table 5.1 gives details of the default values of the parameters that will be investigated. These parameters will be also used in the sensitivity analysis of the parameters for the multiple-topic classification.

During the analysis, there is an initial (low) value, an end (high) value, an increment value, and a default value. The column 'Default' in Table 5.2 shows fixed default values for each parameter when that parameter is not being varied. For each parameter, the parameter value is varied and the results are compared qualitatively and quantitatively. The complete list of the value ranges of the extended AISEC parameters is presented in Table 5.2. The value in the 'Default' column is a value known as a baseline value. The 'Values' column shows the number of different values tested for each parameter. During each of the experiments, only one parameter value was varied, whilst the others were set to a default value. For example, when analysing the parameter Kc, there are 50 possible values of Kc from 0.02 to 1 based on the increment of 0.02, whilst the other parameters used a default value; 0.5 for Ka, 0.3 for Km, 3.0 for Kl, 175 for Ksb, 80 for Ksm and 20 for Kt.

The emails used as the data set in this experiment is from the email corpus of the Enron Corporation[1]. The data set were selected at random and it is independent from the one used in the previous experiment. Collected emails were separated into two groups, one for an initialisation set and one for a running set. For the initialisation set, 400 uninteresting emails were used. The actual initialisation set used for each experiment was constructed by extracting the required

[1]http://www.cs.cmu.edu/~enron/

Parameter	Start	Increment	End	Default	Values	Total Run
Kc	0.02	0.02	1	0.3	50	3700
Ka	0.02	0.02	1	0.5	50	3700
Km	0.02	0.02	1	0.3	50	3700
Kl	1	1	25	3.0	25	1850
Ksb	10	10	500	175	50	3700
Ksm	2	2	100	80	50	3700
Kt	1	1	25	20	25	1850

Table 5.2: The value ranges for the Extended AISEC's parameters

Test Set	Interesting Emails	Uninteresting Emails	Total
Initialisation	0	400	400
Running	250	250	500
Total	250	650	900

Table 5.3: Number of Emails Used for Initialisation and Running in Single Topic Classification

number of uninteresting emails from this set of 400 emails. Table 5.3 gives detailed information of the data set used.

Each experiment was run 74 times and the result for the experiment were analyzed in terms of the median for each performance measure. Once a test has been run and the confusion matrix has been computed, using the metrics of FPR (Equation 5.1) and FNR (Equation 5.2), the measure of predictive accuracy is also possible to calculate based on the Equation 4.1. These performance metrics are used throughout this experiment of sensitivity analysis on parameters. Our aim in terms of classification performance is to maximize the predictive accuracy and minimize the FPR and the FNR. During the experiment, the optimal value for each of the parameters will be selected and is further analysed based on the Vargha-Delaney statistics [126] to measure the effect size between the results based on the default value versus the results based on the optimal value for each of the parameters.

$$FPR = \frac{FP}{FP + TN} \tag{5.1}$$

$$FNR = \frac{FN}{FN + TP} \tag{5.2}$$

5.2.1 Experimental Result and Analysis

This section explains the results of the influence of the extended AISEC's parameters and its performance in the on-line classification of emails based on single-topic classification. The experiment was carried out based on the protocol explained previously. For each of the parameters, the hypothesis is given based on the parameter description given in Section 4.1.2. For each parameter, a chart is shown detailing the behaviour of the algorithm as that single parameter varied. A conclusion is then reached based on the chart to either confirm or refute the hypothesis.

The *classification threshold (Kc)* parameter. The Kc parameter is defined as the algorithm's tolerance level to identify uninteresting emails. A low Kc may allow a low affinity antigen (representation of an email with unknown class) to be classified as uninteresting (positive classification). However, if the level is set too high it might lead the immune cells to wrongly predict the negative class for the antigen (false negative classification). Figure 5.1 shows that at a low Kc value such as 0.02, the FPR level was 50% which is comparatively high compared with the FNR value of less than 15% . As the level of Kc increased, the situation became inversed; FPR decreased towards a low percentage of less than 15% while FNR increased towards a high percentage of 30% . Furthermore, the distribution of the predictive accuracy shows an increase as the Kc level increased, and as the value of Kc increased, the accuracy reached a level of 78% . It can be seen that this parameter had rather a large effect on accuracy, which varies from approximately 60% to 85%, depending on the value of Kc. This is a large variation compared with most other parameters, therefore the value of Kc should be chosen with care. It is clear from Figure 5.1 that the optimum setting for this parameter with regard to accuracy is between Kc = 0.44 and Kc = 0.46.

Affinity Threshold (Ka). The Ka parameter is responsible for population manipulation and is dynamic throughout the algorithm. It influences the selection of immune cells for reward or punishment, depending on the classification results. A low Ka value increases the chance of generating more B cells in the initialisation phase, which may increase the population size. However, as more classifications occur, a low Ka value may allow an increased number of cells to be removed from the population, thus leading to a reduction in cell population. In contrast, high Ka values allow only high affinity cells for the misclassified antigens to be removed from the algorithm. This may encourage the population

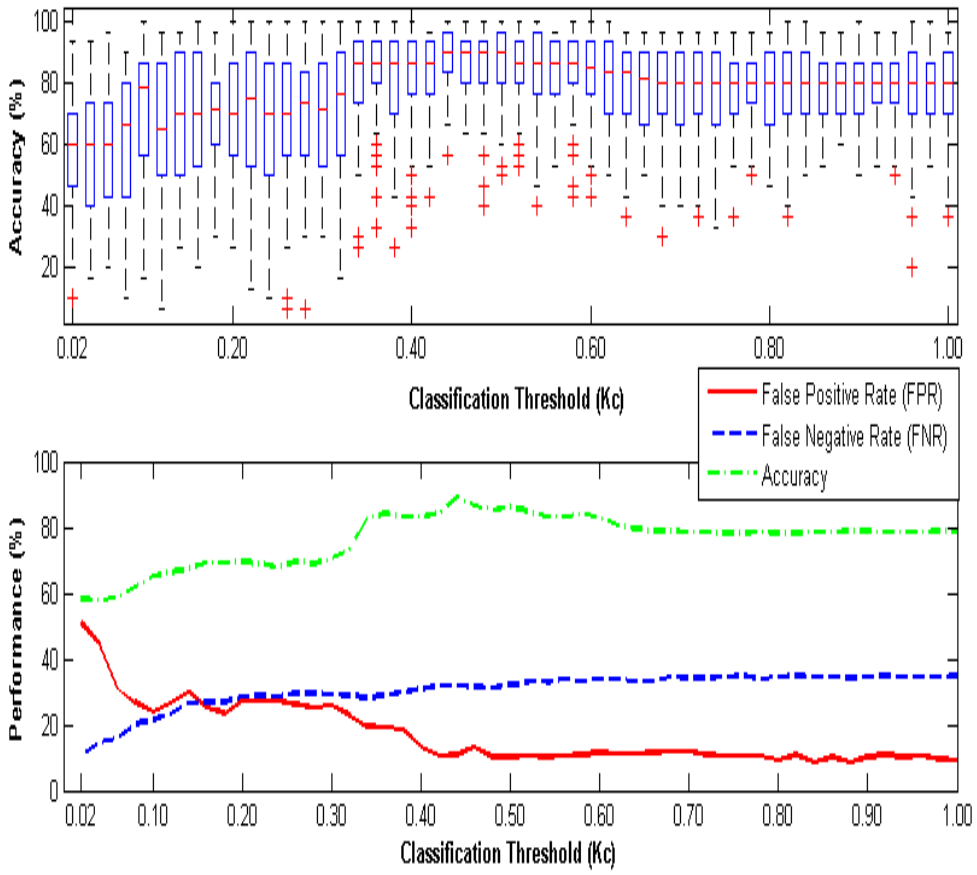


Figure 5.1: Influence of the Classification Threshold (Kc) Parameter

to grow in the long run, resulting in an increase in false positive rate. Figure 5.2 shows that FNR decreases from above 20% to below 10% as the Ka value increases. Inversely, FPR increased from 9% to around 18% as the Ka value increases. The Ka values shows an inversely proportional rate as the value of the parameter increased. As shown in Figure 5.2, all observations appear fairly stable at Ka value around 0.34. Like Kc, Ka also had an effect on the overall accuracy of the algorithm with its value ranging again from approximately 60% to just under 85% . It is believed that the Ka parameter at a certain level (around 0.34 as in the chart), disables the factors that may affect performance, such as refining cells, by removing them in a false positive classification. In other words, there were no more B cells that were close enough to the antigens representing interesting emails with affinities between them greater than the Ka value.

Clone Constant (Kl). The Kl parameter determines the maximum number of clones a cell may produce. High Kl may result in an increase in naive B cell pop-

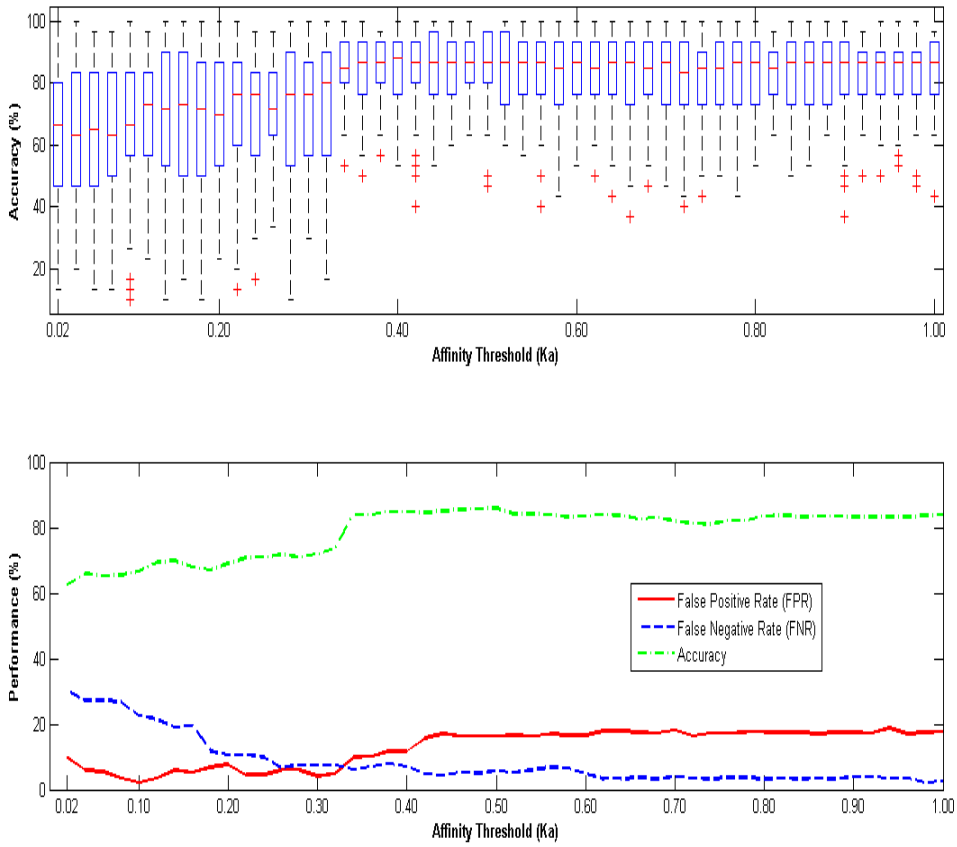


Figure 5.2: Influence of the Affinity Threshold (Ka) Parameter

ulation sizes as more clones may be generated. This might increase redundancy because clones are duplicates of the most competent cell, and the mutation applied in some cases may be insufficient to move the clone outside the area of antigen recognition associated with the affinity threshold. Figure 5.3 shows that the influence of Kl over classification performance was small compared with that of Kc and Ka. A particular reason for this is that the clones generated in the search space may not bring drastic changes in the algorithm's recognition as the existence of their parents provides a certain degree of accuracy. However, an increase in Kl value created a situation more likely to be classified as positive, where there are a slight increase of FPR and the effect of reducing the FNR.

Mutation Constant (Km). The Km parameter is responsible for the rate of mutation. High Km may result in an increase of diversity in immune cells by generating more variants from selected cells. B cells diversity is achieved by a high rate of mutation and it may contribute to a slightly increased false positive classi-

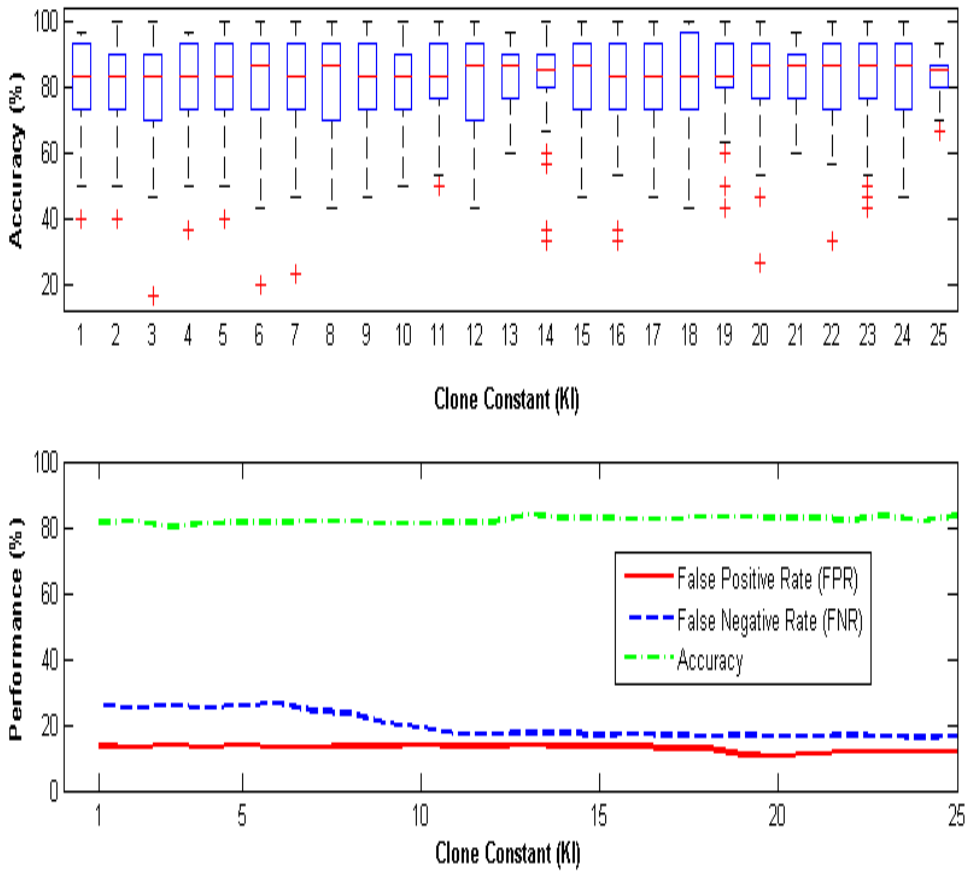


Figure 5.3: Influence of the Clone Constant (Kl) Parameter

fication rate, however, the difference is insignificant. This may be due to the case where unknown emails may contain not only uninteresting emails but also interesting emails, so there is still the possibility that the FPR may increase. Figure 5.4 shows that the B cells diversity achieved by a high rate of mutation contributes to a slightly increased false positive classification rate. However, the difference is insignificant. This suggests that the influence of the Km parameter is not as significant as the other parameters over the predictive performance.

Naive B Cells Stimulation Level (Ksb). The Ksb parameter influences the potential life span of naive B cells. High Ksb may allow non-stimulated B cells to survive longer, which results in an increase in the overall cell population size. However, allowing the non-stimulated B cells to live too long will allow FPR to increase. Upon a positive classification, it may accelerate B cells death in general. Some selected B cells can be stimulated to correctly classified antigen, but even in this circumstance, other cells are to be suppressed. The decrement of B cells

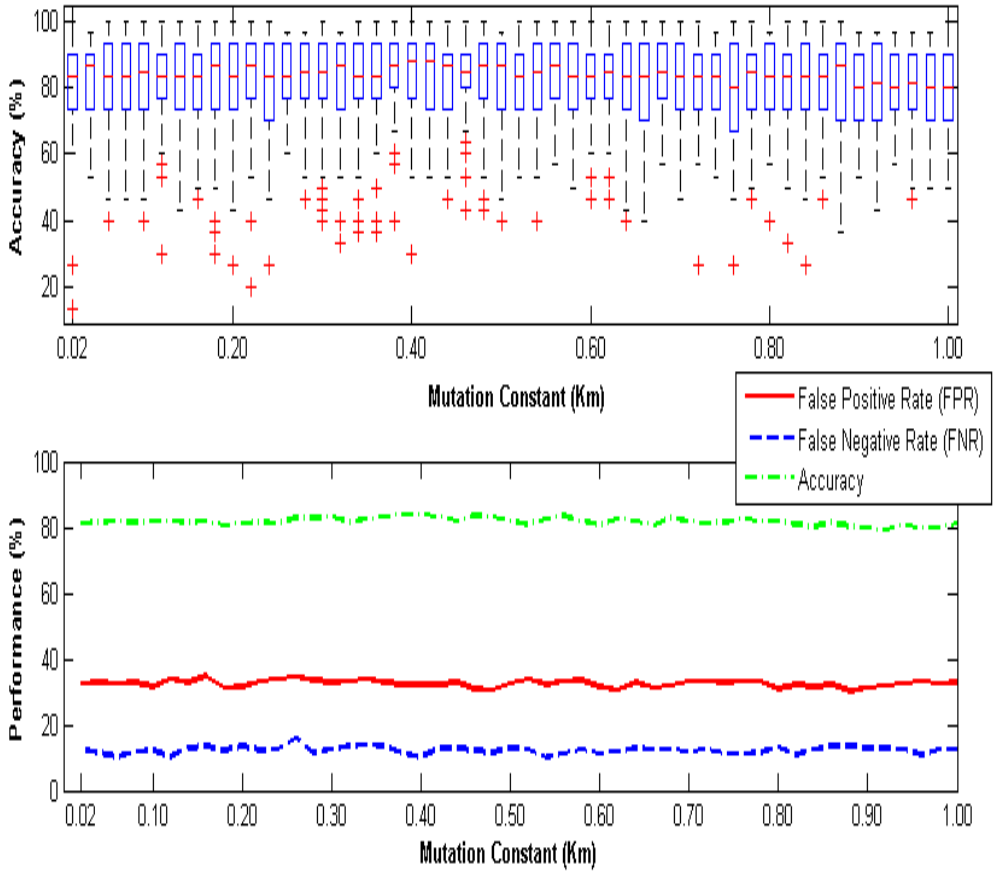


Figure 5.4: Influence of the Mutation Constant (Kl) Parameter

stimulation count is applied after a classification, regardless of the classification being correct or wrong. Thus, a high rate of positive classifications and a low false positive rate will be the ideal combination for drastic cells death. According to Figure 5.5, the influence of Ksb does not seem significant; it seems that there were no noticeable correlations between Ksb and the metrics measured.

Memory Cell Stimulation Level (Ksm). This parameter influences the potential life span of a memory B cells. High Ksm allows memory cells to survive longer. However, allowing the memory cells to live longer, may allow FPR to increase because more emails may be recognised and classified as uninteresting messages by old memory cells. As with Ksb, a high Kc reduces the chance of positive classification, which also decreases the chance of Ksms involvement, and vice-versa. Therefore, the influence of Ksm is considered to be more prominent with a low Kc than with a high Kc. Figure 5.6 shows that the influence of Ksm was not significant in general. The predictive accuracy, the FNR and the FPR appear

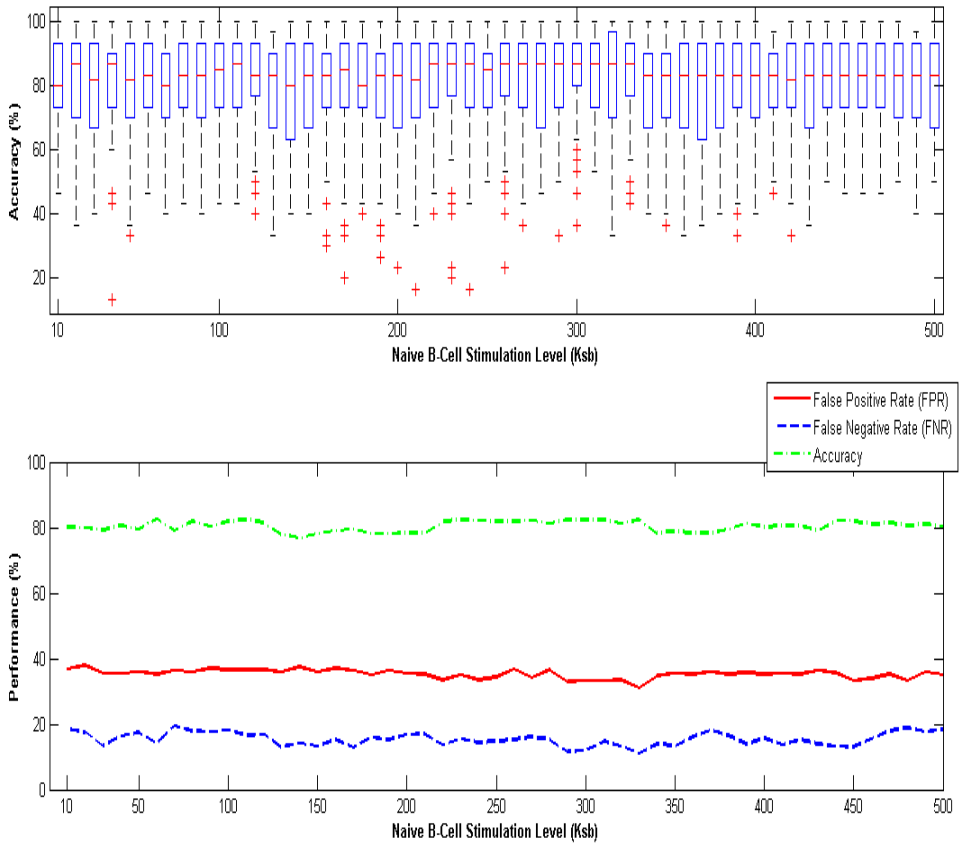


Figure 5.5: Influence of the Naive B Cells Stimulation Level (Ksb) Parameter

stable for all values of Ksm tested.

Number of Memory Cells Generated by Initialisation (Kt). The Kt parameter determines how many memory cells will be selected from the initialisation data set in the initialisation phase. If Kt has a low value, then it is more likely that in all attempts to match a memory cell, mc and a 'tc' in the initialisation phase, the affinity function returns a value less than Ka. In this case, no clones will be produced and the resultant B cells set will be empty. Therefore, a high Kt may guarantee that more B cells are produced during the initialisation, compared with a low Kt value. An interesting observation from the experiment is that when Kt was high, the resultant B cells set was sparse. This is because given that there was a large number of memory cells, then there was a small number of cells left in the initialisation set. Figure 5.7 shows the effect of $Kt's$ performance whereby, with the changes in the value of Kt, it shows that the predictive accuracy appear stable. As the FNR and FPR do vary as the Kt value changes, the value should

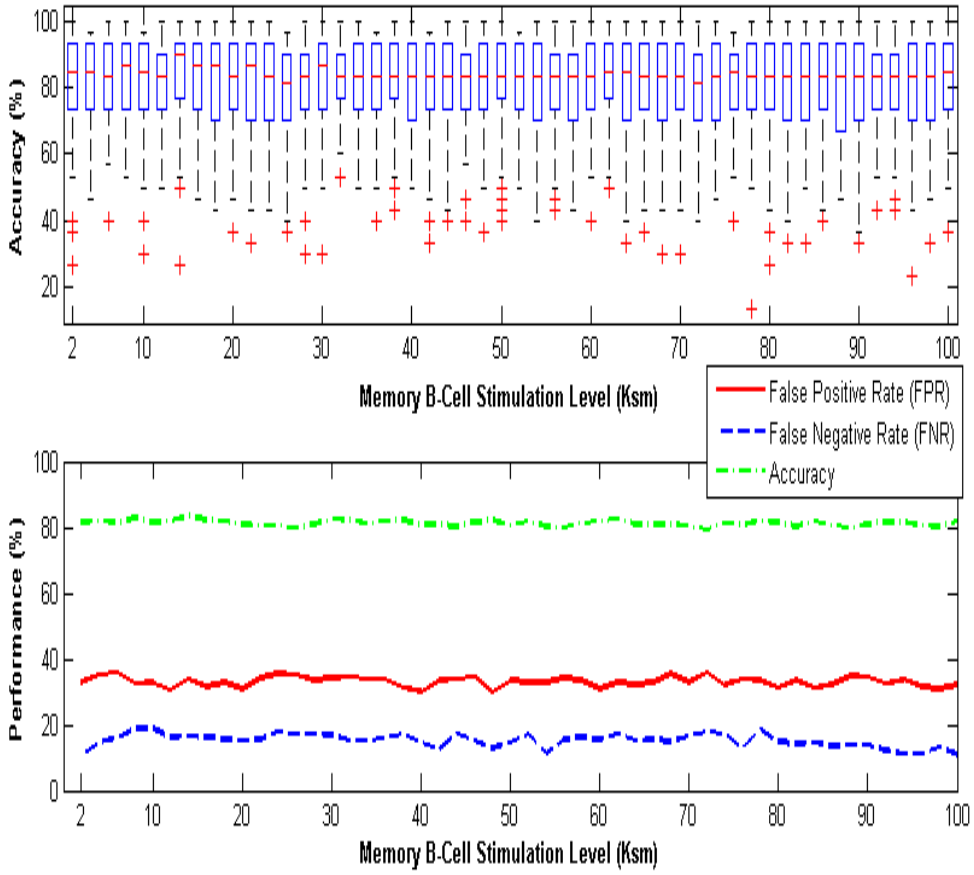


Figure 5.6: Influence of the Memory B Cells Stimulation Level (Ksm) Parameter

once again be chosen on the basis of whether FPR or FNR needs to be minimised or maximised for the particular task. Figure 5.7 suggests that influence of Kt was significant compared with Kl, Km, Ksm and Ksb. The FPR started low but gradually increased, suggesting that more memory cells were produced. However, the FNR was affected by this situation as well, and given a value of $Kt = 1$, a large number of emails were misclassified as negative. It seems that this FNR value was biased by the extreme situation and the single memory cell did not recognize any antigen during the running phase thus shows 20% of false positive rate as observed in Figure 5.7. This provides evidence that the algorithm became vulnerable to bias by the selection of initial memory cells when initial memory cell size was small. This bias can be seen as a huge FNR value for low values of Kt.

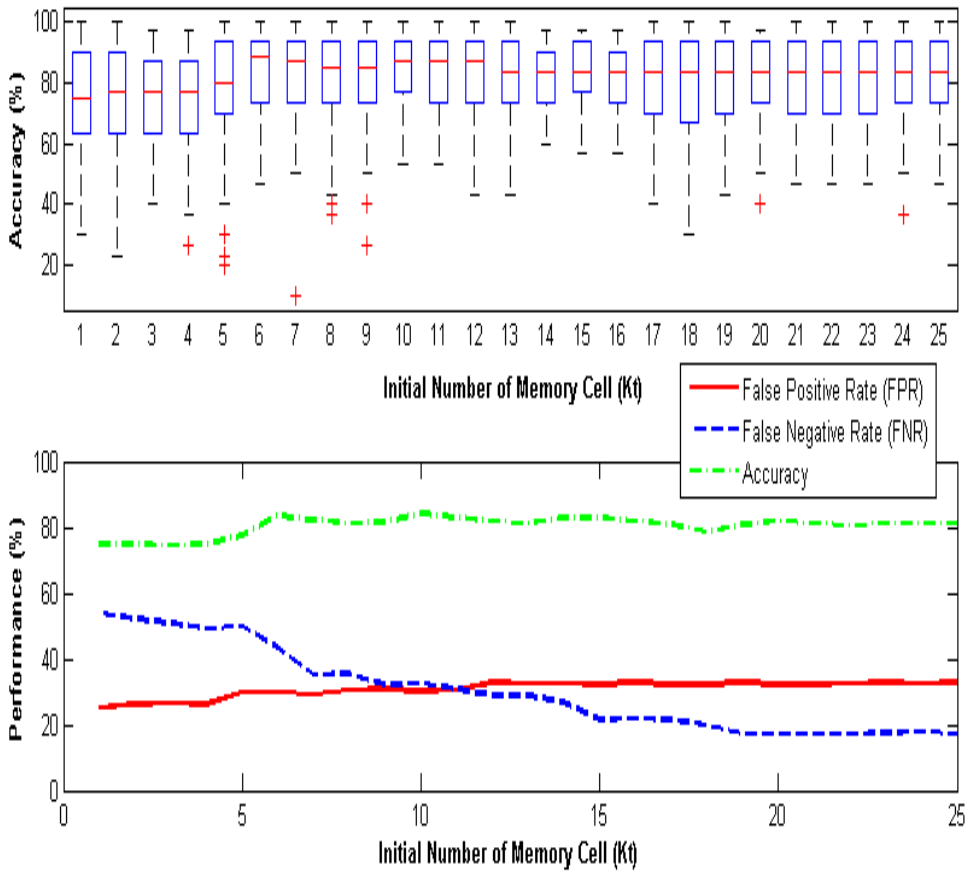


Figure 5.7: Influence of the Initial Number of Memory Cell (Kt) Parameter

5.2.2 Summary of the Experiment and the Assessment of the Optimised Parameters

In the previous section, an analysis of the extended AISEC's parameters in single-interest classification was presented. Based on the test, the optimised value for each of the parameters can be chosen as follows:

- The value of Kc is chosen to be 0.46 as according to Figure 5.1 this exhibits high predictive accuracy and gives a good trade-off with the false positive and the false negative rates.
- The Ka parameter is chosen as 0.34. This gives a better trade-off between the predictive accuracy and the false positive rate, which was fairly constant from 0.34 to 1. With e-mail classification, it is preferable to have a lower false positive rate, therefore a lower Ka is preferable as it may reduce the false positive rate at the expense of the false negative rate with an acceptable

situation.

- As shown in the chart for the influence of the Kl parameter, the predictive accuracy stayed roughly constant for all values of Kl. The value of Kl was chosen as 15. This value is quite different from the previous value (default value), however, the value chosen is the center of the legal range. Based on the justification for this parameter, the value of this parameter is not critical.

- Similarly, the value of Km does not appear to affect the predictive accuracy, FPR or FNR. The Km is chosen as 0.56. Therefore a value near the centre of the valid range is a reasonable choice.

- The value of Ksb has increased a little compared to the default value, since the new value chosen is 220. From Figure 5.5 the value of Ksb does not appear to affect the predictive accuracy, FPR or FNR. Thus, the value is chosen based on the center of range in order to avoid committing to too extreme value and to provide stability.

- For the Ksm parameter, the predictive accuracy and the FPR appear stable for all values of Ksm tested. The value chosen was 30, which is much lower than the default value.

- Finally, it can be summarised that an increasing Kt value slightly increased the accuracy, but at the expense of the FPR. Therefore the value of Kt was kept at 20 to maximise the benefit of this trade-off, although any value in the range of 12 to 22 would be sensible based on Figure 5.7.

With regard to the optimised value chosen, an assessment of the optimised parameters was carried out. First, the assessment was performed to evaluate the classification performance based on the measurement of the predictive accuracy and the MCC score. Second, the assessment was based on the value of Vargha-Delaney A statistics. Table 5.4 presents the set of parameter values used in the test. In Table 5.4, parameter Set A refers to the set of parameter values described in Table 5.1 and thus formed the basis of the investigation. Parameter Set B is a value described above in this section and known as the optimised parameter values.

The results of running the extended AISEC system 50 times using each parameter set presented in Table 5.4 are presented in Table 5.6, where the figures in brackets represent the standard deviations for the values. In this analysis, different random seeds were used. Collected emails were separated into two groups, one for an initialisation set and one for a running set. For the initialisation set,

Parameter	Set A	Set B
Kc	0.3	0.46
Ka	0.5	0.34
Kl	3.0	15
Km	0.3	0.56
Ksb	175	220
Ksm	80	30
Kt	20	20

Table 5.4: The set of baseline and optimised values for AISEC parameters

Test Set	Interesting Emails	Uninteresting Emails	Total
Initialisation	0	300	300
Running	200	200	400

Table 5.5: Number of Emails Used for Initialisation and Running in Comparison Analysis between Baseline and the Optimised Parameters

300 uninteresting emails were used. Table 5.5 gives detailed information of the data set used.

From Table 5.6, it can be summarised that the optimised parameter values did indeed increase the predictive accuracy as well as the MCC score of the test set. The increase in accuracy from parameter set A to parameter set B can be tested further for the size of the effect based on Vargha-Delaney A Statistics. From Table 5.6 it shows that the A statistic value specifically for the comparison of the default parameter value and the optimised value shows that the difference has a medium effect.

Parameter Set	Vargha-Delaney A Statistics	Classification Accuracy	MCC Score
Set A	0.64151	79.83% (2.83)	0.52
Set B		83.40% (3.10)	0.63

Table 5.6: Result of tests using the optimised parameters

Furthermore, Figure 5.8 shows that the A values for the parameter Kc, Ka and Kt show a large effect (with the A value above 0.714)[2] as the parameter value

[2]for a complete list of A value and its description, see Table 4.3 on Section 4.3.1

increased. However, as the parameter value approached the optimised value, the effect between the samples got smaller. However, it appears that for parameters Kl, Km, Ksb and Ksm, the effects were small, with A values less than 0.6. To summarise, the influence of Kl, Km, Ksb and Ksm does not seem significant; it seems that there are no noticeable correlations between these parameters and the metrics measured. This result supports our justification as explained in Section 5.2.1. Further results of the statistical analysis, including the p-value, Mann-Whitney-Wilcoxon or rank-sum test, and the Vargha-Delaney A statistics values can be seen in Appendices C. Table 5.7 summarises the effect of the extended AISEC parameters on the performance of single-topic classification of emails with respect to the predictive accuracy, false positive rate and false negative rate.

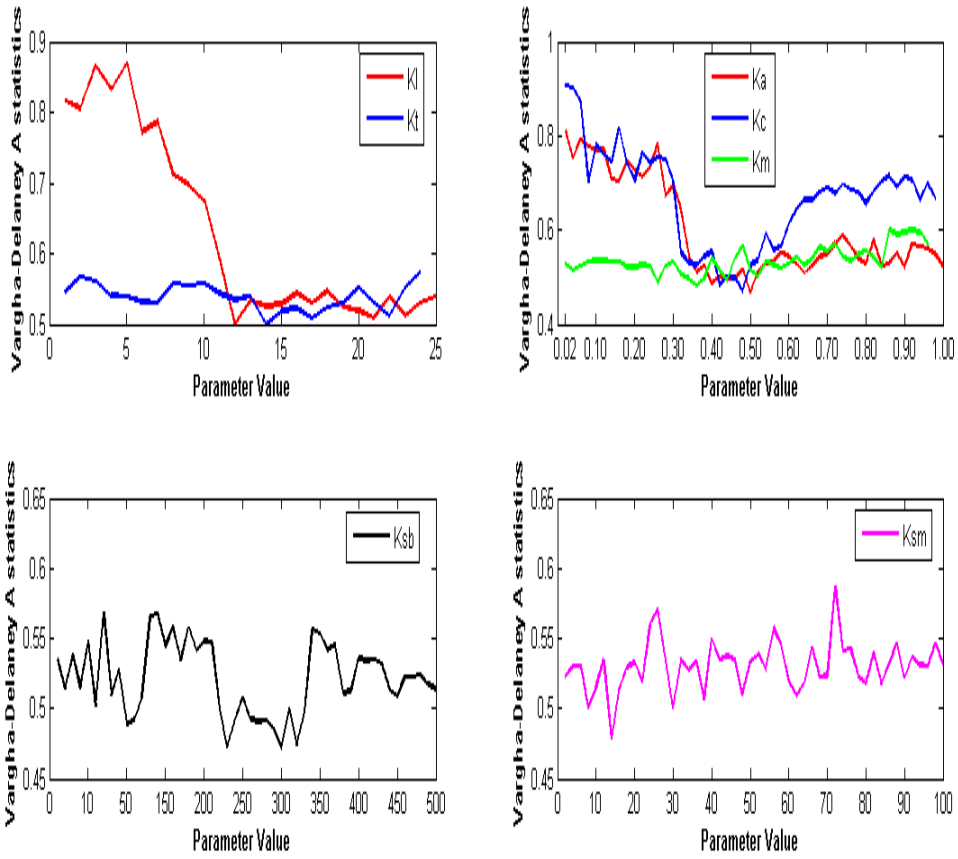


Figure 5.8: Vargha-Delaney A statistics of Optimised Parameter Values in the Single-Interest Classification Scenario

Parameter	Predictive Accuracy	FPR	FNR
Kc	Yes	Yes	Yes
Ka	Yes	Yes	Yes
Kl	Negligible	Little	Little
Km	Negligible	Little	Little
Ksb	Moderate	Little	Little
Ksm	Moderate	Little	Little
Kt	Yes	Yes	Yes

Table 5.7: A Summary of the Influence of the Optimised Extended AISEC Parameters

5.3 Sensitivity Analysis on the Effect of Parameters in Multi Interest Classification

So far, we have been looking at the influence of the extended AISEC parameters in single-interest classification. The results and a justification of the outcome have been presented in the previous section. According to the results of the investigation, the most influential parameters were the classification threshold (Kc) and the affinity threshold (Ka). With regard to the optimised values chosen, an assessment of the optimised parameters was carried. From the evaluation, the extended AISEC based on the optimised value set achieved higher predictive accuracy and MCC score in the running set when compared with the baseline value.

Experiment Objectives: The objective of this experiment is to identify at what level the algorithm becomes reliable and acceptable in the extreme case scenario and the mild case scenario.

In this section, a further investigation by sensitivity analysis on the extended AISEC's parameters is described. The purpose of the experiment was to investigate the influence of the parameters on multiple-topic classification. In this experiment, a different random seed was used. Like previous experiments, the collected emails were separated into two groups, one for an initialisation set and one for a running set. For the initialisation set, 300 uninteresting emails were used. Table 5.8 gives detailed information of the data set used.

The same analysis approach was used as described in the single-topic or called as binary classification that include the parameter description. Table 5.9 presents the parameter configuration for this experiment. The default value used in this experiment was different from the default value used in the single-topic classification.

In single-topic classification, an email is classified as either interesting email

Test Set	Interesting Emails	Uninteresting Emails	Total
Initialisation	0	300	300
Running	250	250	500

Table 5.8: Number of Emails Used for Initialisation and Running in Multiple-Topic Classification

or uninteresting email based on the user's interpretation of the term 'interesting email'. For the multiple-topic classification, we were interested in testing the influence of the algorithm's parameters in terms of its performance to represent multiple topics in parallel. It was also to demonstrate the ability of the algorithm to forget a previous topic when it starts to process a new topic. We conducted this test by considering the following different cases:

1. case 1: topics with large number of emails together with topics with low number of emails;
2. case 2: both topics had low number of emails;
3. case 3: both topics had high number of emails.

The reason for this was that the experiment will be much more meaningful if an investigation of each of the email's topics presented is compared. This is consistent with our experiment objective as stated above. Moreover, experimenting with a number of topics will lead to lots of data and graphs compared with experimenting by cases. Figure 4.6 illustrates the list of the email topics and their corresponding numbers of email. The next part of this section presents the results of the experiment and its justification. The behaviour of the algorithm as the parameter values varied will be charted. A conclusion will then be drawn which will either confirm or refute the behaviour of the parameters based on the explanation in Section 4.1.2.

5.3.1 Case 1: : Topics with Large Number of Emails with Topics with Low Number of Emails

The Influence of the Affinity Threshold (Ka) Parameter Tested on the Case 1 Scenario

Figure 5.9 shows that the behaviour of the parameter Ka in this case has an effect on the accuracy as well as the false negative rate. However, the FNR decreased

Parameter	Start	Increment	End	Default	Values	Total Run
Kc	0.02	0.02	1	0.50	50	3700
Ka	0.02	0.02	1	0.42	50	3700
Km	0.02	0.02	1	0.22	50	3700
Kl	1	1	25	3.0	20	1850
Ksb	10	10	500	140	50	3700
Ksm	2	2	100	50	50	3700
Kt	1	1	25	20	25	1850

Table 5.9: The Parameter Configuration Tested on Multiple-Topic Classification

from about 60% to below 10% as the Ka value increases. Inversely, the FPR has increased from 10% to around 19% as the Ka value increases. The increase, however, was not high. As shown in the Figure, all observations appeared fairly stable at a Ka value around 0.44, which had an effect on the overall accuracy of the algorithm from approximately 70% to just under 88%. It is believed that the Ka parameter at a certain level (around 0.44 as in the chart), disables the factors that may affect performance, such as refining cells by removing them in a false positive classification.

The Influence of the Classification Threshold (Kc) Parameter Tested on the Case 1 Scenario

The Kc parameter showed that, at a low Kc value such as 0.02, the FPR level was 50% which is comparatively high compared with the FNR value of less than 11% as depicted in Figure 5.10. Unlike the Ka, as the level of Kc increased, the situation became inversed; the FPR decreased towards a low percentage of less than 10% while the FNR increased towards a high percentage of 35% . It is clear from the Figure that the optimum setting for this parameter with regard to accuracy is between $Kc = 0.36$ and $Kc = 0.52$. It can be seen that this parameter had rather a large effect on accuracy, which varied from approximately 55% to 90% .

The Influence of the Clone Constant (Kl) and Mutation Constant (Km) Parameters Tested on the Case 1 Scenario

Figure 5.11 and Figure 5.12 shows that the influence of Kl and Km on multiple-topic classification in the Case 1 scenario was small compared with the Kc and Ka parameters. A particular reason for this is that the clones generated may not bring drastic changes in the algorithm as the existence of their parents provides

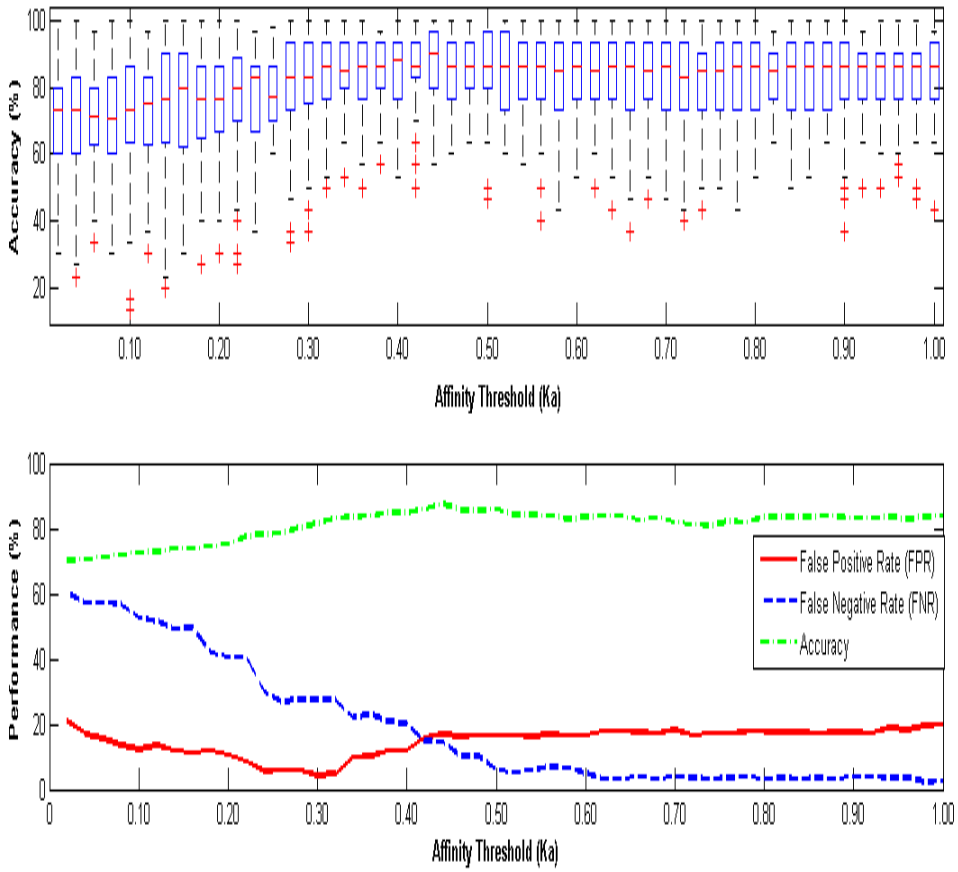


Figure 5.9: Influence of Affinity Threshold (Ka) Parameter for the Case 1 Scenario

a certain degree of accuracy. For the mutation rate, the B cells diversity achieved by a high rate of mutation contributed to a slightly increased false positive classification rate and its difference was insignificant.

The Influence of the Naive B Cells Stimulation Level (Ksb) and Memory B Cells Stimulation Level (Ksm) Parameters Tested on the Case 1 Scenario

The Ksb parameter influences the potential life span of naive B cells. High Ksb may allow the non-stimulated B cells to survive longer, which results in an increase in the overall cell population size. It was different for the Ksm parameter because this parameter influences the potential life span of memory B cells. High Ksm allows memory cells to survive longer in such a situation. According to Figure 5.13, the influence of Ksb does not seem significant; it seems that there were no noticeable correlations between Ksb and the metrics measured. This was similar for Ksm as depicted in Figure 5.14, where the result shows that the influence

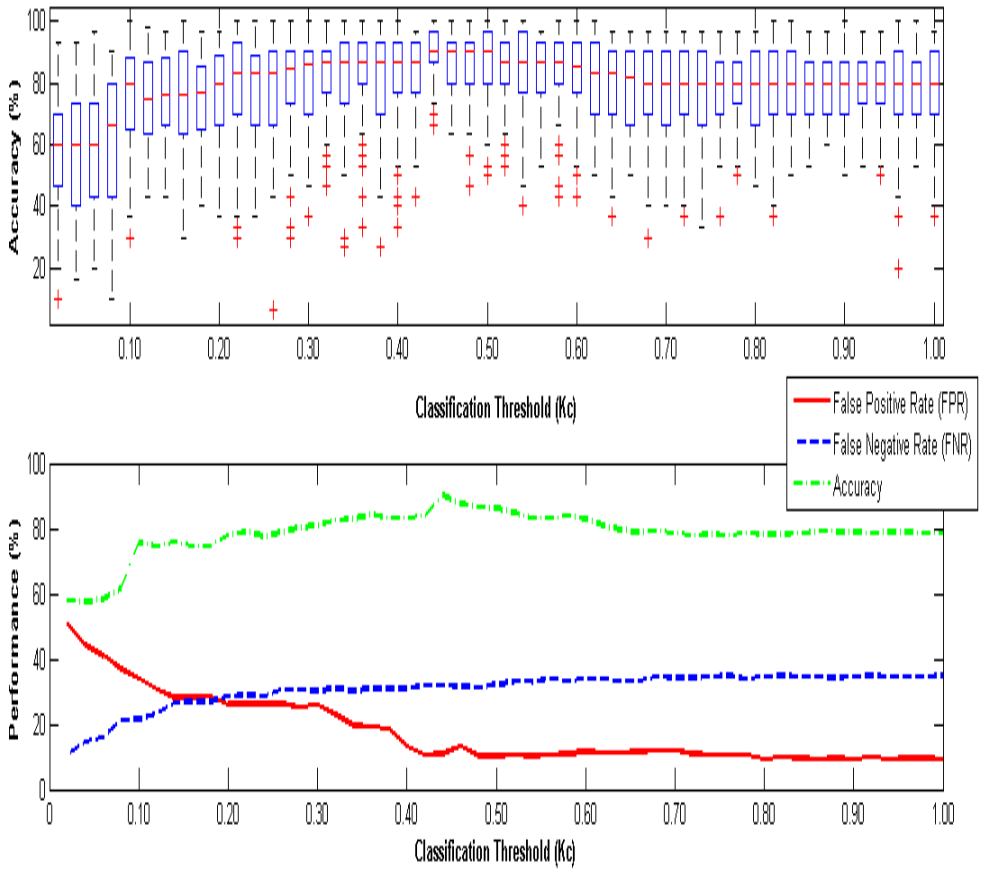


Figure 5.10: Influence of Classification Threshold (Kc) Parameter for the Case 1 Scenario

of Ksm was not significant in general. The predictive accuracy, the FNR and the FPR appear stable for all values of Ksm tested.

The Influence of the Initial Number of Memory Cell (Kt) Parameter Tested on the Case 1 Scenario

The Kt parameter determines how many memory cells will be selected from the initialisation data set in the initialisation phase. Therefore, a high Kt may guarantee that more B cells are produced during the initialisation, compared with a low Kt value. Figure 5.15 shows that, in general, the greater the Kt value the greater the accuracy. However, as the Kt increased, the predictive accuracy started to gradually stabilise. The FNR and FPR varied as the Kt value changed. The false positive rate started low but gradually increased, suggesting that more memory cells were produced. However, the false negative rate was also affected by this situation. Given a value of $Kt = 1$, a large number of emails were misclassified

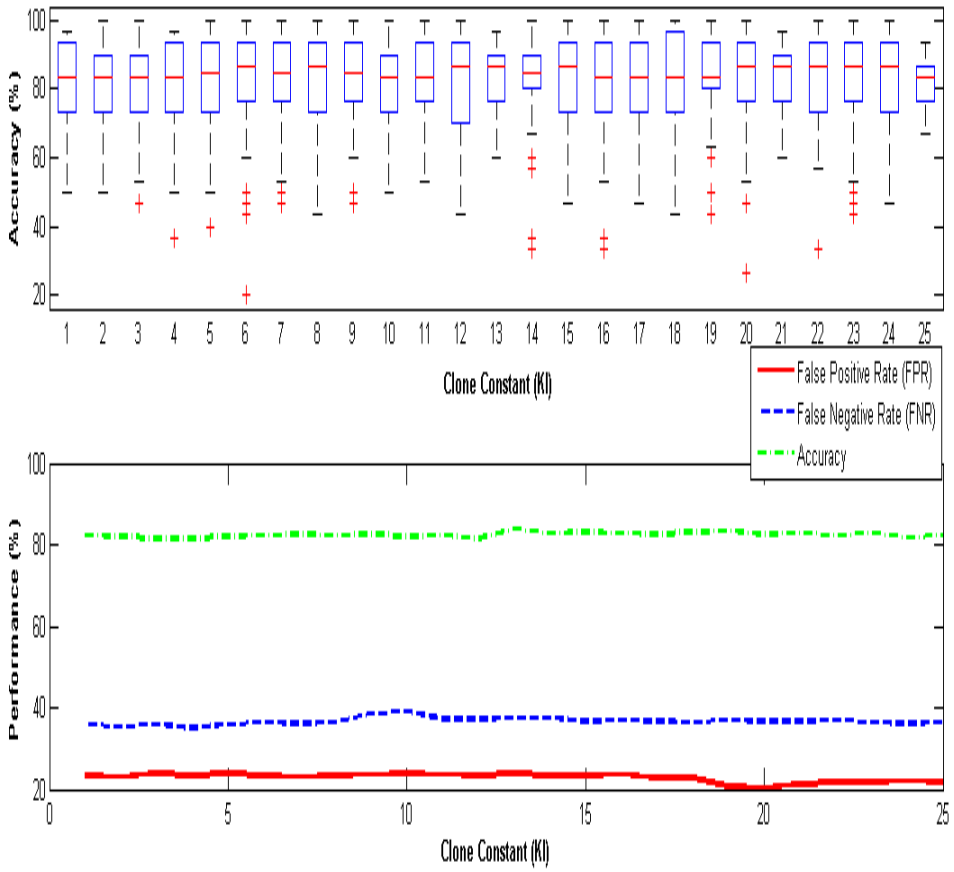


Figure 5.11: Influence of Clone Constant (Kl) Parameter for the Case 1 Scenario

as negative. This situation shows that the FNR value was biased by the extreme situation and the single memory cell did not recognize any antigen during the running phase, thus, shows the TPR rate of 10%. Overall, the results suggests that the influence of the Kt parameter was significant compared with Kl, Km, Ksm and Ksb.

5.3.2 Assessment of the Optimised Parameter for the Case 1 Scenario

After the tests described above, our next task was to investigate the effect on the predictive performance of the difference between the optimised parameter value and the default parameter value using Vargha-Delaney A statistics. In Table 5.10, parameter Set A is a value which was described in Table 5.9 and thus formed the basis of the investigation. Parameter Set B is a optimised parameter value.

Figure 5.16 shows that the A statistic value for the parameter Kc, Ka, Km

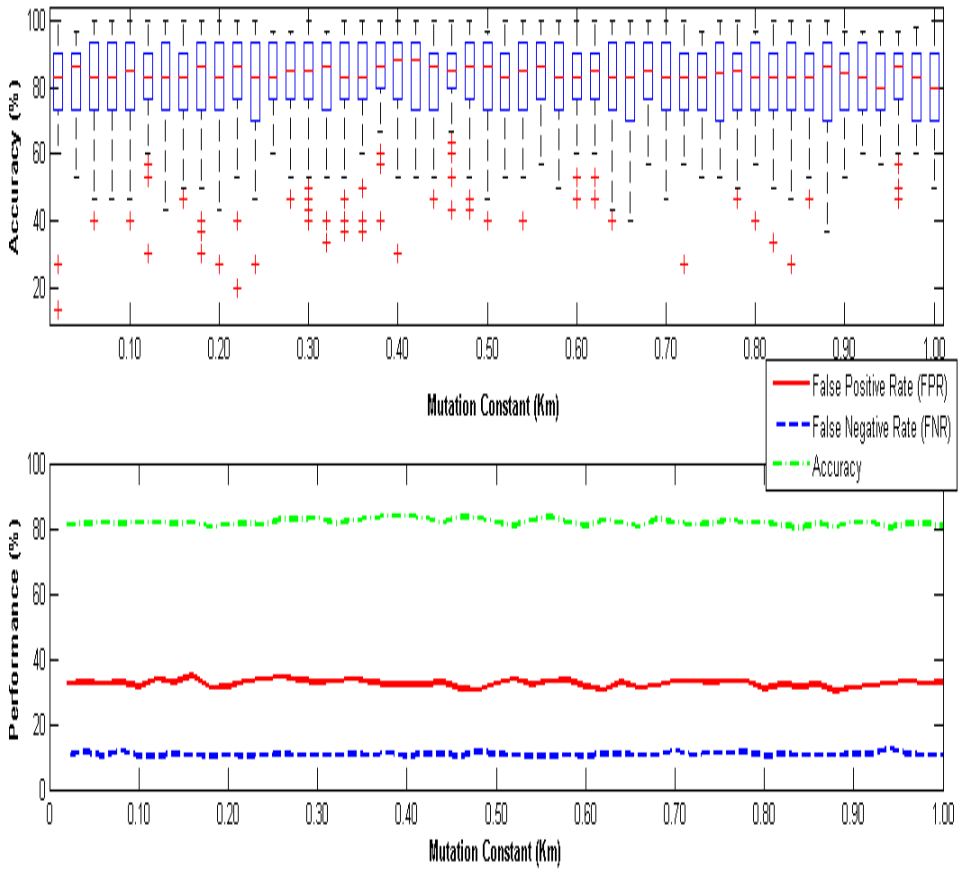


Figure 5.12: Influence of Mutation Constant (Km) Parameter for the Case 1 Scenario

and Ksb had a large effect (with the A value above 0.714)[3] as the parameter value increased. However, as the parameter value approached the optimised value, the effect of the difference between the samples are smaller. For parameters Kl, Kt and Ksm the effect between samples was small, with the A value less than 0.6. Detailed results of the non-parametric statistical analysis, which included the p-value, Mann-Whitney-Wilcoxon or rank-sum test, and the Vargha-Delaney A statistics value, can be seen in Appendices D. Table 5.11 summarises the effect of the extended AISEC parameters on the performance of multiple-topic classification for the Case 1 scenario. To summarise, the influence of Kl, Kt and Ksm was not significant, and the medians between these samples were the same. This indicates that there were no noticeable correlations between these parameters and the metrics measured.

[3]for a complete list of A value and its description, see Table 4.3 on Section 4.3.1

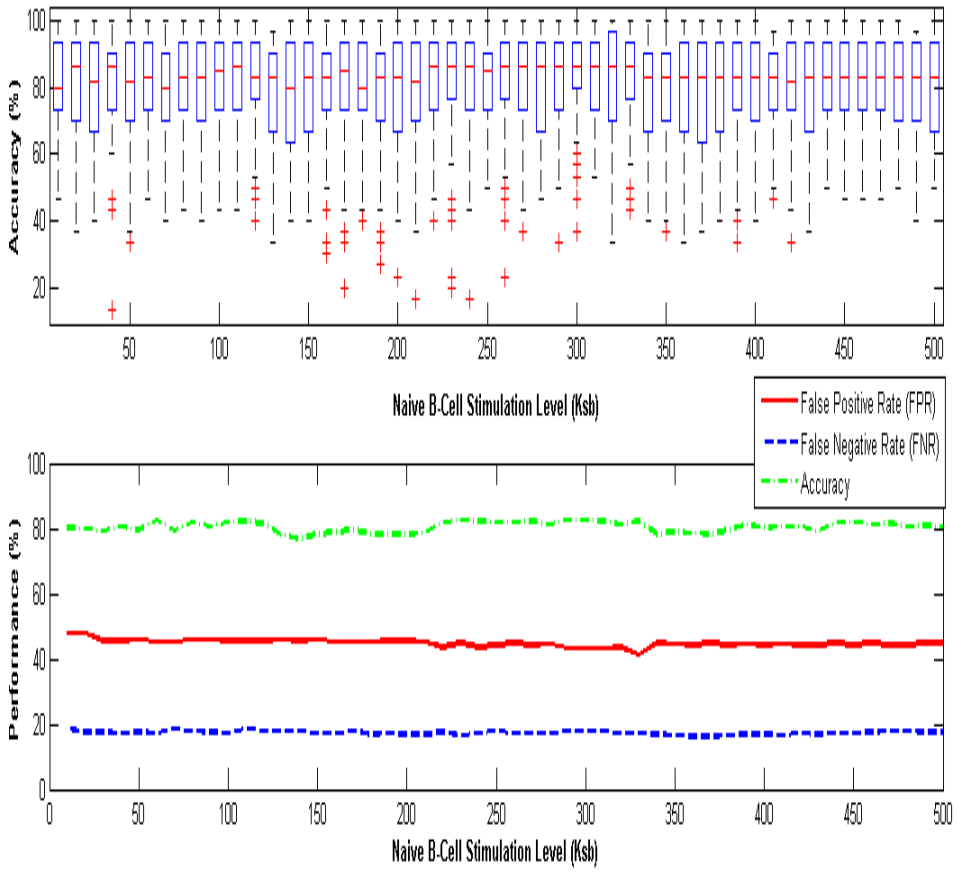


Figure 5.13: Influence of Naive B Cells Stimulation Level (Ksb) Parameter for the Case 1 Scenario

5.3.3 Case 2: Both Topics have Low Numbers of emails

Influence of the Classification Threshold (Kc) and Affinity Threshold (Ka) Parameters Tested on Case 2 Scenario

For the Case 2 scenario, the influence of the overall parameters studied on the classification performance was small compared with the case 1 scenario. However, what can be seen in Figure 5.17 is that an increase in the classification threshold (Kc) created a situation in which a classified email was slightly more likely to be classified as positive, leading therefore to decreasing FPR, and the FPR is almost constant as the affinity threshold, Ka increases. This also had the effect of reducing the FNR. However, the difference was insignificant.

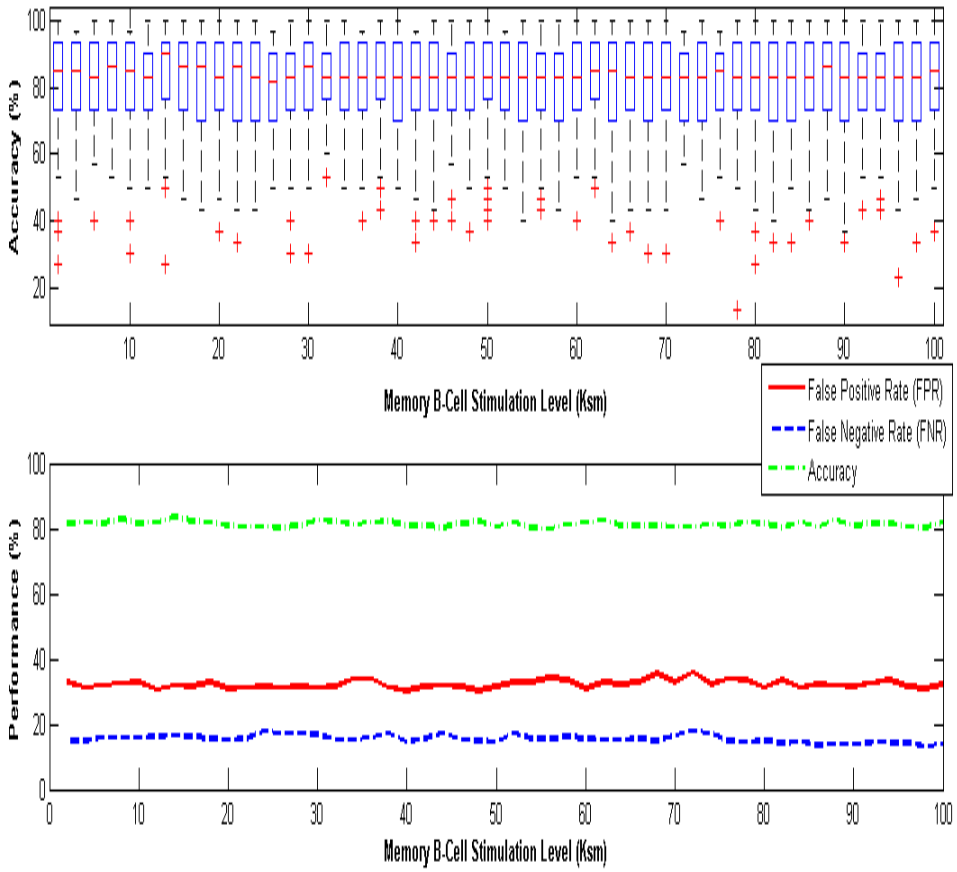


Figure 5.14: Influence of Memory B Cells Stimulation Level (Ksm) Parameter for the Case 1 Scenario

Influence of the Clone Constant (Kl), Mutation Constant (Km) and Number Memory Cell (Kt) Parameter for the Case 2 Scenario

Figure 5.18 shows that the influences of Km and Kl over the predictive performance measures was not so significant. However, the influence of initial number of memory cells associated with the (Kt) appears that the greater the Kt value the greater the accuracy, however, the predictive accuracy gradually stabilised as the parameter value increased. The false positive and false negative rates for the Kt parameter varied as the Kt value changed. The false positive rate for Kt started low but gradually increased, suggesting that too many memory cells are produced. However, the false negative rate was affected by this situation in the opposite direction.

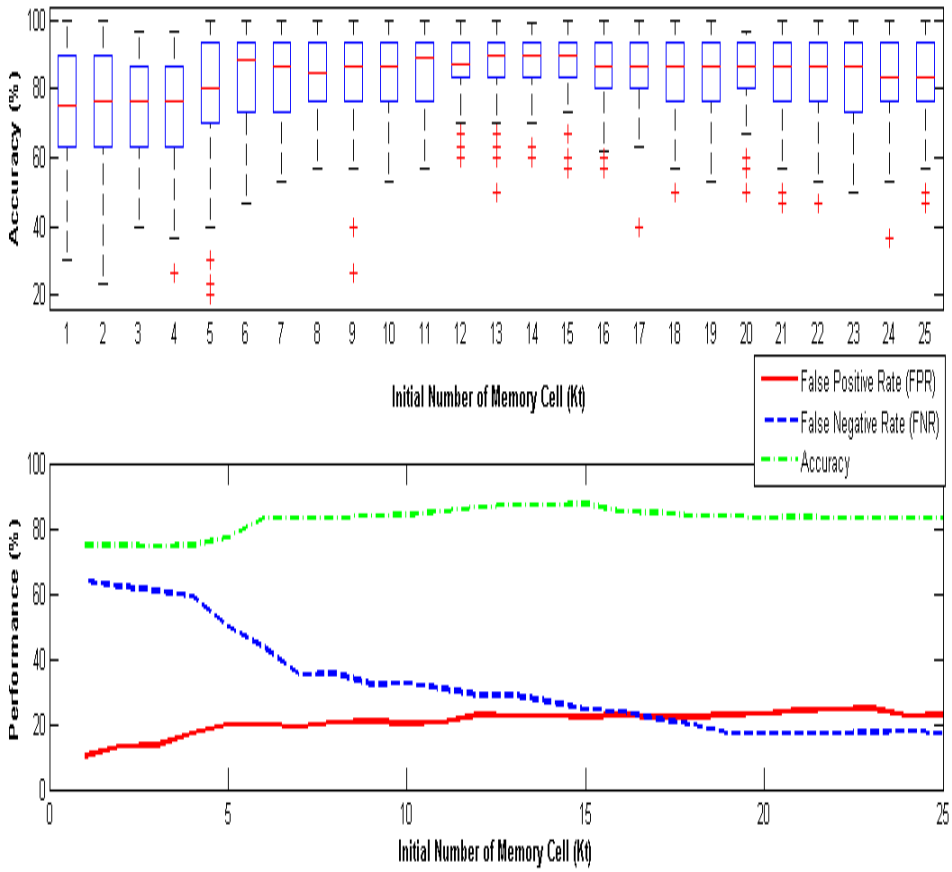


Figure 5.15: Influence of Initial Number of Memory Cell (Kt) Parameter for the Case 1 Scenario

Influence of the Naive B Cells Stimulation Level (Ksb), Memory B Cells Stimulation Level (Ksm) Parameter for the Case 2 Scenario

The situation was similar to that of parameters Ksb and Ksm and Figure 5.19 shows that the influence of Ksb and Ksm was not significant in general. The accuracy, FPR and FNR appear stable for all values tested.

5.3.4 Assessment of the Optimised Parameter for the Case 2 Scenario

Although it was found that for the Case 2 scenario most of the parameters did not show a significant result, we were still interested in evaluating the effect between samples. In this test, the optimised parameter values taken from the experiment described above (the Case 2 scenario) were tested with the parameter values based on the Case 1 scenario. Table 5.12 presents the set of parameter val-

Parameter	Set A	Optimised Value for Case 1
Kc	0.50	0.40
Ka	0.42	0.44
Kl	3.0	9.0
Km	0.22	0.38
Ksb	140	220
Ksm	50	30
Kt	20	13

Table 5.10: Set of Baseline and the Optimised Value for Case 1 Scenario

Parameter	Predictive Accuracy	FPR	FNR
Kc	Yes	Yes	Yes
Ka	Yes	Yes	Yes
Kl	Little	Little	Little
Km	Little	Little	Little
Ksb	Moderate	Little	Little
Ksm	Little	Little	Little
Kt	Yes	Moderate	Large

Table 5.11: Summary of the Influence of the Optimised Extended AISEC Parameters in Multiple-Topic Classification: Case 1 Scenario

ues where parameter Set A is a value based on the Case 1 scenario, while parameter Set B is the optimised parameter value taken from the experiment described above.

Overall, Figure 5.20 shows that the A value for the parameter Kc, Ka, Km and Ksb had a large effect (with the A value above 0.714)[4] as the parameter value increased. However, the effect between samples was smaller when the parameter value approached the optimised value. For parameters Kl, Kt and Ksm the effect between samples was small with the A value more than 0.6. Detailed results of the non-parametric statistical analysis, including the p-value, Mann-Whitney-Wilcoxon or rank-sum test, and the Vargha-Delaney A statistics value, can be seen in Appendices E. Table 5.13 summarises the effect of the extended AISEC parameters on the performance of multiple-topic classification for the Case 2 scenario. To summarise, the influence of Kl, Kt and Ksm was not significant and the medians between these samples were the same. Overall for the Case 2 scenario, the influence of the algorithm's parameters was low and not significant compared with the Case 1 scenario. This may due to the low number of emails to

[4]for a complete list of A value and its description, see Table 4.3 on Section 4.3.1

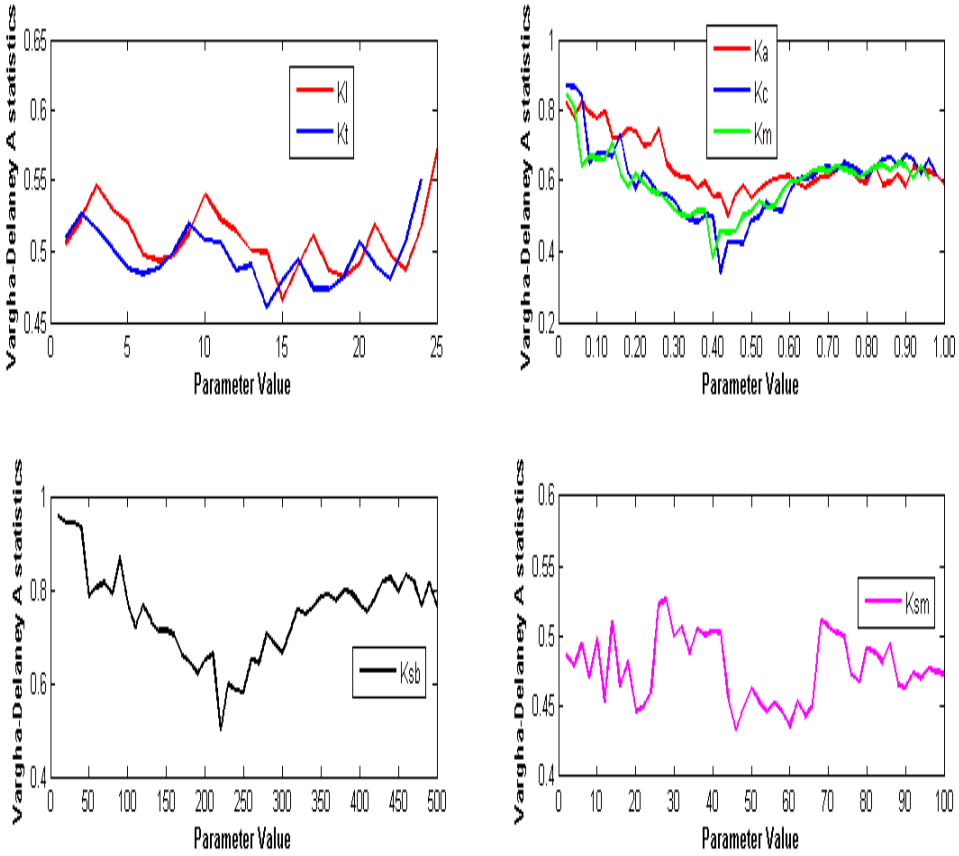


Figure 5.16: Vargha-Delaney A Statistics in Multiple-Topic Classification: Case 1 Scenario

be classified which may have affected the algorithm's overall performance.

5.3.5 Case 3: Both Topics had High Numbers of emails

In the Case 3 scenario, the influence of the overall parameters studied over the classification performance was higher compared with the classification performance in the Case 2 scenario. However, a similar classification performance occurred when compared with the Case 1 scenario. A particular reason for this may be that there were sufficient emails to classify, which enabled the algorithm to perform better. Figure 5.21, Figure 5.22 and Figure 5.23 show the influence of the algorithm's parameters in this experiment.

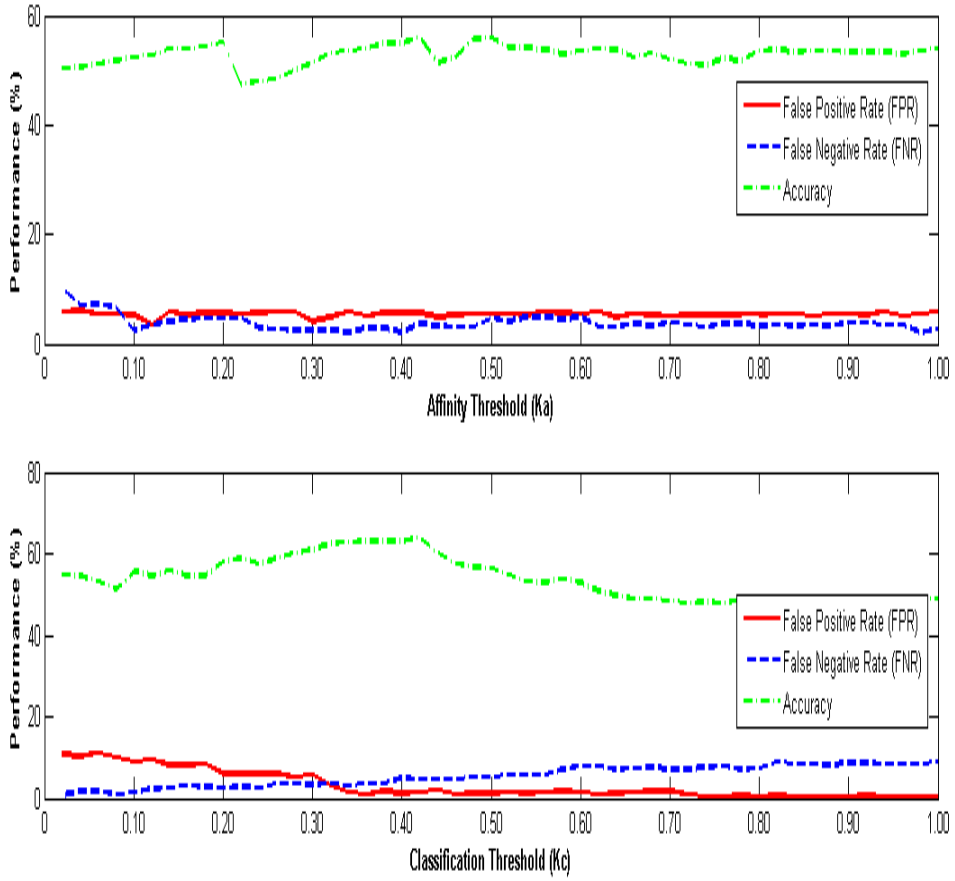


Figure 5.17: Influence in Affinity Threshold (Ka) and Classification Threshold (Kc) Parameter for the Case 2 Scenario

The Influence of the Affinity Threshold (Ka) and Classification Threshold (Kc) Parameters Tested on the Case 3 Scenario

As shown in Figure 5.21, an increase in the affinity threshold (Ka) and the classification threshold (Kc) created a situation in which a classified email was slightly more likely to be classified as positive, led to a decreasing FPR. This also has the effect of reducing the FNR. At a low Kc value such as 0.02, FPR is above 80%, which is comparatively high compared with the FNR value of less than 11%. As Kc increased, the situation became inversed; FPR decreased towards less than 10% while FNR increased towards more than 34%. The distribution of accuracy forms a peak curve at low values of Kc, due to the fast increase in FNR and quickly falling FPR, but as the value of Kc increased, the accuracy levels became stable at 80%. To summarise, at a high Kc it can be observed that the Kc value is too high for any email to be classified as positive and therefore in every run every

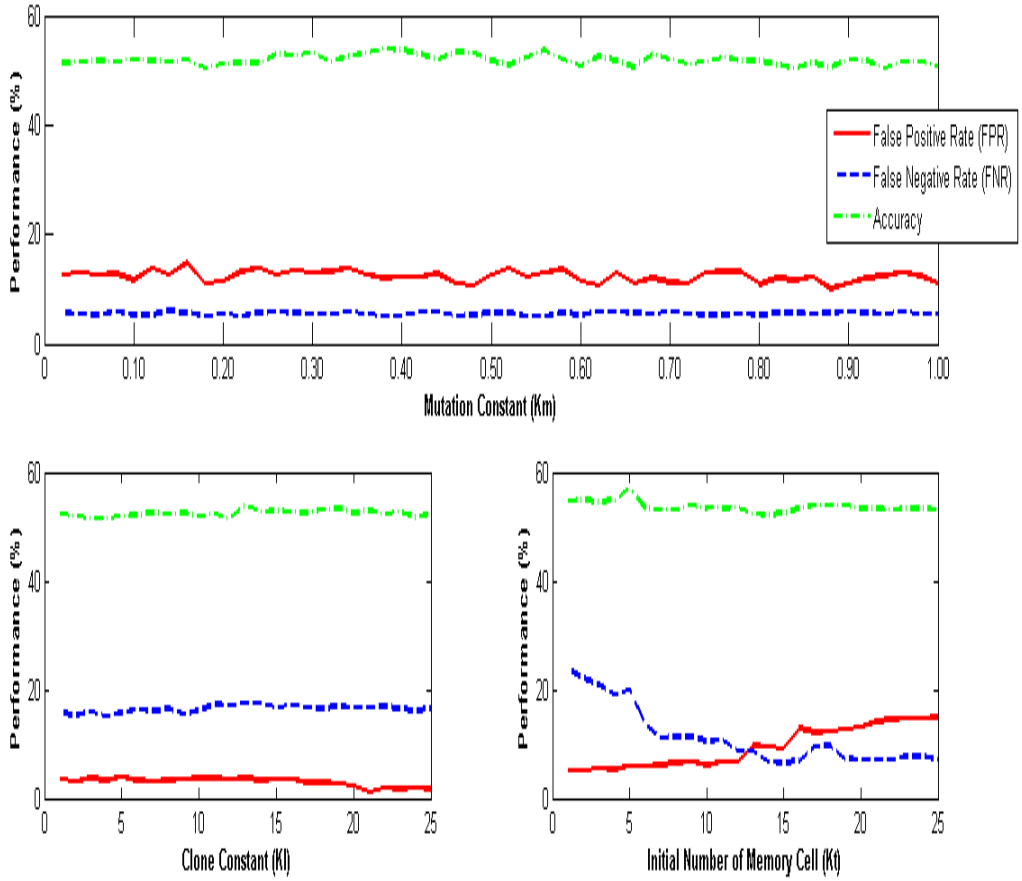


Figure 5.18: Influence in Clone Constant (Kl), Mutation Constant (Km) and Initial Number of Memory Cell (Kt) Parameter for the Case 2 Scenario

email was classified as negative, resulting in an accuracy of 60%.

For the Ka parameter, Figure 5.21 shows that FNR decreased from above 70% to 9% as the Ka value increased. Inversely, FPR increased from 10% to above 34% as the Ka value increased. Unlike Kc, for Ka values higher than around 0.38, all observations of FPR appear fairly stable. Like Kc, the Ka parameter also had a large effect on the overall accuracy of the algorithm, with this value ranging again from approximately 50% to just under 80%. It is believed that a Ka at a certain level, around 0.38 here, already disables the factors that may affect performance, such as refining cells by removing them in a false positive classification, therefore values above this level make little difference. In other words, there were no more B cells close enough to the antigens representing the interesting emails with affinities between them greater than the Ka value.

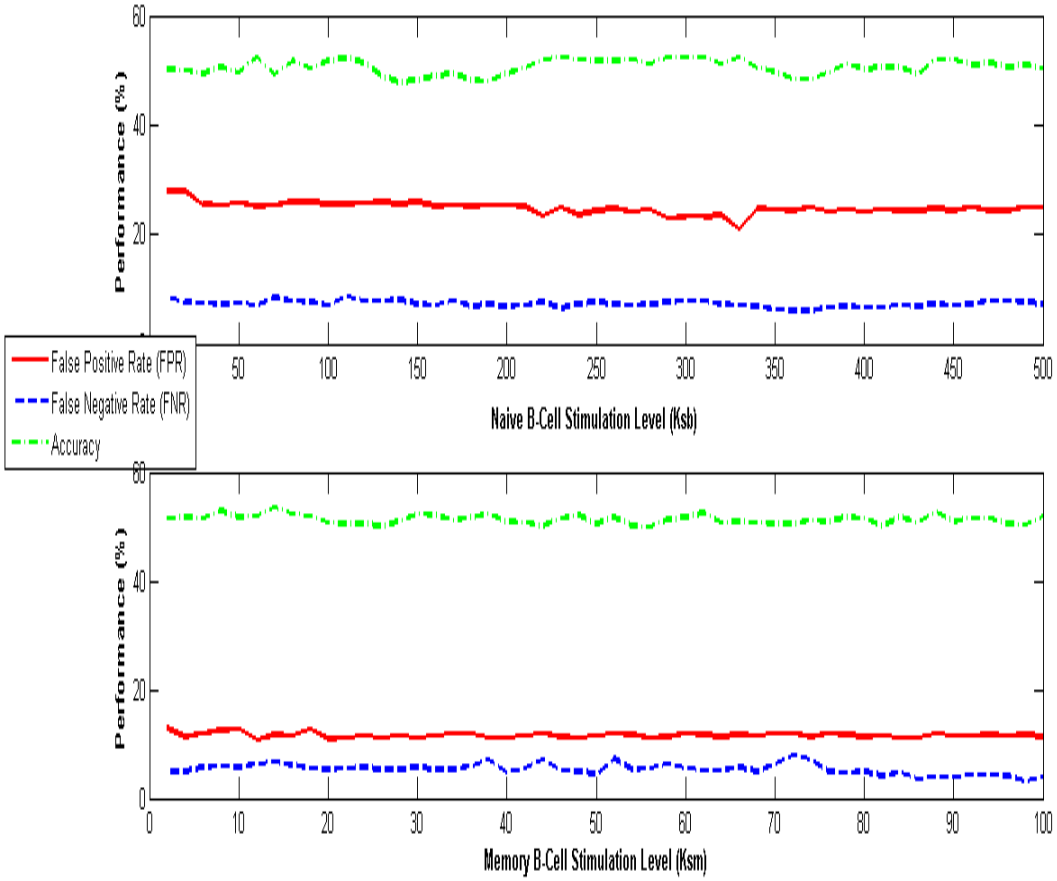


Figure 5.19: Influence in Naive B Cells Stimulation Level (Ksb) and the Memory B Cells Stimulation Level (Ksm) Parameter for the Case 2 Scenario

The Influence of the Clone Constant (Kl), Mutation Constant (Km) and Initial Number of Memory Cell (Kt) Parameter Tested on the Case 3 Scenario

Figure 5.22 suggests that the influences of Km and Kl over the predictive performance measures was very low compared with the influence of Ka and Kc. What can be seen in the figure is that an increasing clone constant creates a situation in which an email is slightly more likely to be classified as positive, leading therefore to negligible FPR. This also has the effect of reducing the FNR. The mutation constant parameter, with a high rate of mutation, contributes to a negligible false positive classification rate, however, the difference is insignificant.

In terms of the Kt parameter, during the testing of this parameter, as Kt was increased, the number of initialisation antigens available decreased. An interesting observation from the tests is that when Kt was high, the resultant B cells set was sparse. This is because, given there are large numbers of memory cells, then

Parameter	Optimised Value for Case 1	Optimised Value for Case 2
Kc	0.40	0.46
Ka	0.44	0.44
Kl	9.0	12.0
Km	0.38	0.38
Ksb	220	230
Ksm	30	15
Kt	13	20

Table 5.12: Set of Optimised Value for Case 1 and the Optimised Value for Case 2 Scenario

Parameter	Predictive Accuracy	FPR	FNR
Kc	Yes	Yes	Yes
Ka	Yes	Negligible	Yes
Kl	Little	Negligible	Negligible
Km	Yes	Negligible	Little
Ksb	Moderate	Yes	Little
Ksm	Moderate	Little	Little
Kt	Yes	Moderate	Large

Table 5.13: Summary of the Influence of the Optimised Extended AISEC Parameters in Multiple-Topic Classification in Case 2 Scenario

there are a small number of cells left in the initialisation set. Figure 5.22 suggests that the influence of the Kt parameter was much more significant compared with the influence of Kl and Km parameters. It appears that the greater the Kt value the greater the accuracy, however the performance gradually stabilised as the parameter value increased. The FPR started low but gradually increased, suggesting that too many memory cells were produced. This will create a situation in which an email is more likely to be classified as positive. However, the FNR is affected by this situation in the opposite direction. It seems that the FNR value was biased by the extreme situation when $Kt = 1$, and maybe some emails were correctly classified as positive.

The Influence of the Naive B Cells Stimulation Level (Ksb) and Memory B Cells Stimulation Level (Kt) Parameters Tested on the Case 3 Scenario

Figure 5.23 shows that the influence of the Ksb and Ksm parameters was not significant in general. The accuracy, FPR and FNR appear stable for all values of

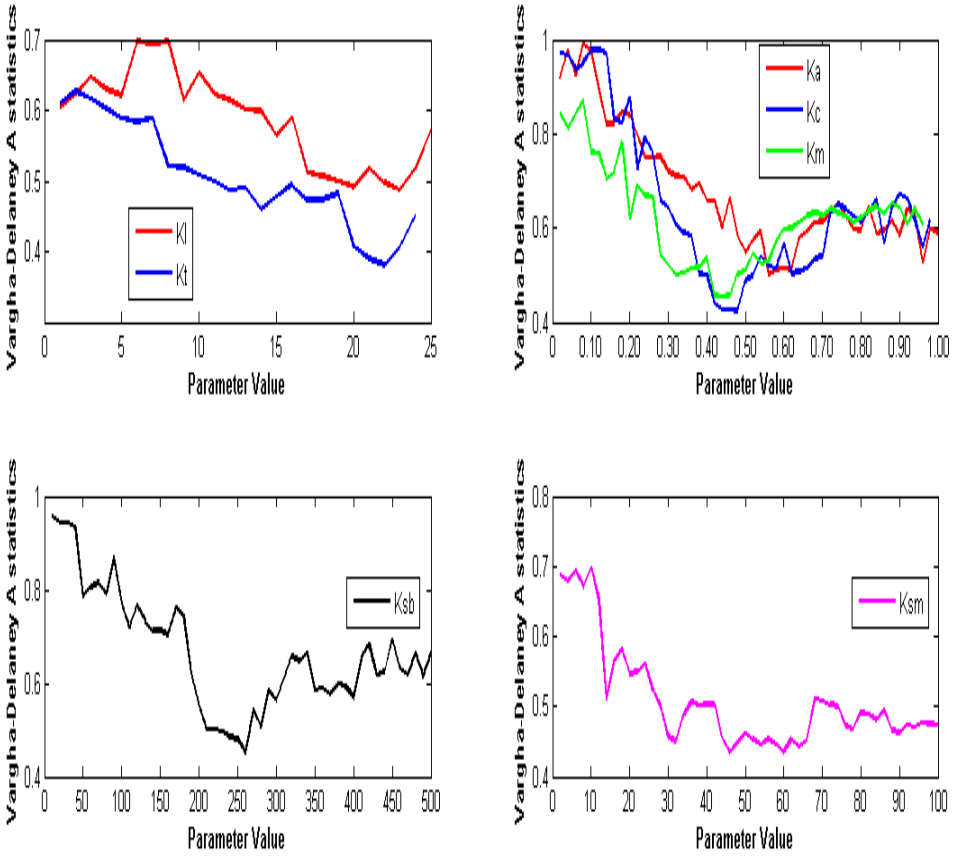


Figure 5.20: Vargha-Delaney A statistics in Multiple-Topic Classification: Case 2 Scenario

Ksb and Ksm tested.

5.3.6 Assessment of the Optimised Parameters for the Case 3 Scenario

After the tests described above, the next step was to investigate the effect on the samples between the optimised parameter values obtained by the above experiment with the set of parameter values from the Case 1 scenario. Table 5.14 presents the set of parameters used in the statistical test. In Table 5.14, parameter set A is the values obtained in a Case 1 scenario while parameter set B is the optimised parameter values from the experiment described above.

Figure 5.24 shows that the A statistic value between samples from Case 1 and Case 3 for parameters Kc, Ka, Km, Ksb and Kt had a medium effect (with the A

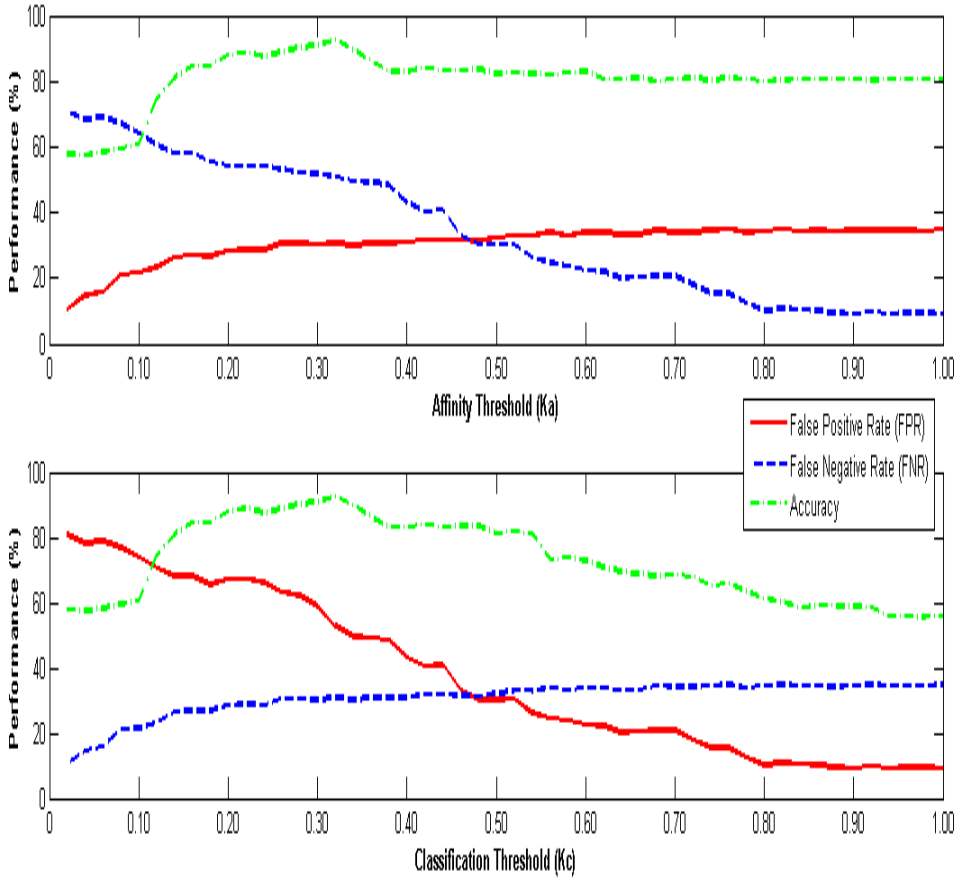


Figure 5.21: Influence in Affinity Threshold (Ka) and Classification Threshold (Kc) Parameter for the Case 3 Scenario

statistic value above 0.64)[5]. However, parameters Kl and Ksm between samples from Case 1 and Case 3 had a small effect, with the A value equal to or less than 0.56. Detailed results of the non-parametric statistical analysis, including the p-value, Mann-Whitney-Wilcoxon or rank-sum test, and the Vargha-Delaney A statistics, can be seen in Appendices F. Table 5.15 gives a summary of the effect of the extended AISEC parameters on the performance of multiple-topic classification in the Case 3 scenario.

5.4 Summary

This chapter has presented and discussed the results of a sensitivity analysis carried out on the parameters of the extended AISEC algorithm and has described

[5]for a complete list of A value and its description, see Table 4.3 on Section 4.3.1

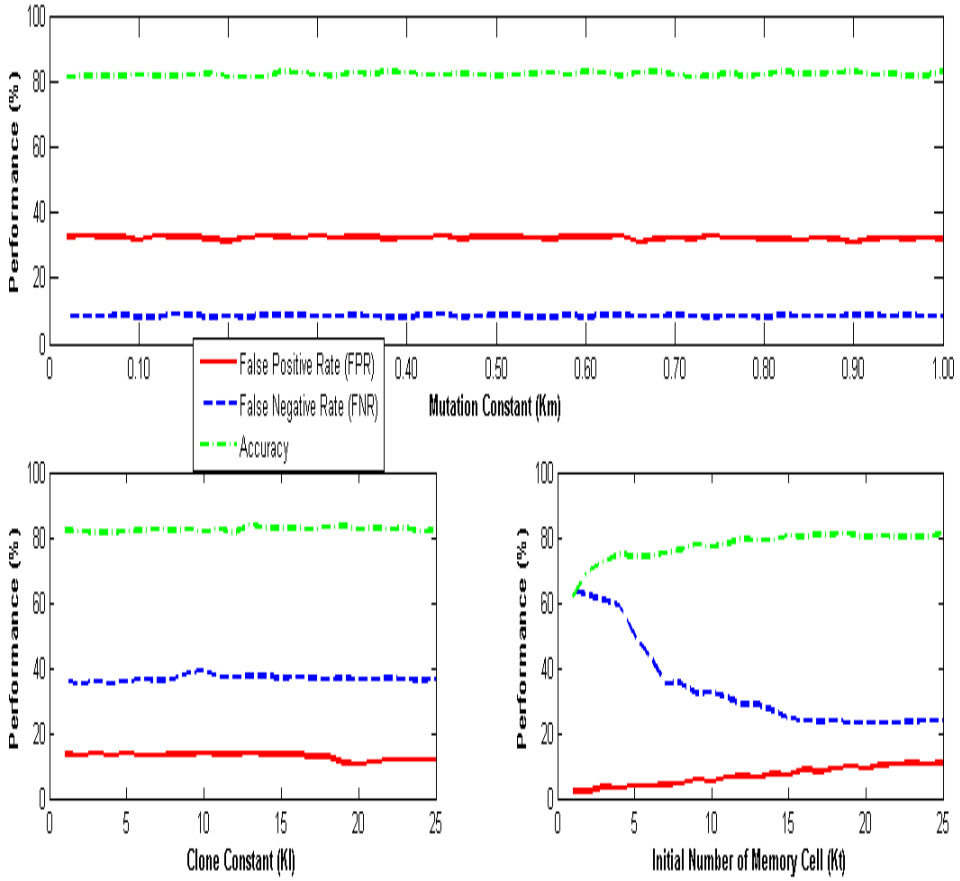


Figure 5.22: Influence in Clone Constant (Kl), Mutation Constant (Km) and Initial Number of Memory Cell (Kt) Parameter for the Case 3 Scenario

an investigation by a non-parametric statistical analysis to evaluate the optimised parameters and their influences on classification performance. The results from the sensitivity analysis give insights into how these parameters can be optimised to provide a better performance for each of the performance metrics. In the first part of this chapter, a hypothesis for the influence of each of the parameters of the revised algorithm was put forward. A chart displayed the behaviour of the parameters and a conclusion was reached based on the chart either to confirm or to refute the hypothesis. In single-topic classification, the most influential parameters were the classification threshold (Kc) and the affinity threshold (Ka). In the evaluation of the optimised parameter values (Set B) with the default parameter values (Set A), the predictive accuracy and the MCC score over parameter set B had increased compared with set A. Thus, the extended AISEC based on set B had higher predictive accuracy in the running set. The increase in accuracy from parameter set A to parameter set B was further tested for statistical signifi-

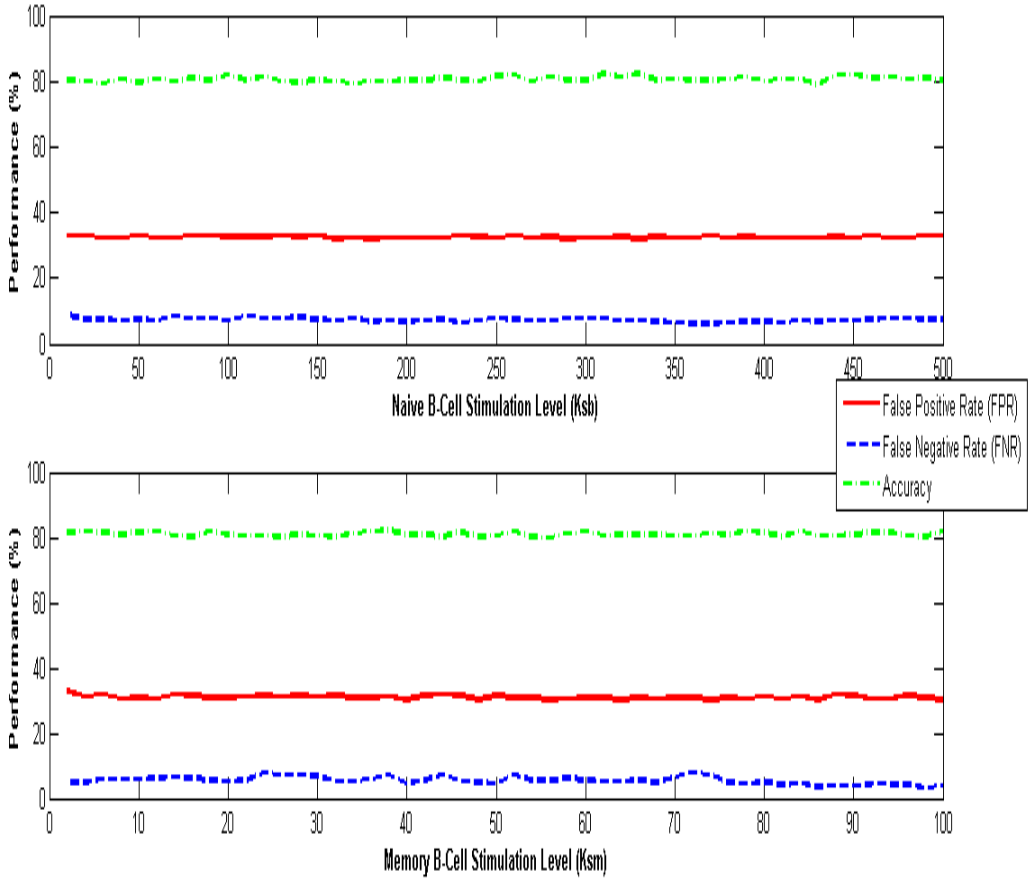


Figure 5.23: Influence in Naive B Cells Stimulation Level (Ksb) and the Memory B Cells Stimulation Level (Ksm) Parameter for the Case 3 Scenario

cance. The analysis showed that the classification threshold (Kc) and the affinity threshold (Ka) parameters were statistically and scientifically significant with a large effect on the difference of predictive performance between the samples. For parameters Kl, Km, Ksb and Ksm, the effect sizes between samples were small and the influence of these parameters was low. In the multiple-topic classification, the sensitivity analysis was carried out based on email topic. These email topics were investigated in three types of scenario, termed Case 1, Case 2 and Case 3. These scenarios were based on a situation with extreme cases (email topics having low numbers of emails) and mild cases. The purpose of the test was to identify the level at which the algorithm became reliable and acceptable in the extreme case and mild case scenarios. The results of the investigation show that in the extreme cases with low number of emails, the test did not produce satisfactory result. This may have been due to the insufficient number of emails to be classified, which might have affected the algorithm's overall performance. An

Parameter	Optimised Value for Case 1	Optimised Value for Case 3
Kc	0.40	0.32
Ka	0.44	0.38
Kl	8.0	9.0
Km	0.38	0.38
Ksb	220	180
Ksm	30	40
Kt	13	10

Table 5.14: Set of Baseline and the Optimised Value for Case 3 Scenario

Parameter	Predictive Accuracy	FPR	FNR
Kc	Yes	Large	Moderate
Ka	Yes	Moderate	Large
Kl	Little	Little	Little
Km	Little	Little	Little
Ksb	Little	Little	Little
Ksm	Little	Little	Little
Kt	Moderate	Yes	Large

Table 5.15: Summary on the Influence of the Optimised Extended AISEC Parameters in Multiple-Topic Classification: Case 3 Scenario

additional memory detector is needed to extend the system's capability to perform the classification task. However, in the mild cases with email topics having high numbers and low number of emails (Case 1) and email topics having high number of emails (Case 3), the experiment showed that the most influential parameters were the classification threshold (Kc), the affinity threshold (Ka) and the initial number of memory cells (Kt). While parameter Kl, Km, Ksb and Ksm had very little effect in general. In a non-parametric statistical analysis for the Case 1 scenario, the optimised parameter value sample was tested against the default parameter value sample and showed that the Kc, Ka and Kt parameters had a large effect based on sample size. However, for the Kl, Km, Ksb and Ksm parameters, the effect size was small. In a statistical significance test for the Case 2 scenario, the analysis was based on the sample size between the optimised values for Case 2 and the sample size from the optimised parameter values in the Case 1 scenario. The analysis revealed that the Kc, Ka, Km and Ksb parameters had a large effect while the effect of the Kl, Kt and Ksm parameters between the samples was small. The results were, however, different when the sample size

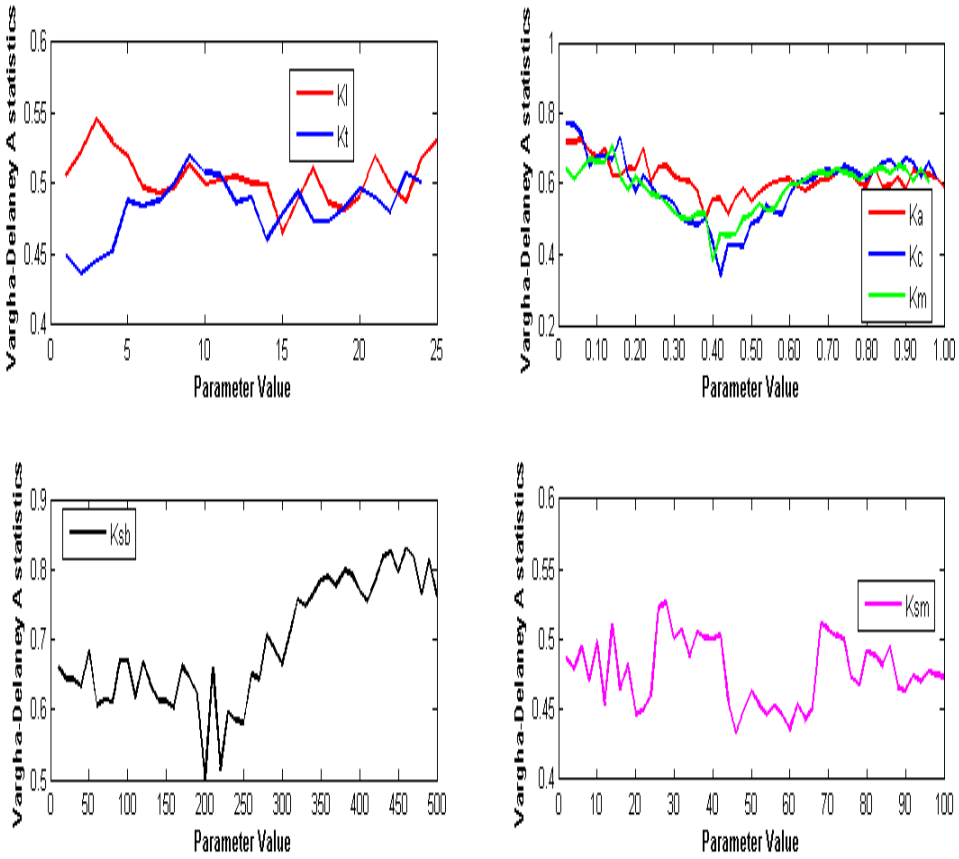


Figure 5.24: Vargha-Delaney A statistics in Multiple-Topic Classification: Case 3 Scenario

of the optimised parameter values from the Case 3 scenario were compared with the optimised parameter values from the Case 1 scenario. The analysis showed that the Kc, Ka, Km, Ksb and Kt parameters had a medium effect while the Kl and Ksm parameters had a small effect. To summarise, the influence of the algorithm's parameters in multiple-topic classification became acceptable at the level of three or four topics of interest with sufficient numbers of emails.

The extended AISEC algorithm was inspired by clonal selection and the dynamics inherent in this algorithm have been proven in the analysis of the algorithm with the comparative approach presented in Chapter 4, followed by the sensitivity analysis on its parameters discussed in this chapter. The next chapter continues to investigate further an artificial immune system for profile adaptation and widens the experimentation into web documents. Inspired by dynamic clonal selection, the dynamics inherent in AIS algorithms are believed to be powerful enough to make AIS a successful solution to the problem of profile adapta-

tion. It has been shown in the previous chapter that an immune-inspired algorithm written with information filtering as its primary goal might yield a classification accuracy comparable with a baseline approach in this continuous learning scenario.

CHAPTER

SIX

EXPERIMENT ON PROFILE ADAPTATION THROUGH DYNAMIC CLONAL SELECTION

In the previous chapters (Chapter 4 and Chapter 5), the algorithm AISEC was described, tested and evaluated. The results of the tests showed that a clonal selection based AIS algorithm could perform classification of emails with an accuracy and MCC score better to that of a comparative algorithm (the simple recurrent neural network and Naive Bayesian). In addition, AISEC was shown to work well at a continuous classification task. The previous chapter has also demonstrated the ability of the clonal selection algorithm, particularly the extended AISEC, to perform multiple-topic classification based on email topic. The experiment is setup where each user profile has to represent more than one topic in parallel and adapt to both modest and radical variations in them. The objective of the experiment is not only to test the algorithm in representing multiple topics in parallel but also to demonstrate the ability of the algorithm to forget the previous topic as it starts to process the new topic. The experiment has shown that the antibody-antigen interaction of B cells and the introduction of diversity of words based on gene library have a positive effect both on the profiles's adaptability to interest changes and on the profile's response to the email corpus. Later, in Chapter 5 it was described the potential usefulness of sensitivity analysis technique in interest classification based on email. The analysis is carried on the extended AISEC's parameters. The results from the sensitivity analysis have gives insights on how these parameters can be optimised to provide a better performance for

each of the performance metrics.

Based on these positive steps, this chapter is concerned with a study into experimenting with AIS towards profile adaptation to changes on user interests in the content based document filtering. As we have already argued in Section 2.3, user interests are by nature dynamic. A combination of parameters causes a variety of changes. Frequent changes in the user's short-term needs contribute to progressive changes in the user's long term interests and vice versa. The user's interest may shift frequently between different topics or related subtopics. New topics and subtopics of interest emerge and the interest in a certain topic might be lost. A subtopic may attract increased interest to become a general topic of interest. For example, a general interest in *Biologically Inspired Algorithm* can trigger an interest in *Artificial Immune Systems*, which may evolve to include related aspects like *Immune Network Algorithm* and *Clonal Selection Algorithm*. The latter may themselves develop, causing a decay in the initial interest in *Biologically Inspired Algorithm* and the emergence of other topics like, *Complex Adaptive Systems*, *Homeostasis* and so on.

Despite the complex and dynamic nature of user interest, there is an evident tendency in the literature to couple single-topic profile representations. We have highlighted some of the existing approaches which include linear learning algorithm, genetic algorithms (GAs) or Memetic Algorithms (MAs) and connectionist approaches. These approaches are already discussed in Section 2.4. Although these approaches have influenced our work, most of the approaches do not tackle multiple topics of interest, except work from Nanas et. al [44]. Their work was influenced by the *Autopoetic Theory* [144] which allows the profile to self-organise in response to changes in the user interests. Later, in [21] they used an immune-network metaphor to build a network of terms that represents a user's multiple interests and that adapts to changes in them through a process of self-organisation. In this work, we suggest that profile adaptation can be developed by incorporating ideas from aspects of dynamic clonal selection (DCS) with the gene libraries to maintain sufficient diversity through synset relationship based on WordNet. DCS has been identified as an AIS algorithm that supports learning in dynamically changing environment [91, 115–118]. In our proposed DCS it is used to maintain the profiles with a gene library maintaining a sufficient diversity for the set of terms that can be added to the profile during mutation. The goal is to adapt our multiple-topic profile both to short-term variations in the user's need and to progressive, but potentially radical changes in long-term interests. In the next section, we set the theoretical foundation of DCS which will be described in detail in Section 6.1. The algorithm and the process of DCS known as profile

adaptation through dynamic clonal selection (ProAdDCS) will be explained in Section 6.2, Section 6.3 and Section 6.4. It is then evaluated using virtual users in Section 6.5. Comparative performance is important in order to assess whether the works are incremental improvements on the state of the art or evolutions of existing work. The results indicate the profile's ability to respond to a variety of changes in a stream of feedback documents. The profile appears to be able to adapt to a variety of simulated changes in a virtual user's interest.

6.1 Dynamic Clonal Selection (DCS)

In Section 3.5 we have introduced dynamic clonal selection (DCS) as an inspiration for adapting a user profile. Here, we discuss further some of the potential of DCS towards adapting a user profile.

DCS is another variation of clonal selection algorithm (CSA) which has been introduced by Kim and Bentley [120] known as dynamic clonal selection algorithm (DynamiCS). In particular, *DynamiCS* is based on Hofmeyr's [66] idea of 'dynamics' of three different populations: immature, mature and memory detector populations. Initial immature detectors are generated with random genotypes. Using the negative selection algorithm, new immature detectors are added to keep the total number of detectors constant after a predefined number of generations (polarization period, T). If a detector is within its predefined life span L, and the match counts are larger than a predefined activation threshold A, it becomes a memory detector. Mature detectors are used to identify unknown attacks. In this way, the system learns normal behavior by observing only a small set of self-antigens at any one time. Detector cells will be replaced whenever normal behaviors change. However, the system was found to be slow to react to changes, and a sharp change in self behavior resulted in a high false positive rate. This outcome was due to memory detectors not being exposed to the entire set of self-patterns during toleration, a situation also present in the natural system. Later, Kim and Bentley had introduced the extended version of DynamiCS [12, 120] which added the mechanism of removing memory detectors when they showed a poor degree of self-tolerance. This was shown to reduce the high false positive rate, but was at the expense of requiring a larger amount of co-stimulation (user intervention) to achieve this. This work was further augmented in [91] and [116] by the addition of a hyper-mutation operator to produce the effect of gene library evolution. Rather than new detectors being generated randomly, new detectors were produced by mutating deleted detectors. Thus, a 'virtual gene library' made from mutations of deleted memory detectors was

maintained. The test results showed this scheme produced immature detectors that were better suited to cover existing non-self antigens. The Algorithm 2 for DynamiCS can be referred to Section 3.5.1.

Initially, the DCS or DynamiCS was designed for use in a computer security scenario [120], where the threat to computers on a network is continuously changing. Later, DCS was applied in various application domains such as intrusion detection, function optimization (i.e., multi-modal optimization and continuous function optimization), pattern recognition (i.e., binary character and face detection), clustering and others (i.e., time series prediction, classification, fault diagnosis and etc). In this work, DCS is used to maintain user interest profiles with a gene library maintaining the set of terms that can be added to the profile during mutation. The aspect of gene library with synset relationship in WordNet is used to maintain sufficient diversity of terms. Although the studies related with CSA and dynamic clonal selection algorithm (DCSA) are increasingly popular, according to our best knowledge, there is no study so far which has discussed its application in adapting a user profile for content-based document filtering.

6.1.1 DCS Potential for User Profile Adaptation

Profile adaptation is a challenging problem with distinct characteristics and requirements. The user profile must be capable of continuous learning and forgetting. A profile that only learns and does not forget will eventually become saturated with irrelevant features. Forgetting is necessary for maintaining an up to date representation of the user's interest. The dynamic nature of profile adaptation invites the application of biologically inspired approach. Of interest is the artificial immune systems (AIS) which can inherently maintain and boost their diversity and can dynamically control the size of the immune repertoire. The clonal selection theory in the immune system has received the attention of researchers and given them inspiration to create algorithms that evolve candidate solutions by means of selection, cloning, and mutation procedures. Moreover, diversity in the population is enabled by means of the receptor editing process. Timmis in [145] stated that a large part of AIS works have been based on the clonal selection theory. Further review on clonal selection theory and its application can be directed to [8,71,102] and recently in [146].

We have mentioned in an early section of this chapter that dynamic clonal selection is another variation of clonal selection proposed by Kim and Bentley [120]. Next, we identify some of its potential in adaptation of user profile.

1. Profile adaptation is an example of Multi-modal Dynamic Optimization

(MDO), where user may be interested in multiple topics in parallel and interest changes are time dependent. Conventional evolutionary algorithms (EAs) cannot perform well in the case of MDO because they tend to converge through a single optimum. They may lose diversity as the optimum solution proliferates and spreads over the population [16]. By enabling diversity in the population by means of DCS, this drawback is attempted to be solved.

2. The majority of content-based classification in AIS inspired by the clonal selection algorithm have been applied to problems with static data sets. According to Hart and Timmis [101], it may be more applicable with classification over dynamic environment in which patterns and trends are tracked in data over time with a form of memory detector. Adaptive user profile involved with dynamic environment. A user profile should dynamically adapt to drifts in users' interest and 'learn' with the changing interests. It has to be able to define and maintain an accurate representation of the user interests over time. Thus, DCS may be able to outperform machine-learning methods that do not possess a memory mechanism for the task in dynamic environment.

3. The preservation of diversity in DCS can be achieved namely through *heterostasis* and through the introduction of *heterogenesis*. Heterostasis concerns with the preservation of diversity; heterogenesis concerns with the creation of diversity, either through somatic hypermutation or through recruitment of new cells. In adaptive user profile, heterostasis is the goal, which can be achieved by representing the user's multiple interests and deducing changes in the interest dynamically. Heterogenesis complements heterostasis by facilitating the exploration of new areas of the information space. The DCS ability to dynamically respond to changing context is a potential advantage over static matching algorithms.

6.2 An Overview of Profile Adaptation through Dynamic Clonal Selection (ProAdDCS)

Previously, we have identified some of the potential of CSA particularly the DCS for adapting a user profile. Taking the inspiration, we believed that profile adaptation can be achieved through dynamic clonal selection. In this work, adaptation in dynamic clonal selection algorithms (DCSAs) is improved by incorporating the

gene library with synset relationship extracted from WordNet as well as the hypermutation of the population of memory B cells, referred to as *detectors*. Gene libraries are a "biological mechanism for generating a combinatorial diversity in the immune system" [117]; they are shaped by evolution to create detectors that preserve the ability to respond to novel threats [117, 118]. In fact, the gene libraries are often thought of as a biological mechanism for generating combinatorial diversity of antibodies. Taking this inspiration, we believed that it could produce reasonable coverage of detectors to detect changes of profile in varying incoming documents. Some issues that arise in the design of the algorithm are as follows [147]:

1. **Dynamic Evolution of Self** In immune-inspired systems, an underlying principle is to remove antigens that are not *self*. In adaptive systems, we need to be able to re-define self over time, by the addition of new detectors and removal of ones that are no longer useful.

2. **Change in Thresholds** Memory B cells repeatedly match incoming 'antigen' and the maturation of memory B cells occurs when the predefined affinity threshold is reached. Dynamic adaptation suggests that the affinity threshold should be able to change as user interests change. To accommodate this, we propose to implement two thresholds. An immature memory B cell is continually stimulated by new word matches until a fixed *activation threshold* is reached; it then becomes a mature memory B cell. The mature memory B cell continues to be stimulated by new word matches, and the affinity threshold is used to determine whether it has been used often enough to remain in the population. Adaptation occurs because the affinity threshold changes with changing user interests – and all mature memory B cells are continually rechecked against the changing affinity threshold.

3. **Lifespan of Mature Detectors** Thresholding results in turnover of the mature detectors, which have a finite lifetime. A key issue is to ensure that some representation of memory B cells receiving low stimulation remains in the population. This makes it easier to recognise changed interest in previously-categorised topics.

4. **Treatment of user feedback** In maintaining the population of mature memory B cells, feedback from the user is interpreted as a *co-stimulation signal*. We need to ensure that a B cell can be stimulated both by word matches and by positive user feedback.

6.2.1 ProAdDCS and AISEC

In Chapter 4, we have experimented with an AIS for email classifacation based on topic of interest. The chapter has introduced the AISEC system, an AIS based on inspiration from clonal selection algorithm. In this section, we describe the differences between ProAdDCS and the AISEC system. The first part is explained in terms of the biological analogy and the second part explains the differences in terms of functionality.

In the AISEC system a single e-mail was processed into an antigen and then presented to all artificial cells in the system. The closest biological analogy to this could be that the system represents a single lymph node and during a particular point in time an antigen presenting cell (APC) presents a single antigen to all cells within that lymph node. In contrast to this, the cells in ProAdDCS have a notion of location based on particular web documents and are allowed to move from one document to another document. If they are stimulated, they will then react. This is more akin to a B cells moving through the body, and the antibodies are ready to bind with an antigenic pattern at any time [4]. The AISEC creates a scenario where every B cells is forced to evaluate each new piece of data. For ProAdDCS, it relies on cells finding out new antigenic patterns then assess their affinity. In terms of functionality, the differences between AISEC and ProAdDCS are outlined below:

- Unlike AISEC, which works as a passive filter, ProAdDCS is an active document filter which is based on incoming information item.
- While in AISEC the relevant words adapt over time, in ProAdDCS the relevant words are both shared by all cells and fixed during the lifetime of the algorithm. Instead, the transformations performed on these words change during the algorithm lifetime.
- Classification for ProAdDCS is not a binary classification based on classifying into relevant or non-relevant documents, but their ordering is according to decreasing relevance. The user's feedback triggers the profile's to adapt to changes on users' interests. Whenever a document is relevant to the current topic of interest then the learning process takes place.
- Unlike AISEC, where the affinity between two cells was measured as the proportion of common keywords in their vectors, in ProAdDCS the cosine similarity measure between the two vectors is adopted. A document is assigned a relevance score equal to the highest achieved cosine similarity to

the B cells. Whenever a document is relevant to the current topic of interest then the learning process takes place.

6.2.2 The Flow Chart

A diagrammatic depiction of the ProAdDCS algorithm flow is shown in Figure 6.1. The following explanation can be quite complex and it is hoped that this flowchart is useful as a reference to show where each section of the following explanation fits in the overall algorithm. Following the layered framework which has been described in Section 3.2.1, the following description is explained.

6.2.3 Representation

Each artificial immune cell will encode:

1. A summary of the user's interest
2. A summary of the user's specified document
3. A set of numerical weights indicating which terms of a document are more specific to the underlying topic(topics) of interest
4. A count relating to stimulation

In (1) the users interest on a certain topic must be summarised. The cell must encode this so as to be able to determine the relevance of any antigens (web document). The summary of the user's interest (user profile) comprises of a vector of terms. This vector carries a set of terms relevant to the user's topic of interest and is therefore referred to as the Interest Term Vector (ITV). This vector is not variable in size and will carry the n most important words as ranked out of all words found in the training documents (where n is a user defined parameter).

Similar with cell in (1), vector for cell in (2) carries a set of words relevant to the users specified document and is therefore referred to as the Relevant Words Vector (RWV). The set of attributes include the list of word relationships used by WordNet to extract a semantic concept between term. This vector is not variable in size and will carry the D most important words as ranked out of all words found in the training documents (where D is a user defined parameter).

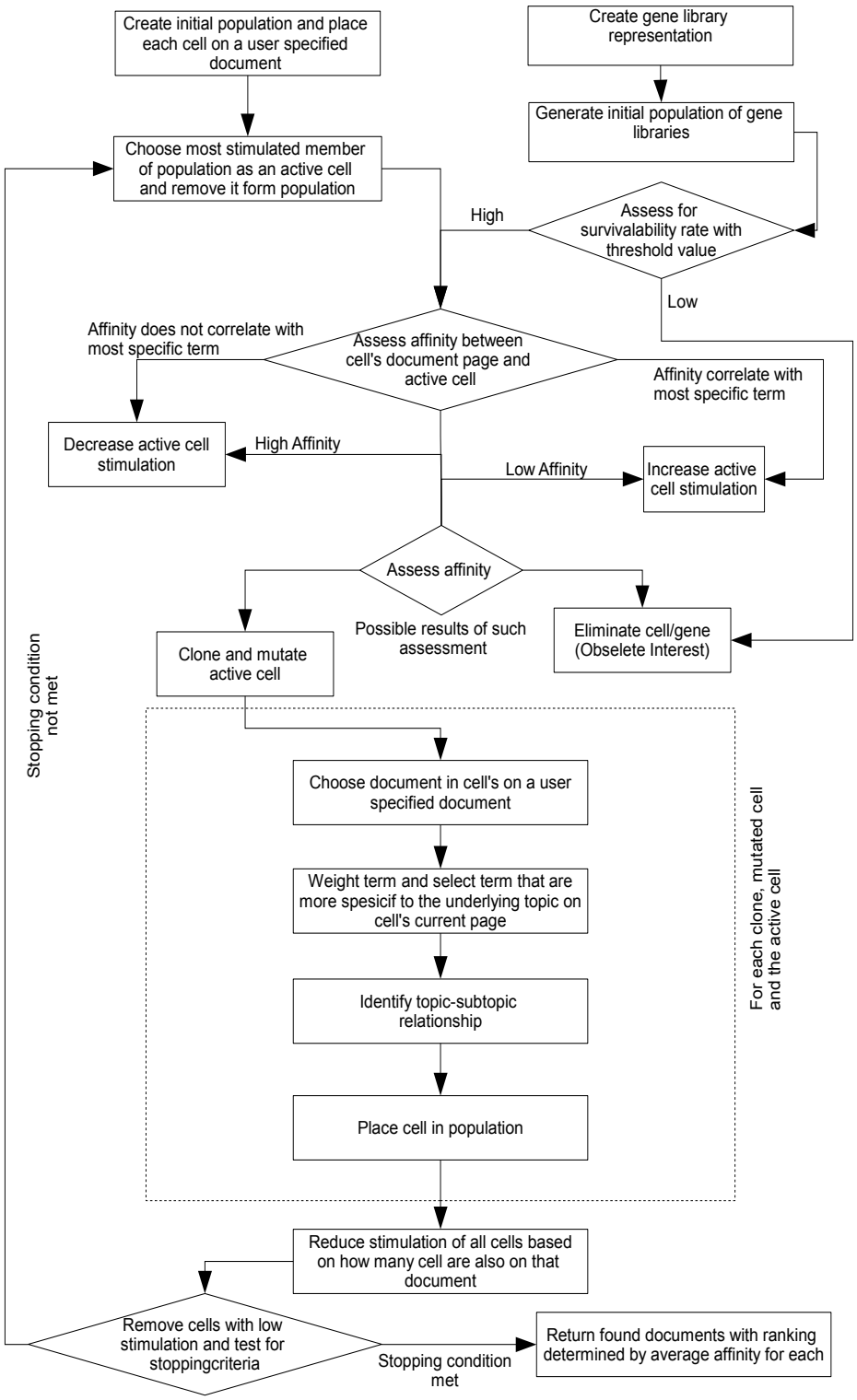


Figure 6.1: The ProAdDCS Flow Chart

The set of attributes describing a cell's content of a specified document i.e. the document title, with most specific to the user's topic of interest (3) does not contain a list of interesting words, but rather a list of weighting term that are more specific to the underlying topic. This vector is therefore referred to as the Term Weighting Vector (TWV). This vector is the same length as the ITV and RWV, with each position containing exactly one transformation that is legal to apply to the corresponding element of the RWV. These transformations form the adaptable part of the immune cell and so, in contrast to the ITV, will change. For simplicity, this vector is actually a vector of numbers, where each represents a term weighting on the document in the corresponding position in the RWV. The goal of the AIS is to change the elements of the RWV with the changes on ITV to find the most relevant document for the user based on topic (topics) of interest. This is guided by the evaluation function to be described later.

Finally, each cell carries a real number representing a level of stimulation for that cell (4). Cells with low stimulation are removed from the population. The details of how the variation in stimulation is calculated are made explicit in Section 6.3.3.

Process for Extracting Semantic Concept From WordNet

The aim of ProAdDCS is to find relevant documents which match to the user's interest in the user profile and be able to adapt to variations of a user's need (interest). Adaptation is not only taking place for the single topic of interest but as well as multi-topic of interest. It is therefore important to employ a strategy for determining the most relevant document among the large number of web document pages. Given an initial set of (indirectly) terms relevant to the user's topic of interest in the ITV it is possible to generate words that satisfy both criteria using WordNet and employing the hypernym (generalisation), hyponym (specialisation) relationships. The hypernym and hyponym relationship in WordNet form hierarchical operations. The hierarchy is followed from a given level of the hypo/hypernym hierarchy for a given number of levels, which is a user defined parameter, w. The WordNet's hypernym and hyponym relationship is exploited in order to determine whether fewer but more general concepts can be obtained and to maintain a sufficient diversity of terms thus, improved classification ability. The synset relationships allow the system to generate diverse terms (keywords) to improve searching for relevant documents. The relationships will generate words that are related to a noun of relevant word but contains slightly different meanings and each of those words maybe useful to the search. The de-

scription on WordNet can be found in Section 4.2.1.

The process used to create a set of words from the RWV using WordNet is straightforward. The Java WordNet Library (JWNL) is used to create an interface between ProAdDCS and WordNet. JWNL is a Java API for accessing the WordNet relational dictionary. JWNL is freely available from the Sourceforge website[1] and is released under the BSD licence. JWNL version 1.3 is used for this work. Given a word w at position i in the RWV, the corresponding operation identifier o at position i in the RWV is retrieved. Using JWNL the set of words that are returned when the operation o is applied to w is determined. Appendices G show an example of XML source code to create the interface between ProAdDCS and WordNet.

Determining the correct concept for an ambiguous word from several synsets is difficult, as is deciding the concept of a document containing several ambiguous terms. In this work the synset is not used directly but rather take an advantage of the synset's *gloss*,which explains each concept and gives an example sentence. For example, the gloss of the word "orange" with the fruit concept is "round yellow orange fruit of any of several citrus trees"; with the color concept it is "any of a range of colors between red and yellow." The relationship of WordNet and ProAdDCS is done by converting the semantic lexicon into its hypernym version word by word and topic by topic. A semantic lexicon is built by collecting the word frequency for a topic and transforming each word into a significance vector. Next, the significance vectors of words occurring is added in a document and is normalized. The process started with the word-topic occurrence matrix, describe as

$$\begin{vmatrix} 0_{11} & 0_{12} & 0_{13} & .... & 0_{1M} \\ 0_{21} & 0_{22} & 0_{23} & .... & 0_{2M} \\ ... & ... & ... & .... & ... \\ ... & ... & ... & .... & ... \\ 0_{N1} & 0_{N2} & 0_{N3} & .... & 0_{NM} \end{vmatrix}$$

where o_{ij} is the occurrence of word i in topic j, M is the total number of topics, and N is the total number of different words.

Each ambiguous word in the original lexicon contains several senses and each sense has its own gloss. Each gloss is treated as a small piece of the document with a core concept and the gloss is transformed based on Equations 6.1 and Equation 6.2 for the document vector, $\boldsymbol{d}$ where N is the number of words in document, d. The broad conceptuality of Equations 6.1 is similar as in TFIDF for weighting factor in text retrieval and data mining which reflects how important

[1]available online (http://sourceforge.net/projects/jwordnet/)

a word is to a document in a collection or corpus.

$$w_{ij} = \frac{o_{ij}}{\sum_{j=1}^{M} o_{ij}} \times \log \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} o_{ij}}{\sum_{i=1}^{N} o_{ij}} \tag{6.1}$$

$$\boldsymbol{d}_{ij} = \frac{1}{N} \sum \boldsymbol{w_{ij}} \tag{6.2}$$

To determine the gloss for an ambiguous word, the specific element weights of each gloss in the specific topic of the original semantic lexicon are compared. Then, the gloss vector with the highest weights in the specific element is chosen to represent the original word. For example, we compared the first element weight of all gloss vectors for an ambiguous word of topic 1. Then, going up w levels in the hypernym tree, this hypernym is then used to build our hypernym version of a semantic lexicon for all terms in all topics. We believe that the process described above can theoretically reduce the total number of words in a data set.

6.2.4 The Affinity Function

The affinity of a cell with a web document is calculated using the words found in the cell's ITV and words generated by the cell's RWV. The affinity calculation begins by generating the set of interest words from the cell's ITV and RWV using the transformations as defined Section 6.2.3. The web document is processed into the form of an antigen as described in Section 6.2.3 to give a set of terms. The number of terms generated by WordNet using the transformations from the RWV that also appear on the web document is counted and then normalised by the length of the number of words generated by the WordNet process. Similarly the ITV is compared against the web document and the count of the number of words present in both the document and the ITV is normalised by the length of the RWV. In ProAdDCS the cosine similarity measure between the two vectors are adopted. A document is assigned a relevance score equal to the highest achieved cosine similarity to the cells. Whenever a document is relevant to the current topic of interest then the learning process takes place. To calculate the affinity, the mean of these scores is taken. The result is the affinity between the antigen and the immune cell and by definition will return a number in the range [0,1]. It is important to give a reason for the choice of term indexing and weighting scheme. Similarly, a distance measure between the RWV and a web document must be defined. More thoughts on these metrics are explained under the process of ProAdDCS in the next section.

6.3 The process of ProAdDCS

The following sections describe the main processes for the ProAdDCS. In general, the process involved are: initialisation, extracting informative term, running, dynamic population on profile term and the automated feedback. Throughout the explanation on ProAdDCS processes, *terms* are used rather than *words* because the *terms* that are considered may be parts of words, single words or combination of words [27].

6.3.1 Initialisation

The purpose of initialisation is to create an initial set of immune cells trained to recognise relevant web documents and place them at a suitable repository. The system is initialised using a set of user specified interests called profile. The importance of these specified documents cannot be overstated as they are used to summarise the user's prior "knowledge". In constructing a user profile, the user is the only source of information about what is of interest. A user profile is a long-term representation that is initialised once and has then to be adapted to changes in the user interests. It is therefore feasible to ask the user to provide more than a set of keywords for profile initialisation. In the case of the "1-1" and population strategies for multiple-topic representation, the user is usually required to specify a set of documents for each topic of interests. Whatever the case, a set of user-specified documents provides both the pool of candidate profile terms and the necessary information for weighting and selection.

When profile terms are selected out of the unique terms in the user specified documents, relevance information is implicitly taken into account. In that sense, when a vector representation of a user profile (ITV) is built out of terms extracted from relevant documents, correlations between terms are implicitly taken into account. The term weighting scheme can exploit the relevance information provided by the user specified documents to measure the specificity of terms to the underlying topic. A term that is specific to a topic can distinguish relevant documents from non-relevant documents. Therefore, specific terms are particularly importance when building a user profile.

We argue that it is easier for the user to specify relevant than non-relevant initialisation documents. The space of non-relevant documents is considerably larger than the space of relevant documents. Therefore, a small number of relevant initialisation documents for each topic of interest is better and are more general expectation of a real situation.

6.3.2 Extracting Term from Initialisation Documents

Web documents tend to contain a certain amount of noise, whether this is from adverts, navigation panels or simply a general mix of topics on one document's page. Therefore it requires an indexing process to develop a document representation by assigning content descriptors or *terms* to the documents. The indexing schemes are categorized as single-term indexing and multi-term or phrase indexing [26]. Steps involved in the indexing for a web documents usually consist of:

- remove HTML mark-up tags
- recognize terms or phrases
- use a stop-list to eliminate unwanted words that carry no information such as words like pronouns, preposition, conjunction etc., ⇒ *stop word*
- perform suffix removal to generate word stem ⇒ *stemming process*

The following process involved term weighting and selection. Term weighting process in this work involved two types, firstly term weighting for document indexing and secondly term weighting for topic representation.

Weighting Term for Document Indexing

In term weighting for document indexing, the purpose of the process is to extract out of the user's specified documents those terms that are more specific to the underlying topic (or topics). From this, a user-specific vocabulary can be identified that distinguishes the documents of interest from the rest of the collection. The assigned weights can then be used to extract an absolute number of the most specific terms or those with weights over a certain threshold. The extracted terms are used to populate the profile.

The ProAdDCS involved dynamic information source, therefore, the better estimation for weighting terms are the *Relative document frequency* (ReIDF). The ReIDF is a measure of the relative importance of terms within the user specified documents and a general collection of documents. Some theoretical advantages of ReIDF are as follows [22] cited in [27]:

1. ReIDF does not require non-relevant documents. It uses probabilities of appearance, which make accurate estimations possible even in the case of a small number of initialisation documents.
2. The statistic (probabilities) can be updated online and therefore, the method is applicable in the case of dynamically compiled document collections

3. ReIDF is not dependent on the number R of initialisation documents therefore it can be applied both in batch and an online mode

The process of weighting terms based on ReIDF found in the initialisation documents proceeds as follows. The initialisation documents are first concatenated to form one single document. The method of ReIDF assigns to each term, a weight in the interval (-1,1), according to the difference between the term's probabilities of appearance in the user specified documents and in the general collection. Given R as the number of (relevant) initialisation documents, then the weight of a term t that appears in these documents is calculated using Equation 6.3 based on the notation of the contingency table in Table 6.1, where the first part of the equation $\frac{r}{R}$ favors those terms that describe the user specified documents and therefore the underlying topic of interest, while the second part $\frac{n}{N}$ biases the weighting towards terms that are specific within the general collection.

		Document		
		Relevant	Non-Relevant	Collection
Term	+	r	$n - r$	n
	-	$R - r$	N - R - n + r	N - n
		R	$N - R$	N

Table 6.1: Contingency Table [139]

$$\text{ReIDF} = \frac{r}{R} - \frac{n}{N} \tag{6.3}$$

Weighting Term for Topic Representation

Previously, we summarised the process of term weighting for document indexing. The process described above is to estimate how closely a term is related to the document's content or how specific it is with regard to the complete document collection. While for term indexing for topic representation, the purpose is to estimate the association between a term and the topic of interest. The process is based on the differences in the distribution of terms between the complete collection, a set of documents that is relevant to the topic of interest, and, in some cases, a set of documents that is not relevant to that topic. Here we based it on the relevant document frequencies (RDF) which exploit relevance information (see Equation 6.4). The assumption is, those terms that appear in the majority of

the documents are more strongly associated to the document's topics than terms that occur less. Some of the profile terms will broadly define the underlying topic, while others co-occur with a general term and provide its attributes and related concepts. Thus, terms are ordered according to decreasing RDF.

$$\text{RDF} = r \tag{6.4}$$

We assign *topic-specific* weight to terms in the relevant set. A contingency table (Table 6.1) summarise the term distribution in the document set [139]. It is based on the differences in the distribution of terms between the complete collection, a set of documents that is relevant to the topic of interest, and, in some cases, a set of documents that is not relevant to that topic. The symbol $+$ and $-$ in Table 6.1 indicates the term occurs or does not occurs in document respectively.

6.3.3 Running

The length of the RWV was set at 50 terms as this was considered a reasonable length, trading accuracy of the result for speed. Once the ReIDF values for the most frequent 500 terms have been computed, the terms are ranked and the top 50 are selected to form the cell's ITV. An initial set of immune cells is then created using the same 50 term ITV. The RWV of each is populated by choosing WordNet transformations from the set of terms that are related to a noun or relevant term but contain somewhat different meanings and each may be useful to the searching mechanism and unlike the ITV, the RWV of each cell will therefore be different. Each cell's stimulation level is initialised at a user defined value, and the location of each cell is set to a starting point of that document. This is chosen at random from the small set of pages specified by a user. The system is then ready to begin the running stage.

In the running stage, an order of cells must be established with which to process the members of the population. There are a number of options, the simplest being that each cell is processed in turn until all have been examined, at which point the process will begin again from the start. The population is held in a sorted queue where the order of the queue is based on cell stimulation level. The higher the stimulation level of a cell, generally the better that cell is doing at finding relevant documents. Therefore it was decided that the most stimulated cells, those at the head of the queue, should be tested first. During each iteration the cell at the head of the queue is removed, this is referred to as the "active cell", and the procedures of immune cell movement and selection, assessing relevancy using affinity, cloning and mutating and automated feedback are applied.

Cell's Movement and Selection

Each immune cell must make a choice of the document it is to move to next, this allows the search space to be explored. It should be noted that extracted terms that appear frequently within each other's topical context and/or appear close to each other, are linked with large weights. This is a way to identify topic-subtopic relations between terms. As a result, extracted terms that are frequent in general may be placed in a high rank, although they are not specific to the underlying topic. This ordering takes into account both the generality of terms within the user specified documents and their specificity within the general collection. If two terms have the same RDF or weight then they are ordered alphabetically. Therefore, there is always a difference between the rank of different terms. This process forms a hierarchical profile term link. To visualize, refer to Figure 6.2. From the figure, terms at the top of the hierarchy are more specific to the user interests. They correspond to concepts that relate to the specific topic of interests. For middle hierarchy is a less specific terms. These are concepts that relate to subtopics of interest. Finally, at the lowest levels of the hierarchy appear terms that comprise the sub-vocabulary used when the topic is discussed. If a strong associative link exists between two terms of different ranks, then it may refer as a relation of topic-subtopic. For multiple topics of interest, the same process maybe applied on a single set of documents that relates to multiple topics of interest. The value associated with the chosen weight (of term) for a particular document is stored as this is now the estimated relevance of the target document and is used to provide an automated feedback.

Assessing Relevance using Affinity

This stage of the running process requires each cell to assess its affinity with the document it has moved to. Therefore, this is the assessment of the relevance of the document. The current document is processed as described in Section 6.2.3 and WordNet based transformations are applied to each term in the RWV in the same manner as described previously. The result of these transformations produces a set of terms. This set of terms is used in the affinity calculation between the document and the cell as described in Section 6.2.4. The affinity between the cell's document and the cell is stored for the purposes of ranking the results to be shown to the user upon completion of the run. If the current document has not been seen by any cells before, then the affinity value is associated with the document and a record of the active cell's ITV is stored. If the affinity between the document and the active cell is greater than that already stored the current

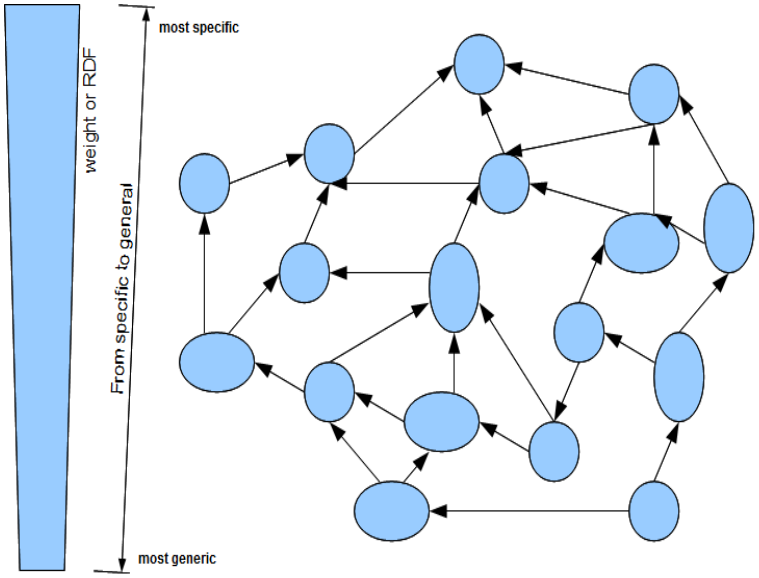


Figure 6.2: A visualization of hierarchical profile of terms

cell's ITV and affinity value will replace the stored value. This affinity value will determine the number of clones produced, which will be described in the following section.

Cloning and Mutation

If the affinity of the cell with the document is above a threshold, the cell has found what is considered to be a relevance document. This is rewarded with the ability to clone and mutate. Both cloning and mutation will be performed with regard to the affinity; the number of clones being proportional to affinity while the number of mutations being inversely proportional to affinity. The number of clones produced based on the affinity of a cell with a document can be defined with the following Equation 6.5,

$$num_{clone} = \begin{cases} \lfloor (k_{clo} \times \text{aff}_{bc,r}) - k_{ct} \rfloor \\ \text{if} > 0 \lfloor (k_{clo} \times \text{aff}_{bc,r}) - k_{ct} \rfloor \\ 0 \text{ otherwise} \end{cases} \tag{6.5}$$

where k_{clo} is a constant controlling the rate of cloning and k_{ct} is a constant that controls the maximum number of clones. The $\text{aff}_{bc,r}$ is the affinity of the cell to be cloned with specified document, r. Meanwhile, the number of mutants produced is defined in Equation 6.6,

$$num_{mutate} = \lfloor (1 - \text{aff}_{parent,r}) \times k_{mut} \times \mid \text{parent}_{RWV} \mid \rfloor \tag{6.6}$$

where, $\text{aff}_{parent,r}$ is the affinity of the parent cell with specified document, r. k_{mut} is a constant controlling the rate of mutation and parent_{RWV} is the parent cell's set of term transformations. Upon cloning, each of the new cells receives the ITV and RWV of its parent. Mutation then occurs and the values in that vector are replaced by other legal values only. After mutation, the location of each clone is temporarily placed on the same location as its parent. Once the clone moved, the cell is initialised with a default stimulation level and is placed in the population.

Automated Feedback

The number of documents that received relevance feedback may vary, depending on the characteristics of the user, the time constraint, the success of the filtering process and other parameters. It may range from one document to many. From the solution based on ReIDF, it attempted to account for the statistical importance of the sample of feedback documents. If the sample is statistically important ($R > 20$), then the weight of term is calculated based on the Equation 6.3. While in the opposite case ($R < 20$), the weight of term is based on the reflection of $R = 20$.

ProAdDCS uses a confirmation signal mechanism similar to that in AISEC, but as the user is not in the loop this confirmation signal must be given automatically. When a cell moves from one document to another it has made an implicit judgment based on relevance about where to go. This judgment is expressed by an estimated value of the degree of relevance of the document where the cell will move to. A high estimated value of relevance can be considered analogous to 'signal one'. The cell can then measure the actual relevance of (and affinity with) the new document where it moved to, considering the entire text of that new document collection. If the estimate and the actual value are differ just a little then the cell has made a correct decision, 'signal two'(a confirmation signal) occurs and the cell will be rewarded. If the estimate and actual value differ greatly then the cell must be penalised.

Upon the signal two, the co-stimulation model is used to stimulate or suppress cells based on their quality. It is reiterated that the user cannot do this in an interactive manner so an automated scheme must be implemented. As cells move between documents they do so in a probabilistic manner, they do not always move to the "best" document out of a set of potential pages. This is to promote diversity of search. The affinity score cannot be used to determine signal two, as the affinity with the document could be "low" while the cell is otherwise useful. This is why cell stimulation is varied based on the difference between the

estimated and actual affinity rather than an absolute value based on the affinity. The stimulation of a cell is proportional to the quality of the estimate it made regarding the current document's relevance as shown. Note that the stimulation of a cell will always decrease, thus bounding a cell's lifetime, which again much as seen in AISEC system. Without this bounding it is theoretically possible for a few cells to suffer continual stimulation and therefore dominate the population. This would presumably lead to a significant reduction in diversity.

To determine cell stimulation level, Equation 6.7 is used. The stimulation of a cell at time $t+1$ is calculated where aff is the affinity of the cell, c, with the document, r, (or known as antigen). While aff_{est} is the estimated affinity. The value of 10 is set as the arbitrary constant for the cell's stimulation level.

$$stimulation_{t+1} = stimulation_t - abs(10 \times (\text{aff}_{est} - \text{aff}_{c,r})) \tag{6.7}$$

6.3.4 Establishing Population Dynamics

In the previous step cells may be punished for finding documents that are not relevant. This is achieved by reducing a stimulation counter for the cell. It is important to take care against the redundant cells (those in area of the search space that is already covered by other, fitter cells) in the population. This is in order to prevent unlimited increasing number of cells. If the number of cells on a single document is above a threshold then each cell currently on that document will incur a penalty of a reduced stimulation count. This reduction will be in proportion to the number of other cells. Given a document on which a number of cells are currently placed, if the population size is low, cells tend to have a chance of moving from that document before their stimulation is reduced below the threshold at which they will be removed. However, if the population size is high, cells will have its stimulation reduced a number of times before it becomes the focus of the main procedure again and can move, thus only the very best few survive. This technique allows the population to dynamically grow as the search area (number of incoming documents) grows. As this suppression only occurs when the number of cells on a single document is above a threshold (a user-defined parameter), it does not impart a global limit on the numbers of cells in a population, but imposes population restrictions on a local level which tend to result in global population control. At the end of each iteration each cell's stimulation count is checked. If it is found to be below a threshold the cell is removed from the population. Otherwise the cell is maintained in the population. From the explanation on 'cell' overview, the next section discussed the processes

involved in updating a profile term, removing the incompetent profile term and adding a new term. The explanation shows that the overlap between the profile and the extracted terms has a significant effect on the adaptation pace.

Updating a Profile Term

Updating a profile term concerns with the process on those extracted terms that already appear in the profile to be updated. For each such profile term, t, an update weight $\acute{w}_t$ is calculated based on Equation 6.8, where D refers to a document.

$$\acute{w}_t = \begin{cases} w_t + w_t^D & \text{if } D \text{ is relevant} \\ w_t - w_t^D & \text{if } D \text{ is non-relevant} \end{cases} \tag{6.8}$$

Furthermore, in the case of a relevant document, the additional weight that has been assigned to the profile terms is summed up and then the sum is subtracted evenly from all profile terms. This process is expressed by Equation 6.9 where NP is the number of profile terms. The opposite takes place in the case of a non-relevant document. Therefore, given a profile term with a specific set of terms, this last process assures that the overall weight of profile terms remains stable.

$$w_t^n = \begin{cases} \acute{w}_t - \frac{\sum_{t \varepsilon D} w_t^D}{NP} & \text{if } D \text{ is relevant} \\ \acute{w}_t + \frac{\sum_{t \varepsilon D} w_t^D}{NP} & \text{if } D \text{ is non-relevant} \end{cases} \tag{6.9}$$

Removing Incompetent Profile Terms

Another side-effect in the weight of profile terms which is caused either implicitly in the case of a relevant document or explicitly in the case of a non-relevant one, is that the weight of some profile terms become less than zero. Terms that run out of weight are purged from the profile. With this mechanism, it is aimed to remove terms that were mistakenly added to the profile or that have become incompetent(obsolete) due to changes in the user interest. This kind of alteration gives the profile ability to forgetting unexciting or a non-relevant topic.

Adding a New Term

Having updated the weight of profile and removed incompetent terms, those terms that have been extracted from a relevant document and do not appear in the profile are added to the profile. Adding the terms do not replace terms that have been removed. There is no relationship between the number of removed terms and added terms. The number of profile terms is not fixed, but rather

changes dynamically according to user feedback. After the new terms are added, the sum of the initial weights of those terms that have been purged is subtracted evenly from all profile terms. This is expressed by Equation 6.10 where NP' is the number of profile terms after the addition of new terms. While W_{purged} is obtained based on the Equation 6.11. This is done to avoid the escalation of the overall weight of profile terms due to the addition of new weight with every new term.

$$w_t''' = w_t'' - \frac{W_{purged}}{NP} \tag{6.10}$$

$$W_{purged} = \sum_t \text{purged} w_t^{init} \tag{6.11}$$

6.3.5 Returning Result

When a stopping criterion is met, the user is presented with a ranked list of results. This stopping criteria may be a certain number of relevant documents that have been discovered or a certain number of iterations have taken place. These results consist of a list of relevant documents that at least one cell visited during the run of the algorithm. For each document found during a run, the mean affinity between all cells that encountered that document and the document itself is computed. The document are then ranked according to this mean affinity, the higher the mean affinity, the higher the ranking of that document.

6.4 Algorithm Description for ProAdDCS

This section presents the pseudo code for the ProAdDCS, as described in the preceding section. The cell representation is defined as follows:

$$\text{B cells} = < ITV, RWV, TWV, stimulation >$$
$$\text{where,}$$
$$\text{ITV} = < term1, term2, ..., term_w >$$
$$\text{RWV} = < word1, word2, ..., word_n >$$
$$\text{TWV} = < weight1, weight2, ..., weight_r >$$

Let BC refer to an initially empty set of naive immune cells (B cells) where bc is used to denote one element of BC, that is, one individual cell, where also:

- bc_{ITV} is the set of relevant words related to bc. E.g. $< football, club >$.

- bc_{RWV} is the set of terms from WordNet transformations related to bc. E.g. $< english, league >$.
- bc_{TWV} is a set of real numbers for the current position of weight in the hierarchical of profile term E.g. $< 0.32, 0.12 >$.
- bc_{stim} is a real number representing bc's current stimulation level.

In general, the main algorithm features of ProAdDCS are defined in the Algorithm 10 and Algorithm 11. The main algorithm consists of 8 stages within a loop. The stages are:

1. Choose next cell of population
2. Check if cell's current location (specified initialisation document) is legal, if not then backtrack
3. Compute affinity between cell and document
4. Perform automated feedback on cell and stimulate or suppress cell based on outcome
5. Clone and mutate cell based on affinity, picking a new document for each new cell and the parent cell to move to next
6. Estimate and remember the estimate of quality for this new document
7. Add new clones to population
8. Perform population meta-dynamics. That is, updating cells, removing the incompetent cells and adding new cells in order to avoid a significant increase in the population size. The population is also ordered by descending stimulation level

Next, the initialisation, the affinity procedure and the cloning and mutation procedure are described in detail.

Initialisation

This procedure produces a set of cells, the number of which is dictated by $Init_{size}$. Then, a set of all words in all training documents ($Init_{train}$) is generated where te is an element of $Init_{train}$. Detailed procedure for initialisation is described in Algorithm 12.

```
input : S = set of terms to be recognised in a document,
        n = number of elements selected for removal
output: M = set of memory detectors
begin
    Create an initial random set of detectors ;
    for terms in S do
        Determine the affinity with each detector ;
        Generate clones of the detectors with the highest affinity ;
        Mutate attributes of these clones inversely proportional to their
        affinity ;
        while mutation is needed do
            Selects a random site within the detector;
            Select a random term from the gene library;
            Replace the term in the detector ;
        end
        Add these clones to the detector set;
        for detectors with highest affinity do
            Place a copy into the memory detector set, M;
        end
        Renew memory detectors;
        while renewing memory detectors do
            Check detector suppression and stimulation levels;
            Place surviving detectors in the memory detector set, M;
        end
        Replace the n lowest affinity detectors in the memory detector set
        with new randomly generated detectors;
    end
end
```

Algorithm 10: General feature for ProAdDCS

Affinity

Given a current cell, bc, and a document processed into the form of an antigen, ag, the affinity between bc and ag is illustrated by Algorithm 13. In this pseudocode, $count_{x,y}$ is a count of features found in both x and y. TWV is a vector of term weighting generated by WordNet using the RWV and the WordNet transformations defined by the elements of ITV. Therefore $count_{INT,ag}$ is a count of features found in both ag and the set INT. The result of this function by definition will always return a value in the range [0,1].

Cloning and Mutation

Algorithm 14 shows the procedure used for cloning a cell a number of times, and mutating those clones. The number of clones is proportional to the affinity

```
begin
    Create gene library representation;
    Generate gene library ;
    Select a random portion of genes in detector clones;
    Perform gene mutation ;
    Replace the mutated gene in the detector's feature vector;
    Remove incompetent gene from library;
end
```

Algorithm 11: Gene Library for Maintaining Diversity in ProAdDCS

of the cell and is calculated according to the equation given in Equation 6.5. The number of positions of the RWV vector to be mutated in each clone is inversely proportional to the affinity of the cell and is calculated by the Equation 6.6. The symbol $\lfloor x \rfloor$ denote the floor of x, that is x rounded down to the nearest integer.

6.5 Experimental Evaluation and Methodology

IF systems are by nature interactive. They don't only provide the user with relevant information, but also require the user's involvement for both profile initialisation and adaptation. In the previous section, a ProAdDCS has been described. ProAdDCS is an algorithm proposed for adapting a user profile in IF domain which is inspired from dynamic clonal selection. The goal for ProAdDCS is to adapt multiple-topic profile both to short-term variations in the user's need and to progressive, but potentially radical changes in long-term interests. In this section, ProAdDCS is tested in order to evaluate its performance via real corpus of web documents. The profile is tested for their ability to adapt over time in the content of documents. The evaluation is carried based on simulation procedure whereby it involves the use of *virtual* or *synthetic user* which is used to simulate such radical changes. Given a pre-classified collection of documents, a virtual user's current interests are defined by a subset of the classification topics. Training documents that relate to the topics in the subset comprise the positive feedback. Interest changes can then be simulated by modifying this subset. To simulate the loss of interest in a topic, it is removed from the subset. Similarly, the emergence of a new topic of interest can be simulated by adding a new topic to the subset. System can therefore be tested against radical drifts in the topic of interest.

Although evaluation based on sample of users may provide a good insight into the human related issue that IF systems have to resolve [48, 53], the hetero-

PROCEDURE Initialise ()
Initialise T as null ;
Initialise t as null;
Initialise BC as null ;
Initialise SCORE as null ;
foreach *(T $\varepsilon Init_{train}$)* **do**
  foreach *(term t in te)* **do**
    T $\longleftarrow$ T $\cup$ t
  end
end
foreach *(w ε W)* **do**
  RF = Relative frequency of term, t as computed ;
  TW = term weighting of t in $Init_{train}$;
  $tscore$ = ReIDF of t as computed ;
  SCORE $\longleftarrow$ SCORE ε (t,$tscore$)
end
T_{top} = Determine top Kt terms as ranked by $tscore$ in SCORE ;
DO $Init_{size}$ TIMES
$BC_{ITV} \longleftarrow T_{top}$;
$BC_{stim} \longleftarrow K_{stim}$;
foreach *position i in bc_{RWV}* **do**
  i $\longleftarrow$ random value in range [0,4]
end
$BC_{pos} \longleftarrow$ random element of $Init_{start}$;
BC $\longleftarrow$ BC $\cup$ bc ;
Return BC ;

Algorithm 12: Initialisation Procedure

geneity of users and the difficulties in controlling the experimental parameters render this kind of evaluation difficult to reproduce [44]. Furthermore, simulated experiments can be reproduced accurately and it has been claimed in [27] that experiments with simulated users were more conclusive than experiments with real users.

To evaluate our approach, we adopted an evaluation measure based on *Average Uninterpolated Precision* (AUP) measure. The AUP (Equation 6.12) of a given topic, τ is defined as the sum of precision value of relevant documents in the r top ranked documents divided by the number of relevant documents for that topic, $R(\tau)$. Hence, relevant documents which do not appear in the top r ranked documents receive a precision score of 0:

$$AUP(\tau) = \frac{1}{R(\tau)} \sum_{i=1, y_i=+1}^{r} \frac{\mid \{j \mid y_j = 1 \wedge rank(j) \leq rank(i)\} \mid}{rank(i)} \tag{6.12}$$

For example, if the first 5 out of a list of 10 documents are relevant to a specific

PROCEDURE Affinity(bc,ag)
INT $\longleftarrow$ null ;
foreach *(location i in* bc_{RWV}*)* **do**
  t $\longleftarrow$ term in location i of bc_{ITV} ;
  int_{term} as generate set of terms using wordNet operation in location i of bc_{RWV} ;
  INT $\longleftarrow$ INT $\varepsilon(int_{term})$
end
aff $\longleftarrow \frac{1}{2} \times (\frac{count_{bc_{ITV},ag}}{|bc_{ITV}|} + \frac{count_{INT,ag}}{|INT|})$;
RETURN aff

Algorithm 13: Affinity Function Procedure

PROCEDURE CloneMutate(bc,Affinity)
Set numClones as null ;
Set numMutate as null ;
numClone $\longleftarrow aff \times K_{clo} \perp -K_{ct}$;
numMutate $\longleftarrow \perp (1 - aff) \perp \times \mid bc_{RWV} \mid \times K_{mut}$;
DO numClone TIMES
  bcx as a copy of bc;
  DO numMutate TIMES
    p as a random point of bcx's feature vector ;
    i as random value in range [0,4] ;
    replace value in bcx_{RWV} at location p with i
  end
end
$bcx_{stim} \longleftarrow K_{stim}$;
numClones $\longleftarrow$ clone ε bcx;
RETURN numClones

Algorithm 14: Cloning and Mutation Procedure

topic and there are a total of 100 relevant documents, then the AUP score of this list is $AUP = (1/1 + 2/2 + 3/3 + 4/4 + 5/5)/100 = 0.05$. If the last 5 documents in the list are relevant, the corresponding AUP score becomes $AUP = (1/6 + 2/7 + 3/8 + 4/9 + 5/10)/100 = 0.0177$.

It should be noted that the evaluation of IF systems has benefited by the long experience in the evaluation of Information Retrieval (IR) systems [140]. IR system have been traditionally evaluated in the basis of *precision* and *recall*. For the IF system that produces an ordered list of documents, measures that combine precision and recall are suggested [140]. The AUP measure is a combination of precision and recall with an absolute value that depends on the total number of relevant documents. Thus, the AUP measure is the suitable measure which will be used to evaluate the performance the proposed ProAdDCS.

To evaluate this continuous learning approach, we have initially performed

experiments using a variation of the TREC-2001 filtering track[2] for experiments described in Section 6.6.1 until Section 6.6.3 and Reuters-21578[3] document collection for the experiment described in Section 6.7. The objectives for these experiments are presented in the respective section.

TREC-2001 filtering track adopts the Reuters Corpus Volume 1 (RCV1). The latter is a archive of 806,791 English language news stories that recently has been made freely available for research purposes. The stories have been manually categorised according to topic, region and industry sector. The RCV1 is split into 23,864 training and 782,927 test stories and is categorised into 84 out of the 103 topic categories. Documents in a training set have been ordered according to their date of publishing. Therefore, the distribution of documents per topic during an online training phase reflected the temporal variations in the publication date of documents about each topic. In this experiment, we make an assumption that these variations reflect changes in a virtual's user's short-term needs. The training documents were preprocessed by stop word removal and stemming using Porter's algorithm [148]. After training documents, the profile is used to assess the relevance of a document in the test set. An AUP score was then calculated for each topic, on the basis of the best 3000 scoring documents.

6.5.1 The Comparative Approach

To determine the performance of our proposed approach, it was necessary to test it against another similar system which focused on adapting a user profile. Therefore we need a baseline approach in order to evaluate the performance of our approach. In this work, we adopted two types of comparator baseline which are the Rocchio's learning algorithm [23] and the Nootropia system [44], a connectionist based algorithm. The description on these baseline approaches is given below:

Rocchio's Learning Algorithm

Rocchio's learning algorithm is an algorithm for learning user interests that has been well studied in information retrieval (IR) [22, 23] cited by [27]. This algorithm is an example of adaptation in IR system. Systems employing the Rocchio's algorithm typically assume the stability of user interests and apply the algorithm as a batch process. Rocchio's algorithm was first applied in IR for calculating an optimum query out of a set of relevant and a set of non relevant documents. In

[2]http://trec.nist.gov/data/filtering.html
[3]http://www.daviddlewis.com/resources/testcollections/reuters21578/

Rocchio's algorithm, both the user profile and the incoming documents are represented as weighted keyword vectors in a common vector space, with as many dimensions as the number of unique words in the documents' vocabulary. The profile vector is linearly moved towards the vector of an incoming document that received positive feedback and vice versa (see Equation 6.13). The coefficient α, β and γ define the relative contribution of the existing profile and the relevant and the non relevant document respectively, on the new profile's position. The coefficient α plays the role of a decay function and allows the profile to "forget" over time. In other words, it determines how much previous feedback document affect the current position of the profile's vector. A large value of α means a small decay and vice versa. Since the decay is proportional to a keyword's weight, the latter can coverage towards zero. It is important to note that Rocchio algorithm does not include a mechanism for removing keywords from the profile, because it assumes a space with predefined dimensions. Meanwhile, the coefficient β on the other hand, defines how much the user profile is influenced by the relevant document. If β is larger than α the weights of the profile terms can keep on incresing. More detail about Rocchio's algorithm can be directed to [22, 23]. We have carried the preliminary experiments with various value of α (0.90, 0.95, 0.98 and 0.99) and found that it achieves the best AUP score when the decay function, α is 0.95. For the β we choose a value of $\beta = 0.25$ because that value is a better 'fit' when tested on the dataset mentioned above.

$$Q_{t+1} = \begin{cases} \alpha.Q_t + \beta.D & \text{if } D \text{ is relevant} \\ \alpha.Q_t - \gamma.D & \text{if } D \text{ is non-relevant} \end{cases} \tag{6.13}$$

where:

Q_{t+1} is the new query vector
Q_t is the previous query vector
D is the vector of the feedback document

Nootropia's self-organisation Algorithm

Nootropia is a user profiling model based on a self-organising term network, influenced by Autopoetic theory. It is described in detail in [44]. When applied to textual information, Nootropia maintains a weighted network of terms (single words) to represent a user's multiple interest. In Nootropia, a hierarchical term network that takes into account term dependencies is used to represent a user's multiple topics. Given a set of documents about various topics that the user has

specified as interesting, the network is synthesised in three steps:

1. Informative terms are extracted from the interesting documents using a term weighting method. Extracted terms populate the profile.

2. Correlations between profile terms in the interesting documents are identified within a sliding window of 10 contiguous terms. Two profile terms are linked if they appear at least once within the window.

3. Finally, the profile terms is ordered according to decreasing weight.

These three steps syntheses a cyclic term network that formulates a separate hierarchy for each general topic discussed in the documents. A topic of interest discussed in the majority of the user specified documents will be reflected by a hierarchy with larger depth. A hierarchy's depth is therefore a measure of a topic's importance within the profile. Adaptation is then achieved through a self-organising process that allows the profile to respond structurally to variations in feedback. The process involved comprised five deterministic, but interwoven, steps that collectively allow the profile to self-organise in response to user feedback. The process is as follows:

- Step 1: Extract informative terms. Here, the term extraction process results in a set of weighted terms, some of which may already appear in the profile and some may not.

- Step 2: Update profile term weight. The second step of the process concentrates on those extracted terms that already appear in the profile. The effect of this process is an appropriate redistribution of profile term weights that causes a change in the hierarchy's ordering.

- Step 3: Remove incompetent terms. In this third step, terms that run out of weight are purged from the profile together with all of their links to other terms.

- Step 4: Add new terms. The number of terms that are added depends on the semantic novelty of the relevant document in relation to what is being already represented. A document about a topic that is not already covered by the profile will contribute a lot of new terms and vice versa.

- Step 5: Reestablish links. This fifth final step is concerned with updating the link. For this purpose the second step of the profile generation process is referred again.

6.6 Experimenting with ProAdDCS Profile in Adapting Learning and Forgetting Task. Case Study: TREC 2001 Filtering Track

Experiment Objectives: To investigate the profile in adapting, learning and forgetting task, given types of cases.

In this work, we tested the performance of ProAdDCS in three types of cases. There are as follows:

1. Task α : Parallel Interest in Two Topics
2. Task β : A New Topic of Interest Emerges
3. Task γ : Forgetting a Topic of Interest

These cases reflect a radical change in a virtual user's interest. In the experiment, a series of two topic combinations separated by "$\longrightarrow$", symbolising the interest change. Furthermore, we defined the following general tasks, where C represent a two-topic combination, C' the corresponding three-topic combination and T_i defined as a specific topic. For example, a virtual user may be initially interested in topic combination of $T1/T2$ and then additional interest in topic $T3$ emerges. Therefore, the learning task for this scenario is formulated as $T1/T2 \longrightarrow T1/T2/T3(C')$.

The experiment test cases are described further below:

- (α) $T_1/T_2(C)$: This learning task tests the ability of an empty profile to learn from scratch two topic of interest (T_1 and T_2) in parallel. This task involves only one two-topic combination and therefore it does not simulate a radical changes of interest.
- (β) $T_1/T_2(C) \longrightarrow T_1/T_2/T_3(C')$: This task test an existing profile's ability to learn an additional topic of interest. The virtual user is initially interested in topics T_1 and T_2 alone and after some time an interest in the third topic T_3 arises in addition to the existing interests.
- (γ) $T_1/T_2/T_3(C') \longrightarrow T_1/T_2(C)$: The task test the ability of an existing profile to forget one of the initial three topics of interest. Here, the virtual user is initially interested in topics T_1, T_2 and T_3 and then the interest in the first two topics is maintained while the interest in topic T_3 is lost. Therefore, topic T_3 became uninteresting.

For each general task described above, the experiment is carried with specific task formulations which are summarised in Table 6.2. In the next sections, the experimental results are discussed according to the type of task. Each topic combination in a task corresponds to a training phase, a period of time during which the virtual user's interests remain stable. During the training phase, a profile is trained online using a set of documents comprising the first 30 training document per topic in the combination (60 for two topics of interest and 90 for three topics of interest). We only used the first 30 training documents per topic to enable a common experimental setting for all combinations (refer to Table 6.2) including those with a small number of training documents. Although this practice is not realistic, nevertheless, it is not statistically incorrect.

α Tasks	
$\alpha(1)$	R20/R21
$\alpha(2)$	R41/R50
$\alpha(3)$	R10/R68
β Tasks	
$\beta(1)$	R20/R21 $\longrightarrow$ R20/R21/R28
$\beta(2)$	R41/R50 $\longrightarrow$ R41/R50/R2
$\beta(3)$	R10/R68 $\longrightarrow$ R10/R68/R40
γ Tasks	
$\gamma(1)$	R20/R21/R28 $\longrightarrow$ R20/R21
$\gamma(2)$	R41/R50/R2 $\longrightarrow$ R41/R50
$\gamma(3)$	R10/R68/R40 $\longrightarrow$ R10/R68

Table 6.2: List of Task for Learning and Forgetting Task

To evaluate ProAdDCS, we have initially performed experiments using a variation of the TREC-2001 filtering track[4]. TREC-2001 filtering track adopts the Reuters Corpus Volume 1 (RCV1), as described earlier in Section 6.5. Appendices H provide the summaries regarding the test stories based on TREC-2001 filtering track.

The performance of the retrieval on the relevant document is evaluated based on combined AUP score from the topic studied. The approach of AUP has been presented in the previous section, while the approach of combined AUP score has been described in Section 4.4.3. To evaluate a profile, it is tested periodically during the last training phase in each task. In other words, after a radical change of interest has occurred. Note that, for task α it does not simulate radical changes

[4]http://trec.nist.gov/data/filtering.html

therefore task α has only one training phase. The experiment is conducted with 250 runs with different random seed number. After every five training documents, the profile is used to filter the complete document set. A combined AUP score was then calculated for the topics, on the basis of the best 3000 scoring documents. In the next section, the results of the experiments are discussed.

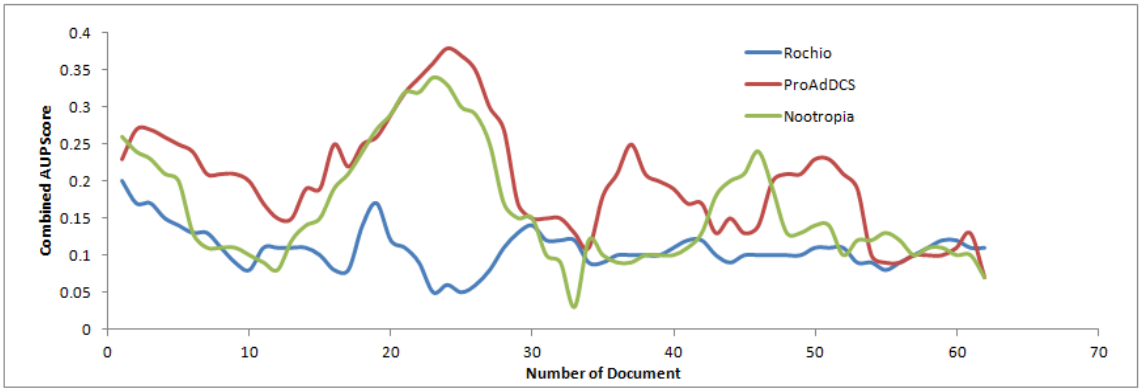
6.6.1 Task α: Parallel Interest of Two Topics

For each of the tasks α, two types of graphs are generated. The first graph represents the evaluation function of AUP score for topics studied. Note that topic of interest is referring to notation R_i. While, the second graph shows the distribution of documents per topic in the training set. Figure 6.3 to Figure 6.5 shows the experiment results for the α tasks, which test the ability of profiles to learn two topics of interest simultaneously. The values on the y-axis count the number of document per topic within each 5 document interval, between subsequent profile evaluations. Since this task does not simulate a radical changes of interest, it allows us to concentrate on how the profile responds to such short-term variations in the feedback stream.

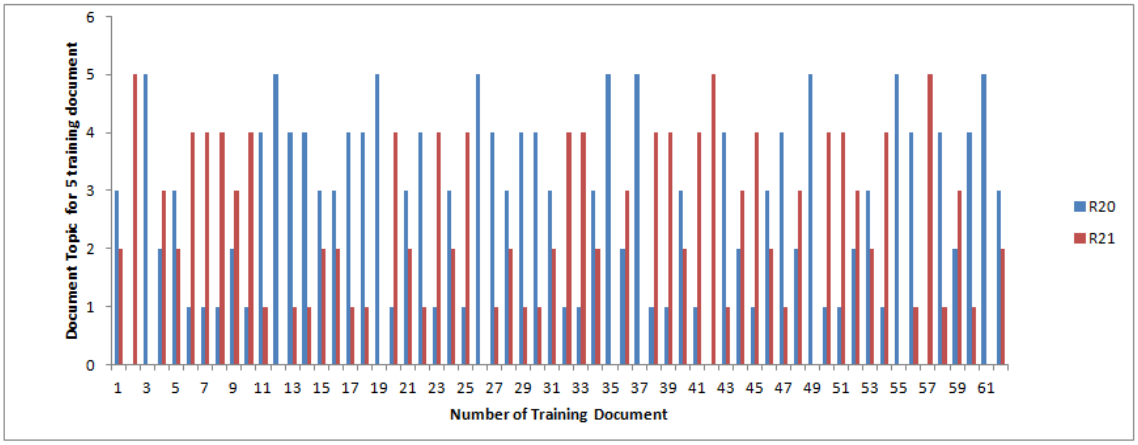
In this experiment, tasks are divided into 3 types of condition, as follows:

- Task $\alpha.1$: Both topics are related and are learned in parallel
- Task $\alpha.2$: Both topics are not related and are not learned in parallel
- Task $\alpha.3$: Both topics are not related and are learned in parallel

The task is organised as above because we want to test the algorithm's performance in terms of continuous learning given different types of situations on learning topics. Our interest is to see the algorithm's behavior when it needs to learn in parallel the unrelated topics. To summarise, the results on task $\alpha.1$ do not show a progressive increase in the score of the two topics being learned. Such a behavior is only clear for task $\alpha.3$, which comprise relatively unrelated topics which are learned in parallel. The distribution of the documents for $\alpha.3$ is homogeneous and this is the case where the profile appears to learn both unrelated topics in parallel. For task $\alpha.1$, which comprises related topics with a large number of documents in the test which are learned in parallel, it appears that the relevant documents extracted from the distributed documents are sufficient for increased performance.



a: AUP Score for Topic R20/R21



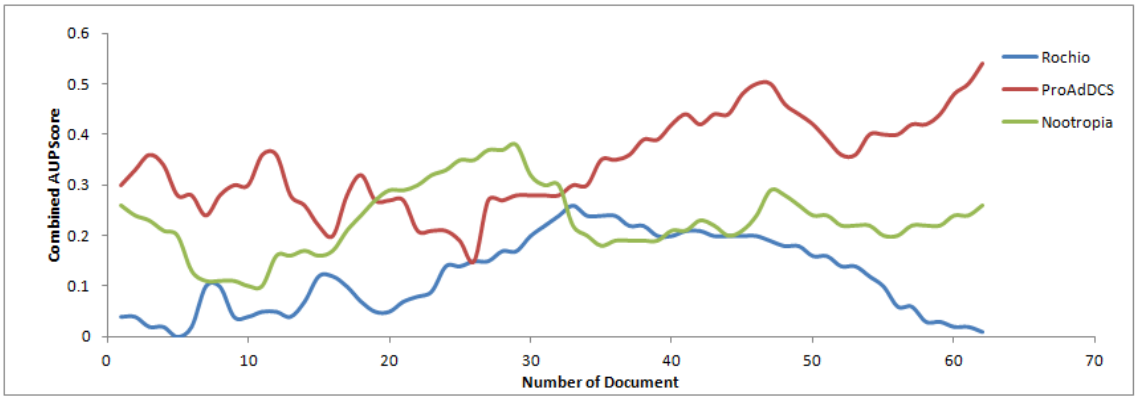
b: Training Document Distribution per Topic

Figure 6.3: Result for Task $\alpha.1$: Both topic are related and are learned in parallel

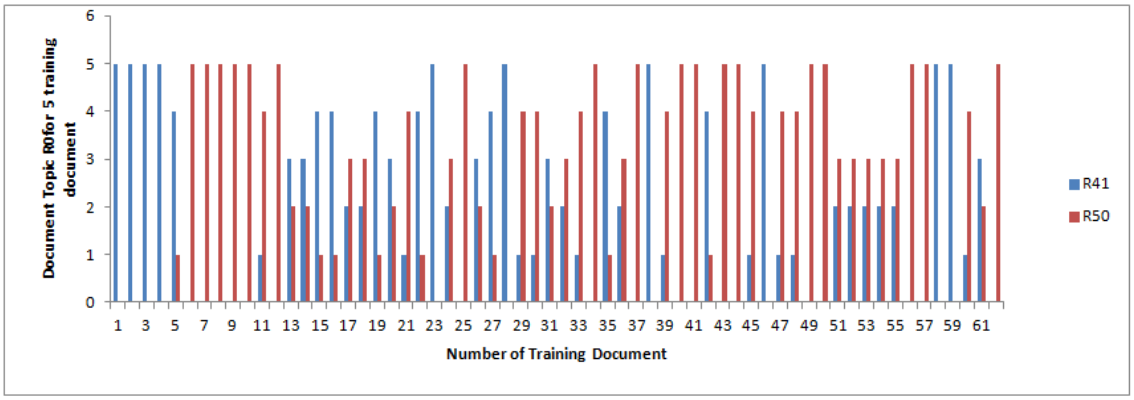
In the case of task $\alpha.2$ as depicted in Figure 6.4, the two topics are not related and are not learned in parallel. In this task, topic R41 is learned first, followed by topic R50. In this task, for the most part of the training period, only one of the topics (R41) is learned and the score for topic R50 increases only towards the end of the training period. In the task of unrelated topic which are not learned in parallel when more feedback documents about a certain topic are processed, the AUP score increases, while the score of the less exciting topic drops. As a result, a topic may be quickly forgotten in absence of feedback documents.

In the extreme case like $\alpha.3$, the training set is initially dominated by documents about topic R68 causing an increase in its score. For the same period, topic R10 is not learned. For a period of more than 20 documents all training documents are about topic R10 and its score increases substantially, while the score for topic R68 drops. This behavior can be seen in Figure 6.5 in graph (b).

The summary so far addressed the issue of measuring the profile's filtering



a: AUP Score for Topic R41/R50

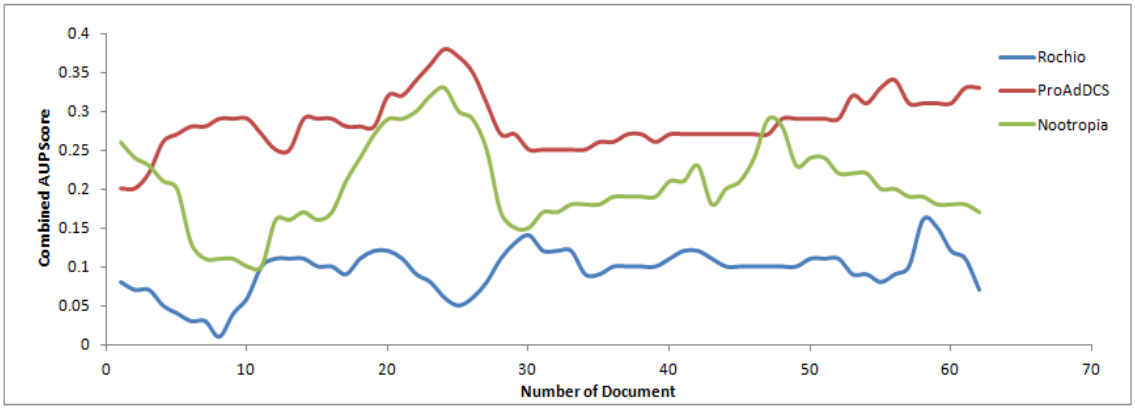


b: Training Document Distribution per Topic

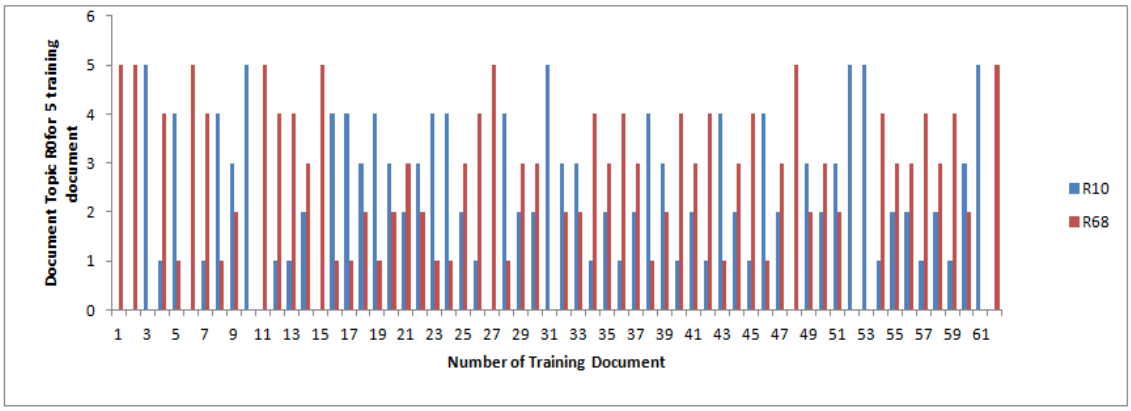
Figure 6.4: Result for Task $\alpha.2$: Both topic are not related and are not learned in parallel

performance, when the profile is trained online with documents about two different topics. Some of the profile shows structural changes and some of the profile does not show structural changes that cause the observation fluctuate in performance. Another interesting issue is to investigate the ability of the comparative approach in adapting profile given different kind of task.

As depicted in Figure 6.3 the tasks which comprises a related topics (the $\alpha.1$ case), the Rocchio's algorithm and Nootropia's show a significant performance where the results do not show a progressive behavior in terms of AUP score for both topics learned. However, the ProAdDCS and the Nootropia system provide a higher combined AUP score when compared to Rocchio's learning algorithm. A progressive behavior can be seen for task $\alpha.3$, where as depicted in Figure 6.5, by maintaining a sufficient diversity in the gene library through transformation of WordNet, the ProAdDCS profile is able to maintain adaptivity of the profile when not related topics are learned in parallel. The situation is different for the



a: AUP Score for Topic R10/R68



b: Training Document Distribution per Topic

Figure 6.5: Result for Task $\alpha 3$: Both topic are not related and are learned in parallel

Nootropia system whereby as the number of documents increased the score for combined AUP is decreased in some part of the graph, but the same also happened for ProAdDCS, although in smaller regions of the graph. One possible explanation is that the generated hierarchies may be relatively shallow and therefore the trained documents do not provide enough informative terms. Finally, for task $\alpha.2$ where the topics are not related and are not learned in parallel, the ProAdDCS profile is able to adapt the profile compared to the Rocchio's learning algorithm and the Nootropia profile model. Although at the beginning of the experiment, with small number of documents, the Nootropia exhibits a higher AUP score compared to the ProAdDCS profile. However, with the capability of continuous learning in ProAdDCS, the profile adaptation is progressively increasing the AUP score. From the experiment, this suggested that the ProAdDCS profile responds to variation in the distribution of feedback documents where its able to adapt, according to our assumption, to frequent changes in a virtual user's

short-term needs.

6.6.2 Task β: New Topic of Interest Emerges

In the β task, our focus shifts from variations in a virtual user's short-term needs to radical changes, the emergence of a new topic of interest. We test the ability of profiles to respond to the introduction of documents about a new topic in the feedback stream. In other words, we test the ability of profiles to learn a new topic of interest. In the subsequent task (task β), we do not include graphs showing the distribution of training documents per topic during the training period. However, separate lines are drawn for each of the document evaluations. The separate lines have two types; the first line presents the average for combined AUP score for the initial two topics and is denoted as a solid line, while the second line presents the AUP score of the new topic with dashed lines. We have chosen to present the average score of the first two topics for visualisation reason and also to be able to concentrate on the new topic that has to be learned. In this experiment the same comparative approach will be used to benchmark the performance for our proposed approach.

Like experiments in task α, the experiments in task β are also divided into 3 types of conditions as follows:

- Task $\beta.1$: Both topics are related and are learned in parallel
- Task $\beta.2$: Both topics are not related and are not learned in parallel
- Task $\beta.3$: Both topics are not related and are learned in parallel

Figure 6.6 to Figure 6.8 present for each β task, the average for combined AUP score compared with the comparative approaches. In general, the trends shown in task β are clearly different form those in the α task, but a common pattern can be again identified. In task $\beta.1$ (in Figure 6.6), Nootropia adapts to new topic better than (higher AUP topic score) ProAdDCS. This may be due to the reason which in this case the profiles already contain terms related to the new topic, due to the semantic proximity between the latter and the initial two topics (refer to Appendices H). They already represent aspects of the topics in the profile. As a consequence, the results for this task show that it is difficult for related topics to distinguish themselves from other topics in the profile.

Nevertheless, for tasks $\beta.2$ and $\beta.3$, which include more unrelated topics, the results reveal the performance of ProAdDCS in $\beta.2$ (Figure 6.7) seems worse than performance of Nootropia in $\beta.1$ (Figure 6.6). Although the Nootropia shows the

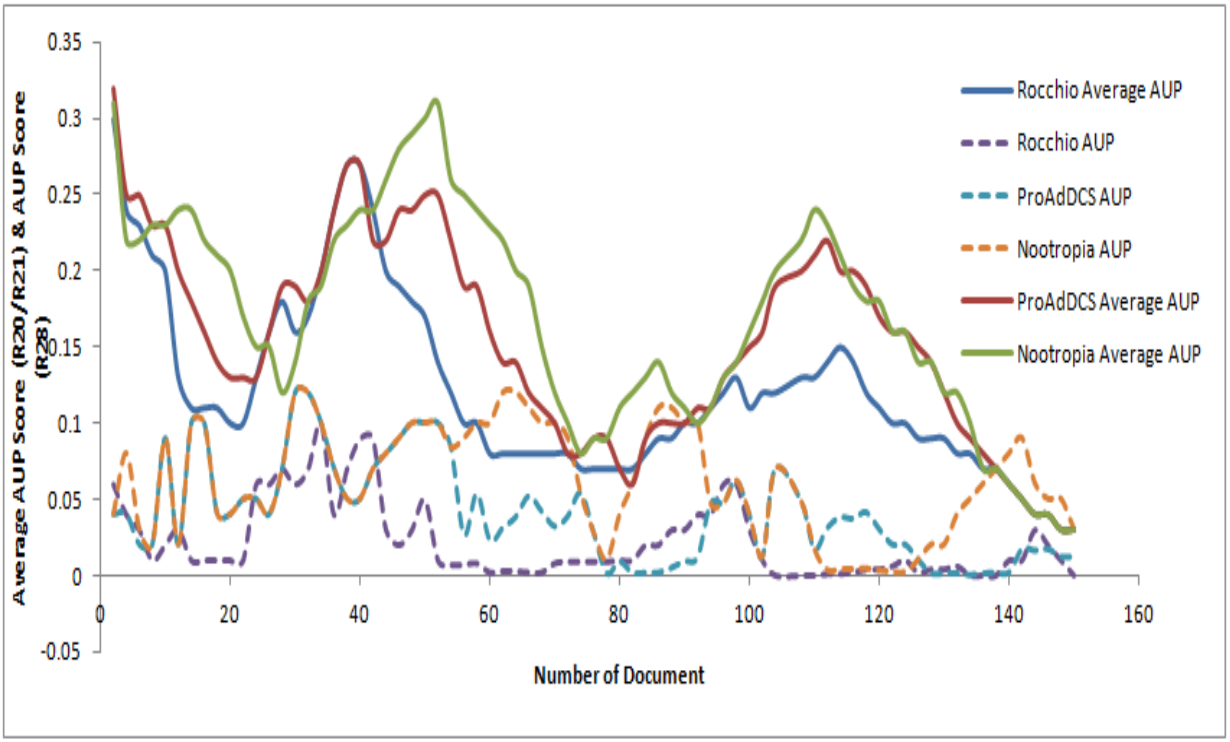


Figure 6.6: Result for Task $\beta 1$: Both topic are related and are learned in parallel

profile's ability to learn new topic of interest, however, the average AUP score is smaller compared to the ProAdDCS in most parts of the graph in Figure 6.7 and Figure 6.8. For task $\beta.2$ in particular, for ProAdDCS's profile, there is a further drop in average AUP score towards the end of the training phase. One particular reason might be a sufficient vocabulary of terms has been assembled in the profiles and so the average AUP score does not increase further.

To summarise, eventually, the profile acquires a sufficient vocabulary of informative terms or, in other words, to store more information about the emerging topic. With the preservation of diversity (heterostasis) in our approach, the adaptive user profile can be achieved by detecting changes in the interest dynamically. Furthermore, through the introduction of diversity (heterogenesis) in the the gene libraries with synset relationship based on WordNet, the profile is further able to learn a new topic of interest.

6.6.3 Task γ: Forgetting Topic

In task γ, we test the ability of profiles to forget one of three topics of interest. For each γ task, a profile is initially trained wth documents about three topics and subsequently with documents about only two of the topics. As before, for each γ

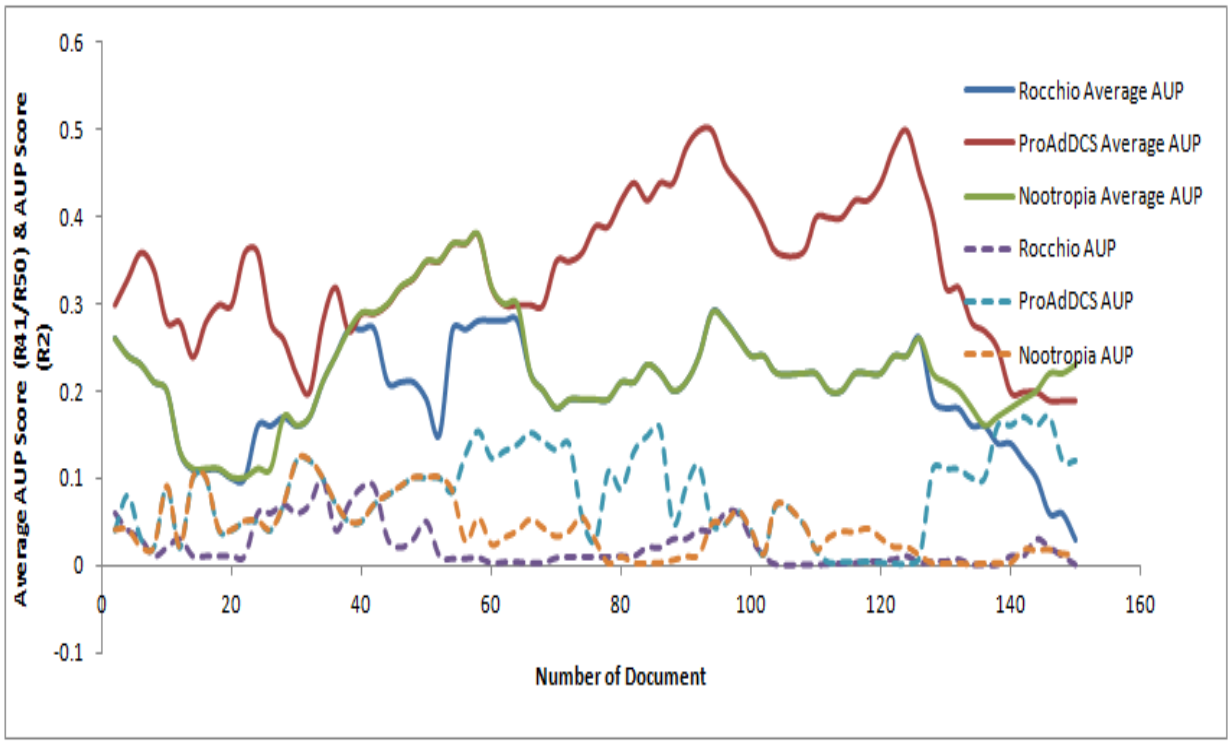


Figure 6.7: Result for Task $\beta 2$: Both topic are not related and are not learned in parallel

task, the graph is presented with two types of lines; the average of combined AUP score for the two topics of "long-lasting" interest (which are not to be forgotten) with solid line and the AUP score of the third, unexciting topic, with dashed lines. In this experiment the same comparative approach will be used to benchmark the performance for our proposed approach. Like in the previous experiment, tasks for γ are also divided into 3 types of conditions, as follows:

- Task $\gamma.1$: Both topics are related and are learned in parallel
- Task $\gamma.2$: Both topics are not related and are not learned in parallel
- Task $\gamma.3$: Both topics are not related and are learned in parallel

Figure 6.9 to Figure 6.11 present for each γ task, the average combined AUP score compared with the comparative approaches. As with task $\beta.1$, the results for task $\gamma.1$ do not show any significant difference in the profile's performance among all the algorithms. Although the topic to be forgotten was not effectively learned in the first place for all the tested approaches, however, the score for the third unexciting topic initially increased a little for the ProAdDCS's and the Nootropia's profile.

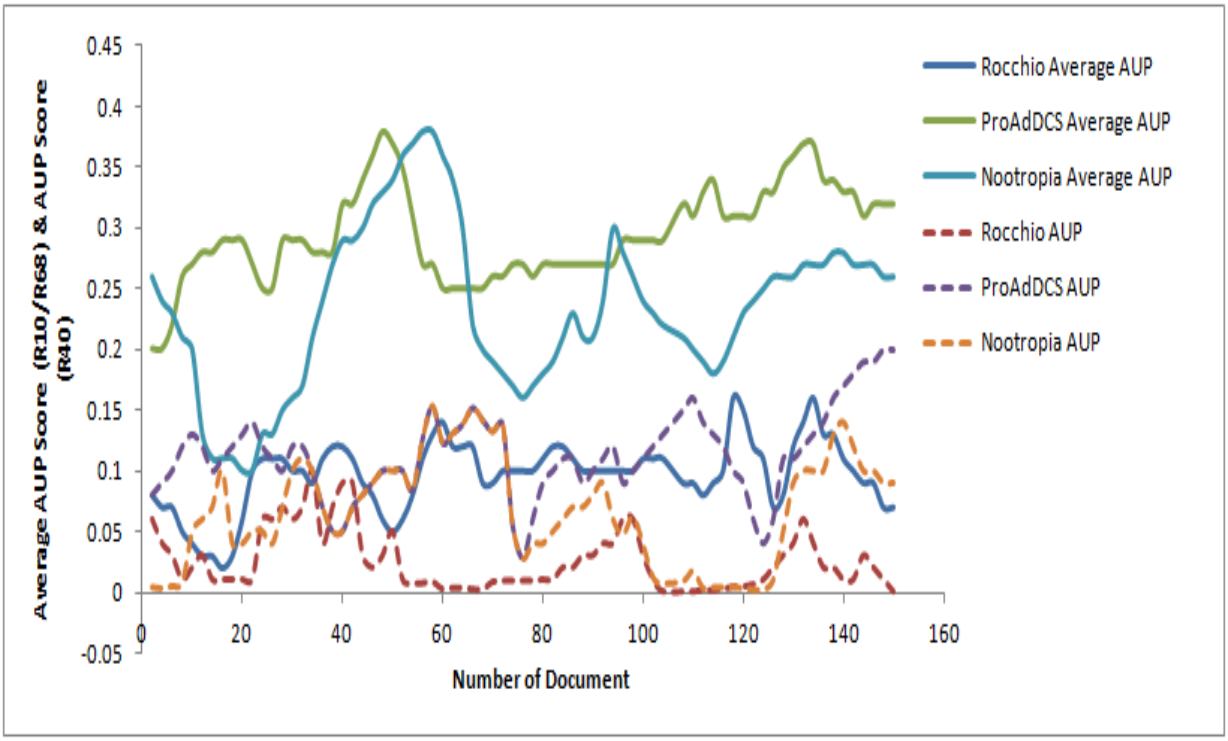


Figure 6.8: Result for Task $\beta 3$: Both topic are not related and are learned in parallel

For tasks $\gamma.2$ and $\gamma.3$, which comprise of unrelated topics, the results reveal the ProAdDCS's profile and Nootropia's profile ability to forget the unexciting topic. For the Rocchio's and Nootropia, the profile's ability to forget unexciting topic is lower with the smaller average AUP score. However, in the case of forgetting, a lower AUP score is not bad, since the goal is to minimise the score of the forgotten topic. For task $\gamma.2$ in particular, the third unexciting topic (R2) for ProAdDCS's profile shows a zero score after 120 documents which indicates that the topic is completely forgotten with the number of document increases. For task $\gamma.3$ (Figure 6.11), Nootropia was much more successful than ProAdDCS in forgetting the third topic.

To summarise, the experiments have shown a positive result where ProAdDCS profile clearly indicate changes in the profile's performance in response to the radical changes of interest. A profile representing more than one topic of interest may forget a topic that, in contrast to the rest of the topics, no longer receives positive feedback. The results also have shown a significant difference for tasks comprising unrelated topics when learned in parallel or not.

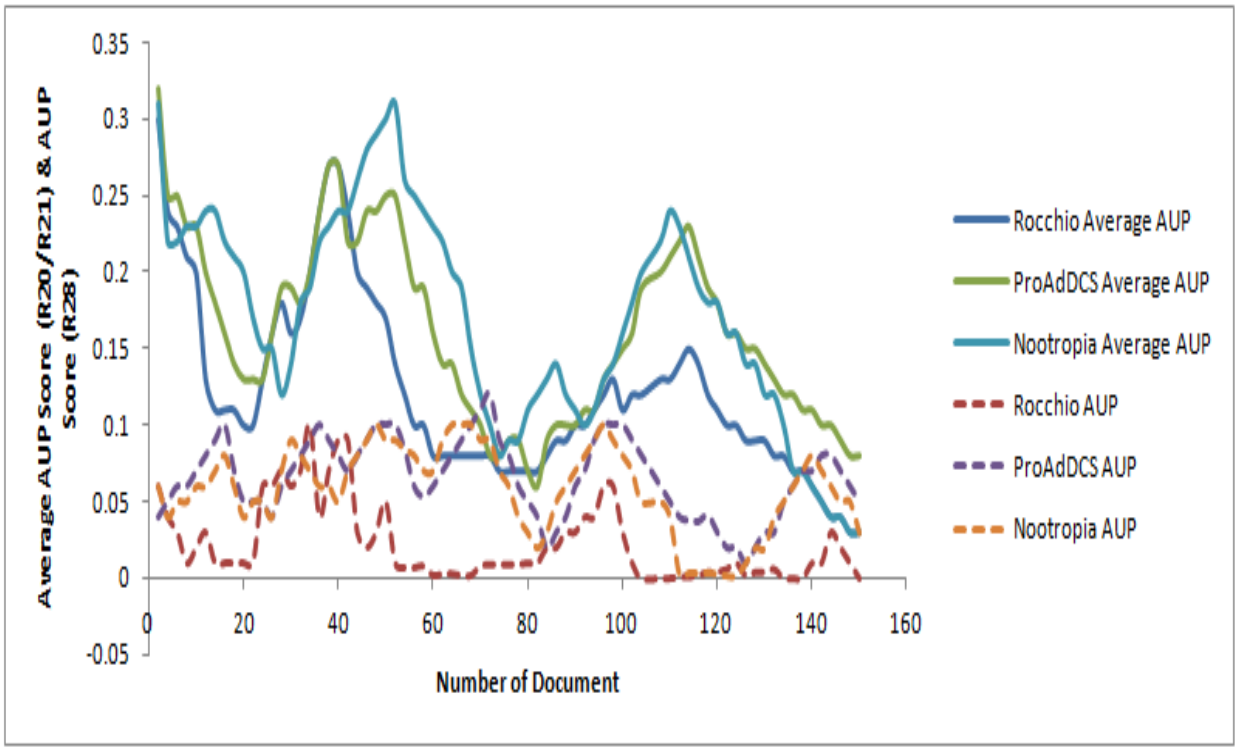


Figure 6.9: Result for Task $\gamma.1$: All topics are related and are learned in parallel

6.7 Experimenting with ProAdDCS Profile in Adapting Learning and Forgetting Task. Case Study: Reuters-21578 Document Collections

The experiments so far have investigated the ProAdDCS profile tested on TREC-2001 filtering track and it has shown positive results. Next, the experiment is investigated further based on Reuters-21578 Document Collections[5]. The data was originally collected and labeled by Carnegie Group, Inc. and Reuters, Ltd. and include 21578 news stories that appeared in Reuters newswire in 1987. The documents have been manually classified according to 135 topic categories, but for this experiments we concentrate only on the 23 topics with at least 100 relevant documents. Usually in text classification experiments, for each topic category a classifier is first trained using relevant documents from the training set and is subsequently evaluated against the test set. However, for the purpose of this experiment we use the collections in a different way. The documents in Reuters–21578 are ordered according to publication date and their topicality changes accordingly. We exploit this ordering to test the ability of the algorithm to continuously

[5]http://www.daviddlewis.com/resources/testcollections/reuters21578/

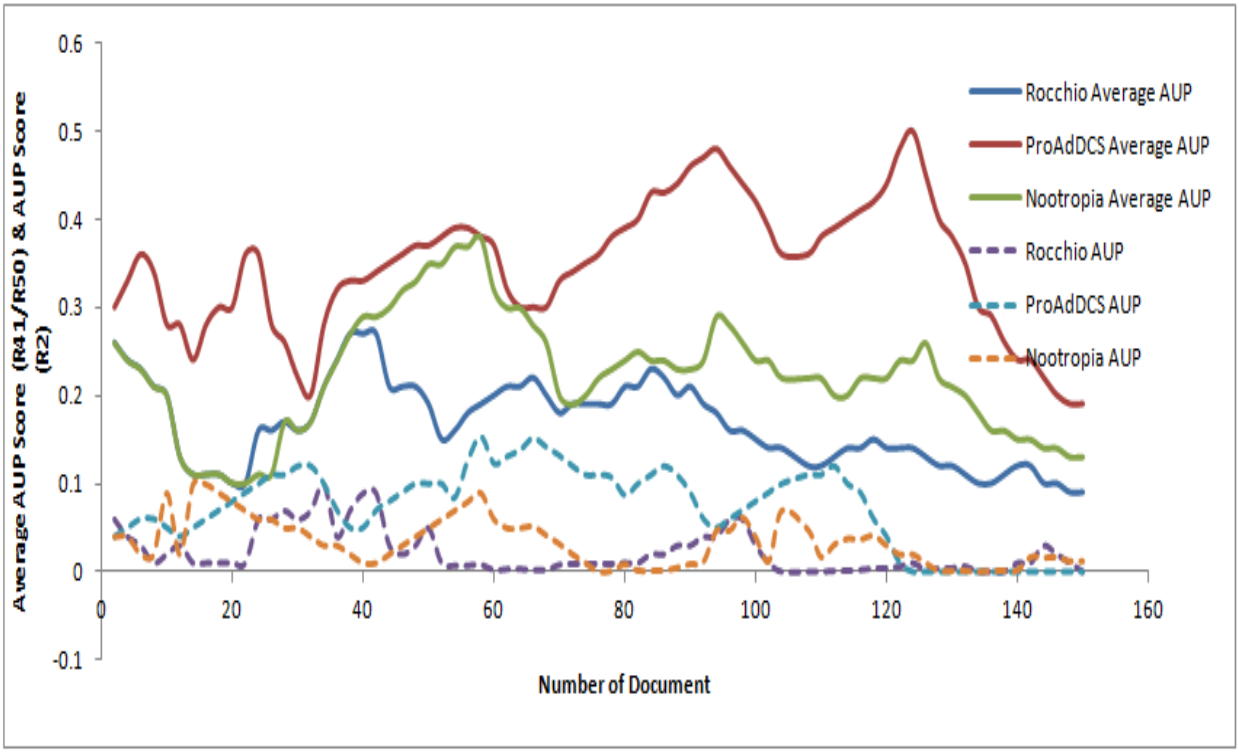


Figure 6.10: Result for Task $\gamma.2$: Both topic are not related and are not learned in parallel

learn, but also to forget. The 23 topics involved in the experiments are ordered according to decreasing size, i.e., the number of documents in the collection that are relevant to a topic. Table 6.3 provides the 23 topics and their corresponding sizes.

Topic	earn	acq	moneyFx	crude	grain	trade
Size	3987	2448	801	634	628	552
Topic	interest	wheat	ship	corn	dir	oilSeed
Size	513	306	305	254	217	192
Topic	moneySupp	sugar	gnp	coffee	vegOil	gold
Size	190	184	163	145	137	135
Topic	natGas	soyBean	bop	livestock	cpi	
Size	130	120	116	114	112	

Table 6.3: Topics Involved in the Experiments and their Corresponding Size

6.7.1 Experimental Methodology

We followed a methodology proposed in [132]. This methodology was proposed for evaluating the ability of a user profile for continuous adaptation in a dy-

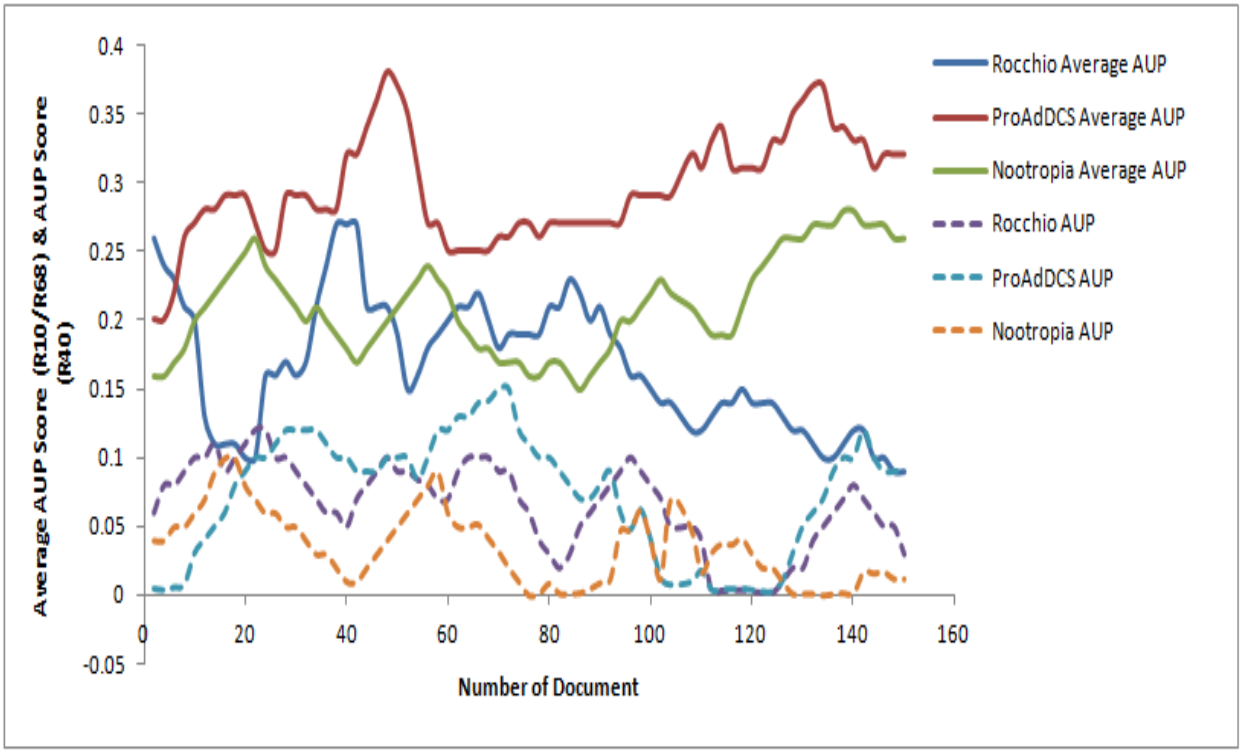


Figure 6.11: Result for Task $\gamma.3$: Both topic are not related and are learned in parallel

namically changing environment. The same methodology has also been used in Section 4.4.1 to evaluate the extended version of AISEC for classification on multiple-topic of email. We adopt this methodology because we believed that this methodology can be used for testing the ability of adaptive systems, such as AIS, for online learning (and forgetting) in a complex, multi-dimensional and dynamic environment. Furthermore, iterating over the same set of documents and suddenly switching between topics of interest causes discontinuities. Nevertheless, how the system reacts to these discontinuities is an interesting test for the system's dynamics.

In this experiment, each experimental run starts with an initially empty profile. The profile then evaluates (i.e., assigns a score to) the 21578 documents in order. The empty profile assigns a zero score to documents until it encounters a document relevant to the first of the 23 topics (i.e., earn) and then it is initialised. The initialised profile assigns a score to the remaining documents in the collection and every time it encounters a document that belongs to topic *earn*, it evaluates the document and then uses it as positive feedback and adapts based on it. When all 21578 documents have been evaluated, they are ranked according to decreasing relevance score and the AUP for each of the 23 topics is calculated on the

ordered list of documents. After the first evaluation period, the process is reinitiated and the profile, which now represents the first of the 23 topics, starts anew to evaluate the document collection. This time, however it uses as positive feedback documents that belong to the second of the 23 topics (i.e., *acq*). In other words, the profile has to forget the no longer interesting first topic (*earn*) and learn the new topic of interest (*acq*). Once all documents have been evaluated, a new set of AUP values is calculated and the process is repeated for the third topic (*moneyfx*). The experiment finishes once all 23 topics have been used as positive feedback.

For multiple-topic experiments, the same process is applied. Multiple-topic experiments include the two-topic and three-topic. In the two-topic experiment, the profile initially learns the first two topics in parallel. The system then forgets the first and learns the second and third topics, and so on. Similarly, in the three-topic scenario, the profile has to be able to represent three topics in parallel, the first triple first, then the second triple and so on.

To evaluate the performance of the algorithm, we carried out a statistical analysis based on the non-parametric Mann-Whitney-Wilcoxon or rank-sum test [125] to test whether two algorithms' performances had different distributions (each having a different median), and the Vargha-Delaney A statistics [126] to measure the effect size between these algorithms' performances. The ProAdDCS profile is compared with the baseline approach mentioned in Section 6.5.1.

6.7.2 Experimental Results and Analysis

In this section, the results of the experiment are discussed. Table 6.4 to Table 6.6 present the complete comparative experiment results for the single-topic, two-topic and three-topic experiments. In particular, Table 6.4 presents the AUP scores achieved for the single-topic for the tested algorithms (column two to column four). The final three columns ("diff(A)% ", "diff(B)% " and "diff(C)% ") present, respectively, the differences in percentage between the ProAdDCS with Rocchio, the ProAdDCS with Nootropia and the Rocchio with the Nootropia. Similarly, Table 6.5 to Table 6.6 (column two to column four) shows the AUP score for the two-topic and three-topic experiments respectively. The combined values of AUP scores for the multiple-topic (including the three-topic) are calculated based on the aggregate set of relevant documents for all constituent topics. Note that the combined value is not the sum of the individual AUP scores. The differences between the ProAdDCS with the baseline approaches were calculated based on these combined AUP values and the percentage values were taken. The statistical analysis for the tested algorithms is presented in the last row of the list

of the topics. The analysis includes the median, standard deviation and the non-parametric statistical analysis based on the Rank-Sum and the Vargha-Delaney A statistics.

Table 6.4 to Table 6.6 present the AUP score of the topic, or the topics, of interest at the end of the corresponding evaluation period. For the single-topic experiments, Table 6.4 presents the topic of interest at each evaluation period, the corresponding AUP values for ProAdDCS, Rocchio's and Nootropia, respectively, and the differences between the performances (AUP scores) for each pair of algorithms.

In the single-topic experiment (Table 6.4), the ProAdDCS profile produces a better AUP score than the baseline approaches for the overall topics tested. On the contrary, for the Nootropia and the Rocchio's profile, the Nootropia's profile performed better for 19 topics out of 23 topics. However, as already discussed, representing a single topic of interest is a relatively simple problem and does not accurately reflect a real situation. In reality, a user is typically interested in more than one topic in parallel. In contrast, in terms of statistical non-parametric analysis, the p-value indicates with 95% confidence that there was a statistically significant difference between the ProAdDCS's profile with the Rocchio's algorithm and Nootropia. In summary, these samples had a different distribution which indicates that their medians were different. The A value from the Vargha-Delaney A statistics shows that the ProAdDCS's profile with the tested algorithms showed a large effect with the A value above 0.71[6]. This indicates that the performance difference between the ProAdDCS's profile with Rocchio's and Nootropia shows a large effect when tested on the single-topic of Reuters 21578 document collections.

The results for the two-topic (Table 6.5) and the three-topic (Table 6.6) clearly show that as the complexity of the user's interests increases, the preservation of diversity become increasingly important. As can be seen in Table 6.5, the ProAdDCS's profile performs better compared with the Rocchio's and Nootropia's for overall topics combined. Similarly, when compared with Nootropia's profile, the proposed approach performed better in 19 out of 22 topics combined. The Nootropia's profile performed better in 20 out of 22 topics combined compared with the Rocchio's profile. Moreover, in all cases, the Mann-Whitney-Wilcoxon or rank-sum test showed that the differences between pairs of algorithms were statistically significant. This can be summarized as showing that the tested algorithms had different AUP score distributions and there was a large performance effect with the A value above 0.71 when compared with Rocchio's profile. How-

[6]for a complete list of A value and its description, see Table 4.3 on Section 4.3.1

ever, when compared with Nootropia's profile, the ProAdDCS profile shows a medium effect size with the A value 0.634. For the three-topic experiment, as can be seen in Table 6.6 the difference was even larger. The ProAdDCS's profile performed better in 21 topics out of 22 topics when compared with Nootropia, while Nootropia's profile performed better in 20 topics out of 22 topics compared with Rocchio's. This means that ProAdDCS's adaptation, which involves the introduction and preservation of diversity in the the gene libraries is better than the Nootropia's profile as well as Rocchio's profile. This indicated that the ProAdDCS profile is able to respond to short-term variations and occasional radical changes in the composition of a stream of feedback documents.

6.8 Summary

This chapter is concerned with a study into experimenting with AIS towards profile adaptation to changes on user interests in adaptive document filtering. We have argued that the user interests are by nature dynamics where a combination of parameters causes a variety of changes. Frequent changes in the user's short-term needs contribute to progressive changes in the user's long term interests and vice versa. The user's interest may shift frequently between different topics or related subtopics. New topics and subtopics of interest emerge and the interest in a certain topic might be lost. To achieve adaptation of our single or multiple-topic profile to a variety of changes in the user's interests, we have been inspired by biological theories of dynamic clonal selection (DCS). DCS can inherently maintain and boost diversity and can dynamically control the size of the immune repertoire by means of selection, cloning, and mutation procedures. Moreover, diversity in the population is enabled by means of the receptor editing process. In this chapter, we suggested that profile adaptation can be developed by incorporating ideas from aspects of DCS with gene libraries to maintain sufficient diversity through transformation of synset relationship based on WordNet. DCS is a variation of clonal selection and it has been identified as an AIS algorithm that supports learning in dynamically changing environments [91, 115–118]. We have described in the chapter the theoretical foundation of DCS, which also includes the identified potential of DCS towards adapting a user profile. The algorithm and the process of DCS known as profile adaptation through dynamic clonal selection (ProAdDCS), have also been explained in this chapter.

Despite the challenges in a research of profile adaptation, there is an evident tendency in the literature to study the adaptive user profile, such as the Rocchio's learning algorithm and the self-organising Nootropia's profile model. These ap-

proach were selected as baseline for our proposed approach. Comparative performance is important in order to assess whether the works are incremental improvements on the state of the art or evolutions of existing work. The profiles are tested for their ability to adapt over time in the content of documents. To test our approach to adaptive document filtering, we have synthesised virtual users based on web document corpus, namely the TREC-2001 filtering track and Reuters-21578 document collection. We have explained the justification for using the simulated user over the real user experiment. We made the assumption that a user's interest and changes in them are reflected by the feedback that the user provides. On these grounds, we have carried an experiment to test the ability of the profile in learning and forgetting topics based on the identified corpus. We may argue that the experiment results have been positive. The experiments indicate that the ProAdDCS profile responds to short-term variations and occasional radical changes in the composition of a stream of feedback documents. Furthermore, the result on ProAdDCS's profile have also been positive for the tasks comprising the unrelated topics learned in parallel as well as not learned in parallel. As a result, the following adaptive behaviors are observed:

1. More than one topic of interest may be learned from scratch and in parallel with a single profile.

2. The relative importance of topics in the profile varies in response to short-term variations in the distribution of relevant documents in the training set.

3. An existing profile representing more than one topic of interest may learn an emerging topic of interest, without what is already represented being significantly affected.

4. A profile representing more than one topic of interest may forget a topic that, in contrast to the rest of the topics, no longer receives positive feedback.

In conclusion, if our assumption is true, then we may argue that adaptation to both variations in a user's short-term needs and radical changes in long-term interests has been achieved with a single, multiple-topic profile, through a process that exhibits characteristics of DCS. With the preservation of diversity (heterostasis) in our approach, the adaptive user profile can be achieved and infer changes in the user's interest dynamically. Furthermore, through the introduction of diversity (heterogenesis) in the the gene libraries with synset relationship based on WordNet, the system is further able to learn a new topic of interest. The profile

appears to be able to adapt to a variety of simulated changes in a virtual user's interest.

Table 6.4: Results for single-topic experiments: Topic(first col.), AUP Score (two to fourth col.), differences between ProAdDCS with Rocchio (fifth col.), ProAdDCS with Nootropia (sixth col.) and Rocchio with Nootropia (seventh col.).

Topics	AUP Score			Diff.(A)%	Diff.(B)%	Diff.(C)%
	ProAdDCS	Rocchio	Nootropia			
earn	0.731	0.511	0.758	12.542	0.318	4.224
acq	0.700	0.638	0.357	4.367	28.181	6.186
moneyFx	0.647	0.517	0.614	23.247	18.243	5.004
crude	0.729	0.403	0.515	21.481	18.835	2.647
grain	0.695	0.569	0.510	18.535	5.890	12.645
trade	0.745	0.695	0.520	22.444	17.426	5.018
interest	0.714	0.688	0.570	14.368	11.746	2.622
wheat	0.737	0.437	0.571	9.955	6.398	3.443
ship	0.763	0.501	0.582	5.884	4.618	0.266
corn	0.850	0.545	0.557	12.317	8.869	0.448
dlr	0.800	0.516	0.761	13.940	1.547	8.393
oilSeed	0.802	0.509	0.564	14.786	14.523	0.263
moneySupp	0.799	0.412	0.533	19.682	2.910	1.772
sugar	0.731	0.485	0.591	14.601	14.945	0.344
gnp	0.696	0.487	0.554	10.220	7.288	1.931
coffee	0.810	0.346	0.568	20.199	17.832	6.366
vegOil	0.841	0.307	0.527	21.419	17.971	3.448
gold	0.755	0.436	0.590	16.511	14.644	1.867
natGas	0.817	0.456	0.591	22.554	16.450	6.104
soyBean	0.705	0.444	0.593	11.206	5.079	6.127
bop	0.721	0.376	0.601	17.008	5.582	4.426
livestock	0.739	0.334	0.704	18.489	0.930	4.558
cpi	0.696	0.387	0.554	14.220	13.288	0.931
median	0.737	0.591	0.704			
std. dev.	0.054	0.035	0.047			
			rank sum	3341.000	3781.000	3130.000
			Z-val	5.370	2.4375	7.318
			p-value	6.341E-11	1.9103E-13	2.363E-06
			A value	0.821	0.738	0.627

Table 6.5: Results for two-topics experiment: Topic(1st col.), AUP Score (two to fourth col.), differences between ProAdDCS with Rocchio (fifth col.), ProAdDCS with Nootropia (sixth col.) and Nootropia with Rocchio (seventh col.).

Topics	AUP Score			Diff.(A)%	Diff.(B)%	Diff.(C)%
	ProAdDCS	Rocchio	Nootropia			
earn:acq	0.884	0.511	0.658	9.522	5.398	0.224
acq:moneyFx	0.798	0.438	0.557	20.367	8.181	0.186
moneyFx:crude	0.727	0.517	0.584	13.237	11.293	0.034
crude:grain	0.619	0.303	0.665	21.481	1.835	10.647
grain:trade	0.695	0.569	0.510	8.535	5.890	0.652
trade:interest	0.725	0.595	0.720	10.222	0.121	5.018
interest:wheat	0.767	0.534	0.604	11.319	2.377	3.372
wheat:ship	0.772	0.331	0.387	19.955	17.334	1.433
ship:corn	0.733	0.401	0.555	10.184	4.638	2.266
corn:dlr	0.785	0.545	0.857	7.137	2.163	13.238
dlr:oilSeed	0.810	0.316	0.561	20.940	9.558	1.386
oilSeed:moneySupp	0.602	0.309	0.564	23.786	14.523	9.263
moneySupp:sugar	0.799	0.412	0.533	16.692	7.910	2.742
sugar:gnp	0.731	0.414	0.585	16.501	10.442	1.334
gnp:coffee	0.714	0.498	0.570	14.368	7.744	4.622
coffee:vegOil	0.810	0.396	0.568	24.149	7.736	2.346
vegOil:gold	0.741	0.427	0.527	15.419	4.971	0.425
gold:natGas	0.755	0.416	0.590	16.541	11.564	1.867
natGas:soyBean	0.817	0.316	0.591	22.554	6.450	0.604
SoyBean:bop	0.795	0.444	0.593	11.206	5.079	1.127
bop:livestock	0.711	0.436	0.601	12.548	1.582	2.446
livestock:cpi	0.621	0.375	0.641	10.008	1.582	9.436
median	0.755	0.531	0.701			
std. dev.	0.044	0.026	0.039			
			rank sum	3131.000	3553.000	3032.000
			Z-val	9.342	7.645	4.738
			p-value	5.721E-11	2.113E-09	1.321-04
			A value	0.812	0.634	0.737

Table 6.6: Results for three-topic experiment: Topic(1st col.), AUP Score (two to fourth col.), differences between ProAdDCS with Rocchio (fifth col.), ProAdDCS with Nootropia (sixth col.) and Nootropia with Rocchio (seventh col.).

Topics	AUP Score			Diff.(A)%	Diff.(B)%	Diff.(C)%
	ProAdDCS	Rocchio	Nootropia			
earn:acq:moneyFx	0.786	0.511	0.658	9.522	5.398	0.224
acq:moneyFx:crude	0.786	0.438	0.557	20.367	8.181	0.186
moneyFx:crude:grain	0.673	0.497	0.514	13.237	11.293	1.004
crude:grain:trade	0.692	0.303	0.515	21.481	4.835	2.647
grain:trade:interest	0.815	0.569	0.510	18.547	15.560	0.852
trade:interest:wheat	0.725	0.695	0.730	3.252	0.343	0.425
interest:wheat:ship	0.767	0.534	0.604	11.319	2.377	3.372
wheat:ship:corn	0.772	0.371	0.437	19.955	17.334	2.443
ship:corn:dlr	0.733	0.401	0.555	10.184	4.638	2.266
corn:dlr:oilSeed	0.785	0.345	0.557	19.537	8.863	3.438
dlr:oilSeed:moneySupp	0.810	0.316	0.561	20.940	9.558	1.386
oilSeed:moneySupp:sugar	0.841	0.307	0.527	21.419	17.971	3.448
moneySupp:sugar:gnp	0.799	0.412	0.533	16.692	7.910	2.742
sugar:gnp:coffee	0.731	0.414	0.585	16.501	10.442	1.334
gnp:coffee:vegOil	0.714	0.498	0.570	14.368	7.744	4.622
coffee:vegOil:gold	0.810	0.396	0.568	24.149	7.736	2.346
vegOil:gold:natGas	0.741	0.427	0.527	15.419	4.971	0.425
gold:natGas:soyBean	0.755	0.416	0.590	16.541	11.564	1.867
natGas:soyBean:bop	0.847	0.316	0.591	19.535	9.320	3.434
SoyBean:bop:livestock	0.795	0.444	0.593	11.206	5.079	1.127
bop:livestock:cpi	0.817	0.456	0.691	22.554	16.450	6.104
median	0.815	0.611	0.701			
std. dev.	0.0491	0.016	0.025			
			rank sum	3561.000	3671.000	3319.000
			Z-val	6.335	5.663	4.756
			p-value	5.718E-09	7.819E-10	2.173E-02
			A value	0.827	0.781	0.727

CHAPTER

SEVEN

CONCLUSION

This final chapter concludes the thesis, summarises its contribution, and makes suggestions for potential future work. In Section 7.1, the contribution of this study is presented and discussed. In Section 7.2, a reflection on the limitations of the current work is offered and the direction of future work is recommended. Finally, concluding remarks on the work presented in this thesis are given in Section 7.3.

7.1 Thesis Contribution

The aim of this thesis is to investigate the application of AIS to profile adaptation in adaptive document filtering, particularly in the domain of information filtering. To the best of our knowledge, there have been very few researchers working on the discovery of adaptive user profiles with AIS and therefore there is greater potential for valuable research in this area. The research topic identified and investigated by this current study is therefore somewhat unexplored in itself. This section summarises the contributions which this thesis can make.

The Proposal of Principled Abstraction based on Meta-probes for Immune-Inspired Adaptive Information Filtering

The featured properties of adaptability in user profile in a context of changing interest (either of a particular person or a group of persons with a shared interest) and the heterogeneous nature of the information in an incoming data stream are characteristics which are likely to need some form of adaptive information

filtering (AIF) system. The viability of an adaptive user profile for an AIF system relies on the ability of the profile to maintain a satisfactory adaptation. The user profile has to be able to represent the complete range of a user's interests and to continuously adapt, in response to user feedback, to any changes in those interests. As the user's interests and the information environment change, new terms are required to cover new topics of interest, and it becomes at least impractical to maintain in the profile terms that no longer reflect the user's interest. In this work, we were inspired by the biological immune system to build a user profile that can continuously maintain a representation of the developing user interests within a changing information environment. In contrast, AIS has the inherent ability to boost and maintain the diversity of the immune repertoire achieved through the preservation of diversity (heterostasis) and the introduction of diversity (heterogenesis). To further identify characteristics or properties inherent in the biological system, that is, the immune system and the application domain, we followed the principled meta-probes suggested in [5]. We followed the questions that address notions such as Openness, Diversity, Interaction, Structure and Scale, otherwise known as the ODISS meta-probes. The ODISS meta-probes were used to challenge the biological system (as in [5,83]) and the application domain, and to identify matching characteristics and ODISS properties. The results of this high-level abstraction were used as part of the basis for building a biologically-inspired application here, an immune-inspired AIF [1,147].

Applying the ODISS meta-probe analysis to the immune system is not only challenging, but will most certainly be incomplete. Therefore, we merely sought to demonstrate the principled characterization of the immune system and to shed some light on how the immune system can be considered. For the application domain, we applied the ODISS approach to the AIF domain to highlight the features that would be necessary for an ideal AIF system. From these principled abstractions, we can summarize that both AIF and the immune systems are open, and, whilst an AIF system is arguably less open (a software system has less scope for receiving entirely novel inputs or generating entirely novel responses), the AIF shares the need for continual evolution to produce effective adaptation to small changes in a very large range of inputs. Again, the AIF and the immune system have common characteristics of diversity, and we can take inspiration from immune system properties such as degeneracy and pleiotropism in finding effective ways for filtering and learning to adapt to changes in inputs. To summarise, one of the possible advantages of this principled approach is that the ODISS view may lead to a comprehensive set of requirements which would allow us to identify an appropriate property of the immune system and the problem domain studied.

In fact, a study of the application domain led to the search for solutions with derivation of the biological components that are oriented to the problem studied. Therefore, it may help to clearly consider the choice of representation, the affinity measure and the biological algorithm related to the application domain studied. These have been emphasized by Freitas and Timmis [100], who outlined the need to consider carefully the specific characteristics of the application domain when developing a bio-inspired application.

Implementation of Multiple-Topic Classification for an Email Classifier

Email has become an efficient and popular communication mechanism as the number of internet users has increased. The application of AIS in the email classification domain has been implemented for some years using the dynamic nature of the immune system. E-mail classification was chosen for this current study because of its characteristic properties of the dynamics and diversity in users' interests in e-mails. In the e-mail environment, the topics a user may be interested in are liable to drift over time. Thus, the ability of an algorithm to keep track of changes in the application domain is very important in such a filter. The problem of email classification or filtering is not a new one and there are already a dozen different approaches to the problem that have been implemented. Several implementations have had various trade-offs, different performance metrics, and different classification efficiencies. Most of the studies on email classification have been treated under spam filtering, with the exception of a study by Secker et. al [9] who developed an AIS algorithm for email classification (AISEC) which classifies emails as 'interesting' or 'uninteresting'. In the previous AISEC [9, 10] the algorithm extracted words only from the email subject and sender fields. This limits the potential diversity of the *gene library* (the set of all words from feature vectors of B cells that recognise uninteresting emails). This in turn reduces the diversity generated by cloning and mutation, since, when mutation is performed, a word from this library replaces a word from a cell's feature vector. In order to recognise new topics of interest and enable the removal of existing topics of interest, we need a large library of words, and we need to be able to detect synonyms. There are several ways to increase the diversity of words in the gene library. Our first modification to the AISEC was to consider the body of the email in addition to the email subject and sender field as this provides a richer set of words. In addition, we used the WordNet corpus as a source of synonyms. This allowed more accurate classification of emails, and improved the capability to identify new and potentially uninteresting e-mails. Furthermore, to show that the extended AISEC

is capable of continuous learning, and of potentially tracking changes in email topic, we carried out an experiment to verify whether explicit changes in a user's interests could be tracked. The experiment was conducted in two scenarios; binary classification (discriminating between interesting or not interesting email) and multiple-topic classification.

However, to the best of our knowledge, none of the work on email classification has focused on classifying the emails based on multiple email topics. In the real-world situation, users are typically interested in more than one topic in parallel, and both their interests and the information environment change over time. Therefore, users may read one or more emails according to their interest(s). To more accurately simulate a real situation, we devised and performed an experiment based on an extended version of the AISEC system in which each user profile has to represent more than one topic in parallel and adapt to both modest and radical variations in them. The multiple-topic experiment provided a more accurate representation of a real situation. Furthermore, the experiment was not only intended to investigate the algorithm's performance to represent multiple topics in parallel, it was also to demonstrate the ability of the algorithm to forget a previous topic when it starts to process a new topic.

Furthermore, we devised an evaluation methodology for multiple-topic classification based on email. This is presented in Chapter 4 and Chapter 6. This methodology was used for evaluating the ability of a user profile to continuously adapt in a dynamically changing environment. Following this approach, we believe that it can be used for testing the ability of adaptive systems, such as AIS, for on-line learning (and forgetting) in a complex, multidimensional and dynamic environment. Furthermore, iterating over the same set of documents and suddenly switching between topics of interest causes discontinuities. Nevertheless, how the system reacts to these discontinuities is an interesting test for the system's dynamics.

Implementation of Dynamic Clonal Selection (DCS) for Adapting a User Profile

To achieve the adaptation of our single and multiple-topic profile to changes in a user's interest, we were inspired by the immune theories of dynamic clonal selection (DCS). Although studies related to CSA and DCS have become increasingly popular, to the best of our knowledge, there has been no study so far which has discussed its application in adapting a user profile for content-based document filtering. In Chapter 6, we presented an approach using DCS to adapt a user

profile (known as ProAdDCS) in adaptive document filtering. We have briefly outlined some of the potential uses of DCS for creating an adaptive user profile. Through ProAdDCS, the profile appears to adapt to a variety of changes ranging from frequent variations in a user's short-term needs to occasional radical changes such as the emergence of the new topic of interest and a loss of interest in a particular topic. The profile can learn what are interesting topics or forget topics that are no longer of interest. The proposed adaptation for a single and multiple-topic profile through DCS represents a significant innovation over existing practice, since it adapts single-topic profiles with a steady pace or using a discrete adaptation level.

The inclusion of upper limits on the number of naive and memory B cells in DCS was an ad-hoc approach, and we did not manage to identify appropriate parameter values so that the size of the population would reach an equilibrium irrespective of topic of interest. It is also impractical to allow the population to escalate. Given the parameter values, the number of naive and memory cells initially increases and reaches the upper limit after at most 100 relevant documents. Subsequently, after approximately 150 relevant documents the number of naive B cells progressively declines and, given enough relevant documents, it can be depleted. The number of memory cells, on the other hand, remains relatively static after reaching its upper limit. This behaviour is possibly due to competition between naive and memory cells. As the best naive cells become memory cells, naive B cells are progressively left with less competent vectors with decreasing likelihood of being activated.

7.2 Limitations and Future Work

This section highlights some of the limitations identified in this work and recommends some directions for future work in this field.

Implementation of Evaluation by Real Users

One of the main limitations identified in this thesis is that the proposed approach has not been tested and implemented in a real user study. A particular reason for this was the time constraint and the practical difficulty of finding a large number of users willing to participate in the experiment. In order to have participation from users, users must be familiar with the system before they can use it to retrieve a relevant document.

Although evaluation based on a sample of users may provide a good insight

into the human-related issues that an IF system has to resolve [48, 53], nevertheless, the heterogeneity of users and the difficulties in controlling the experimental parameters render this kind of evaluation difficult to reproduce [44]. One solution is to simulate users. The simulation involves the use of a document collection in which the relevance of documents to specific topic categories is known in advance. *Virtual* or *synthetic users* with specific interests in one or more of these categories can therefore be used. Furthermore, simulated experiments can be reproduced accurately and it has been claimed in [27] that experiments with simulated users were more conclusive than experiments with real users.

Improving Web Page Pre-Processing

As the ProAdDCS is applied in web content documents, it could be further improved if it were able to disambiguate noise from content on a web page. Noise is created by adverts, banners, navigation panels and suchlike and is a distraction from the real content on the page. Furthermore, it should be noted that when viewing a web page, two content sections rendered closely on the screen may not appear in close proximity to each other in the raw HTML. Thus, the document generation which relies on the proximity of content to hyperlinks may become confused. However, associating a hyperlink with text as it appears on the screen rather than as it appears in the raw HTML of the page would be a significant research topic in itself.

Ignoring Negative Feedback

In the implementation of the ProAdDCS, we did not consider negative feedback. Allowing for user feedback in negative classification and implementing proper reactions to user feedback could be one way of making the ProAdDCS more adaptive. When non-relevant documents are treated as negative feedback, the population of naive cells would rapidly decline. Although a variable population size is a significant advantage of AIS over conventional GAs, which use a constant population size, the dynamic control of a population is not straightforward at all. It is an important research issue, but outside the scope of the current work.

Application to Other Types of Information Resources

This work has focused on textual information. The ability of ProAdDCS to retrieve information only from hypertext (HTML) documents is an acknowledged limitation. Further work could include developing the ability of ProAdDCS to read the content of files such as the Adobe Portable Document or Postscript, and

this would be a great improvement as it would allow users to find more results. Furthermore, it might have potential to be applied to other media such as audio and image, in which adaptivity to dynamic environments is crucial. These are all interesting potential future directions for this work.

7.3 Revisiting the Research Questions

Chapter 1 defined the main research question of this thesis, which was:

> *Can the AIS algorithm be developed that derives and maintains diversity to manage adaptation on profile on both short-term and long-term changes?*

From the main research question, some of the subsidiary questions were identified in order to answer the main research question. They are as follows:

1. What are the immunological properties that can be addressed in the problem of profile adaptation?
2. What is an effective mechanism to employ the representation of profile and information items?
3. How can we build an AIS that has the ability to adapt to changes on a user profile given the multiple interest and radical changes on interest?
4. What are the baseline algorithm(s) to evaluate the AIS performance?

To investigate the main research questions and subsidiary questions, a list of research objectives was identified. Having summarised all the chapters of this thesis, this section revisits the initial objectives of this research and draws some concluding remarks. Here are the research objectives of this thesis:

- **RO1:** *To identify an immunological principle and the AIF properties based on the principled approach and to work out how to instantiate those properties in the context of the profile adaptation problem.*
  Chapter 2 and Chapter 3 presented a review of the literature regarding theoretical concepts of IF topics, which included the general context of IF. An IF system features properties of adaptability to changing user profile interest and deals with the heterogeneous nature of information in incoming data streams. These characteristics are likely to need to be addressed by an AIF system. Having noted the role-adaptive user profile necessary in AIF, Chapter 2 presented the characteristics of a user profile, the challenges

of profile adaptation, and the existing work that has focused on adaptive user profiles. Based on the dynamic nature of profile adaptation, we were inspired to build a user profile that can continuously maintain a representation of developing user interests within a changing information environment through AIS. An AIS has the inherent ability to boost and maintain the diversity of the immune repertoire achieved through the preservation of diversity. Therefore, a comprehensive review of the literature regarding the human immune system and the AIS was presented in Chapter 3. To further identify characteristics or properties inherent in the biological system, that is, the immune system and the application domain, we followed the principled meta-probes suggested in [5] that address notions such as Openness, Diversity, Interaction, Structure and Scale, otherwise known as the *ODISS* meta-probes. This principled abstraction led to a comprehensive set of requirements which allowed us to identify an appropriate property of the immune system and the problem domain studied. From the principled abstraction, this work suggested that profile adaptation can be developed by incorporating ideas from aspects of dynamic clonal selection (DCS) with the use of gene libraries to maintain sufficient diversity, which supports learning in a dynamically changing environment.

- **RO2:** *To establish an experimental testbed in the email environment in order to investigate the performance of AIS algorithms in classifying emails on the basis of the user's interest with regard to single and multiple email topics.*
  Chapter 4 presented the development of the AIS algorithm in a text-mining related scenario, particularly email classification. The experiment using AIS for email classification was based on Secker's algorithm [9]. In the original AISEC version, the purpose is to classify emails as interesting or uninteresting according to the subject and sender of the email. Development of the AISEC algorithm has attracted researchers to further experiment on email classification, for example, Prattipati and Hart [10]. Our modification to AISEC was to consider the body of the email in addition to the email subject and sender field as this provides a richer set of words, and to incorporate the WordNet corpus as a source of synonyms, which allows more accurate classification of emails and improves the capability to identify new and potentially uninteresting email. Despite experimenting on single topic classification (or binary classification which discriminating either email interesting or not interesting), we were interested in experimenting with AIS in a changing interest scenario and classifying emails based on multiple email

topics because this more accurately reflects a real-life situation. Moreover, the experiment was not only to investigate the algorithm's performance in representing multiple topics in parallel, but also to demonstrate the ability of the algorithm to forget a previous topic when it starts to process a new topic. The evaluation was carried out on the classification performance using non-parametric statistical tests. In these non-parametric statistical analyses, we employed the Mann-Whitney-Wilcoxon or rank-sum test [125] to measure whether the performance samples had a different distribution (statistically significantly different) or a different median, and Vargha-Delaney A statistics [126] to measure the effect size between these algorithms' performances (to determine whether or not they are scientifically significantly different). In the experiment, the extended AISEC performed better than the comparative approach and this gradation was consistent throughout most of the evaluation periods. The results can be summarized as that the antibody-antigen interaction of B-cells and the introduction of the WordNet corpus and consideration of the body of email to help further improve the capability for identifying new and potentially uninteresting emails have a positive effect on the adaptability of the profile to changes in user interest and on the profile's response to the email corpus.

After testing and evaluating the algorithm using the comparative approach, Chapter 5 examined the influence of the algorithm's parameters, known as a sensitivity analysis. This was because the dynamic behaviour of an algorithm can be controlled by the algorithm's parameters. The results from the sensitivity analysis gave insights into how these parameters can be optimised to provide a better performance for each of the performance metrics. The sensitivity analysis was carried out through two tasks: single-topic classification and multiple-topic classification. For multiple-topic classification, the analysis was conducted by considering three different cases:

1. case 1: topics with large number of emails together with topics with low number of emails,
2. case 2: both topics have low number of emails, and
3. case 3: both topics have high number of emails.

The purpose of the test was to identify the level at which the algorithm becomes reliable and acceptable in an extreme case scenario and a mild case scenario. The results showed that in the extreme cases with low number of emails, the test did not produce satisfactory results. This may have been due

to the insufficient number of emails to classify, which may have affected the algorithm's overall performance. An additional memory detector is needed to extend the system's capability to perform the classification task. However, in the mild cases with high number of email topics and low number of emails (Case 1) and email topics with high number of emails (Case 3), the experiment showed that the most influential parameters were the classification threshold (Kc), the affinity threshold (Ka) and the initial number of memory cells (Kt).

- **RO3:** *To develop an AIS algorithm which incorporates adaptivity to adapt to changes in a user profile given multiple interest and radical changes in interest on adaptive document filtering.*
  Chapter 6 presented a further investigation of AIS for profile adaptation and widened the experiment in adaptive document filtering. User interests are by nature dynamic. A combination of parameters causes a variety of changes. Frequent changes in the user's short-term needs contribute to progressive changes in the user's long-term interests and vice versa. The user's interest may shift frequently between different topics or related sub-topics. New topics and sub-topics of interest emerge and interest in a particular topic might be lost. Despite the complexity of the dynamic nature of user interest, this chapter suggested that profile adaptation can be developed by incorporating ideas from aspects of dynamic clonal selection (DCS) with gene libraries to maintain sufficient diversity through a synset relationship based on WordNet. Inspired by clonal selection, the dynamics inherent in AIS algorithms are believed to be powerful enough to make AIS a successful solution to the problem of profile adaptation. DCS has been identified as an AIS algorithm which supports learning in a dynamically changing environment [91, 115–118]. DCS can be used to maintain the profiles with gene libraries maintaining a sufficient diversity for the set of terms that can be added to the profile during mutation. The goal was to adapt our multiple-topic profile both to short-term variations in the user's need and to progressive, but potentially radical changes in long-term interests. This chapter presented the algorithm and the process of DCS known as profile adaptation through dynamic clonal selection (ProAdDCS). The algorithm was tested and evaluated using simulated users. As in Chapter 4, this chapter also presented a comparative study which involved testing against another similar system which focuses on adapting a user profile. Comparative performance is important in order to assess whether the works are incremen-

tal improvements on the state of the art or evolutions of existing work. We may argue that the experiment results have been positive. The experiments' results have indicated that the ProAdDCS profile responds to short-term variations and occasional radical changes in the composition of a stream of feedback documents. Furthermore, the result on ProAdDCS's profile has also been positive for tasks comprising the unrelated topics learned in parallel as well as not learned in parallel. As a result the following adaptive behaviors are observed:

1. More than one topic of interest may be learned from scratch and in parallel with a single profile.
2. The relative importance of topics in the profile varies in response to short-term variations in the distribution of relevant documents in the training set.
3. An existing profile representing more than one topic of interest may learn an emerging topic of interest, without what is already represented being significantly affected.
4. A profile representing more than one topic of interest may forget a topic that, in contrast to the rest of the topics, no longer receives positive feedback.

In Chapter 6, the adaptation to both variations in a user's short-term needs and radical changes in long-term interests has been achieved with a single and multiple-topic profile, through a process that exhibits characteristics of DCS. With the preservation of diversity (heterostasis) in our approach, the adaptive user profile can be achieved and changes in the interest inferred dynamically. Furthermore, through the introduction of diversity (heterogenesis), in the the gene libraries with synset relationship based on WordNet, the system is further able to learn a new topic of interest. The profile appears to be able to adapt to a variety of simulated changes in a virtual user's interest.

APPENDIX

A

Population-Based Immune-Inspired Information Filtering

Table A.1: Population-based Immune-inspired Information Filtering

Application		Algorithm Aspect	Approaches		Limitation
Reference	**Problem Solved**		**Immune Mechanism**	**Representation**	
Haidar and Rocha [149,150]	adaptation on concept drift in Spam detection	• virtual cells: randomly sampled features from messages with a score being determined on its specific recognition by detectors • final classification given by the threshold sum of the feature score • elimination of cell death for long term memory	cross regulation model of T cells	Ag: words of e-mail in subject and body	• concept drift is not much consider • with concept drift the proposed model achieve a similar accuracy with Nave Bayes model • more false positives
Secker et. al [9]	E-mail classification	• cloning: new cells entering the naïve cell set are mutants of existing cell • co-stimulation signal based on user feedback • two recognition regions to increases classification accuracy • lifespan of naïve B cell and memory B-cell	B cell with clonal selection	B cells represent one class of data (uninteresting emails); gene libraries of B cell receptor contains of words; a subject words and a senders word	• small difference of mean accuracy on concept drift • not clearly shown the algorithm capability for continuous learning (no result have been published)

Table A.2: Population-based Immune-inspired Information Filtering

Application		Algorithm Aspect	Approaches		Limitation
Reference	**Problem Solved**		**Immune Mechanism**	**Representation**	
Prattipati and Hart [10]	E-mail classification; adaptability on changing interest; words exploitation	• changing interest on user feedback • position biased mutation operator scheme to exploit words in B cell vector	B cell mechanism and clonal selection	Ag: words of e-mail in subject and body	• positional biased mutation shown to be less effective to small number of e-mail • should consider investigation of e-mail body to maintain tractability • does not consider changes with multi interesting e-mail
Guzella et. al [151]	Spam detection	• mutation operator: point-wise mutation and directed mutation • selection process is randomly selected and does not based on affinity level • message classify as spam if at least one B detector is activated	innate and adaptive inspired; negative selection and clonal selection (CLONALG)	macrophages detectors recognize senders address; adaptive detector analyze the subject and body message; terms represented as binary encoding	• analogous of general immune system mechanism • not clearly mention about the role of user feedback on classification

Table A.3: Population-based Immune-inspired Information Filtering

Application		Algorithm Aspect	Approaches		Limitation
Reference	**Problem Solved**		**Immune Mechanism**	**Representation**	
Oda and White [152–154]	Spam detection	• weighted Ab are matched to a given message • scoring scheme: [152]–simple sum of messages matched, [153]– weighted average of messages matched, [154] – Bayes score • apply different type of libraries [154]	B-cell and antibody (Ab) mechanism	message subject and body: Ag; genes regular expressions (patterns that match variety of string) as Ab; weight as memory for each lymphocytes that bind to a given pathogen; scoring scheme for messages matched by the lymphocyctes	• mutation of B-cell (antibodies) is not consider • performance over period of time is not mentioned • correctly identified 90% of Spam message
Sobecki and Szczepanski [155]	collaborative filtering for user interface and wiki-news article recommendation	• weighted kappa affinity measure to calculate correlation coefficient • article recommendation based on high amount of Ab concentration • verification to interface layout based on Ab correlation	B-cell and antibody (Ab) mechanism	wiki news:body; new user: Ag and similar users: Ab	analogous of general principle of Ag-Ab matching; applied in terms of selecting group of people (Ab) who has similar preference with particular user (Ag)

Table A.4: Population-based Immune-inspired Information Filtering

Application		Algorithm Aspect	Approaches		Limitation
Reference	**Problem Solved**		**Immune Mechanism**	**Representation**	
Sarafijanovic and Boudec [156]	Spam detection with collaborative approach	• learn signatures of patterns by randomly sampling words from a message • incoming messages from collaborating anti-spam system • text string transform to binary string by one way similarity hash functions	immune adaptive inspired; negative selection to process signature and danger signal	message subject and body: Ag; genes regular expressions: Ab; weight as memory lymphocytes; scoring scheme for messages matched by the lymphocytes	analogous to general principle of immune system
Chao and Forrest [18, 157]	computer generated art; music filter; simultaneously group filtering	• negative detector to prevent the retrieval entries that are too similar to previously rejected • co-stimulation based on user relevance feedback • coarse-grained detector to reduce the training time for each user	negative selection and co-stimulation theory	accumulated detector that form the user preferences	group filtering is simplified based on combination of individual filter and was unsuccessful due to the inconsistent preference of user profile

APPENDIX

B

Network-Based Immune-Inspired Information Filtering

Table B.1: Network-based Immune-inspired Information Filtering

Application		**Representation**	**Algorithm Aspects**		**Limitation**
Reference	**Problem Solved**		**Immune Algorithm**	**Modification**	
Acilar and Arslan [158]	sparsity and scalability of data set	Ag-Ab representation in rating (memory) matrix	aiNet algorithm	pearson correlation for distance measurement; k-means clustering for neighbourhood measure	small scope (through compression rate) cover the scalability issue
Cayzer and Aickelin [104]	movie recommendation	• Ag-Ab: for matching; Ab-Ab: for diversity; somatic hypermutation: for sparsity • SWAMI framework: data encoding; Pearson: similarity measure between neighbour	Idiotypic Network	• neighbourhood selection based on matching score • Ab concentration depends on Ag-Ab • prediction of rating based on weighted average over the neighbourhood • scalability: memory matrix of Ab and suppression process	• idiotypic effect based on one measurement while recommendation may be likely to be a combination of factors • suffer from shortage in scalability in time and space • require the entire existing data to be maintained and analyze repeatedly when new user rating is added

Table B.2: Network-based Immune-inspired Information Filtering

Application		Representation	Algorithm Aspects		Limitation
Reference	**Problem Solved**		**Immune Algorithm**	**Modification**	
Yue et. al [159, 160]	rating-based recommendation	Ag: user who have voted; Cosine Similarity: similarity measure	aiNet algorithm and ICAInet algorithm [159]	• no distinction between network cells and their surface molecule • Ag-Ab and Ab-Ab interaction is quantified through similarity measure • involve incremental clustering and Minimal Spanning Tree (MST) to calculate cluster	• does not show changes on user profile; scalability (in time and space) is not tested • MST is not efficient due to high number of resources to produce cluster • correct classification rates of SPAM and legitimate message has not been reported

Table B.3: Network-based Immune-inspired Information Filtering

Application		Representation	Algorithm Aspects		Limitation
Reference	**Problem Solved**		**Immune Algorithm**	**Modification**	
Pablo de Castro et. al [161, 162]	joint analysis of user and item for collaborative filtering	index of row (n) and column (m) with integer as element; population based on cloning, mutation and suppression rate	aiNet algorithm	immune inspired algorithm to generate biclustering (perform clustering of rows and column simultaneously); mutation operator based on simple random of insertion/removal	• diversity through hypermutation on aiNet algorithm; not clearly show biclustering provides diversity on group • scalability based on prediction on various dataset, responsive (performance) of the algorithm to correctly predict; recommend on scalable data set is not consider • user relevance feedback is not involve

Table B.4: Network-based Immune-inspired Information Filtering

Application		Representation	Algorithm Aspects		Limitation
Reference	**Problem Solved**		**Immune Algorithm**	**Modification**	
Bezerra et. al [163]	Spam detection	binary vector of word; dimensionality reduction: words that appear rarely (low weight)	Antibody Network(ABNET algorithm); Real-valued Antibody NETwork (RABNET algorithm)	• supervised learning network of realvalued weight for antibody network • weight is update based on Learning Vector Quantization (LZQ) • network size is dynamically adjusted depends on training data; total cost ration (TCR) as stopping criteria for training	it does not shows in terms of the ability to track changes over time.

APPENDIX

C

Non-Parametric Statistical Analysis on Sensitivity Analysis for Single-Interest Classification

Ka			Kc			Km			Ksb			Ksm			Kl			Kt		
P-Value	Rank Sum	A-Test	P-Value	Rank Sum	A-Test	P-Value	Rank Sum	A-Test	P-Value	Rank Sum	A-Test	P-Value	Rank Sum	A-Test	P-Value	Rank Sum	A-Test	P-Value	Rank Sum	A-Test
8.06E-10	7112	0.809167	8.82E-19	7816	0.920562	0.4593	5705.5	0.535153	0.463383	5704	0.534879	0.640919	5634.5	0.522188	0.333142	5764.5	0.545928	0.013734	6153.5	0.616965
9.29E-07	6789	0.752374	9.89E-18	7744.5	0.907505	0.345718	5758	0.544741	0.775598	5587.5	0.513605	0.519679	5680.5	0.530588	0.346847	5757.5	0.544649	0.026206	6091.5	0.605643
9.69E-09	7005	0.793006	3.31E-17	7708	0.90084	0.532374	5675.5	0.529675	0.426455	5720	0.537801	0.519796	5680.5	0.530588	0.160347	5878	0.566654	0.000481	6421	0.665814
5.98E-08	6922.5	0.777027	8.30E-15	7532.5	0.868791	0.771172	5589	0.513879	0.753557	5595	0.514974	0.992308	5516	0.500548	0.197208	5848	0.561176	0.00542	6236	0.632031
2.28E-07	6858.5	0.76744	2.04E-05	6620.5	0.702246	0.627152	5639.5	0.523101	0.312379	5776	0.548028	0.759103	5593	0.514609	0.388037	5737.5	0.540997	0.142757	5894	0.569576
2.98E-07	6845	0.770453	2.39E-09	7064.5	0.783327	0.476943	5698	0.533784	0.976959	5521	0.501461	0.460748	5705	0.535062	0.403185	5730.5	0.539719	0.545049	5355.5	0.471238
9.63E-05	6527	0.70882	3.71E-08	6943	0.76114	0.466372	5702.5	0.534606	0.150062	5887.5	0.568389	0.6431	5392.5	0.477995	0.500007	5688.5	0.532049	0.782969	5441	0.486852
0.000157	6496	0.702611	2.39E-07	6856	0.745252	0.474516	5699	0.533966	0.841287	5565.5	0.509587	0.785733	5584	0.512966	0.521994	5679.5	0.530405	0.811225	5575.5	0.511413
3.14E-06	6725.5	0.745709	2.28E-11	7251	0.817385	0.478269	5697.5	0.533692	0.562321	5664	0.527575	0.537648	5673.5	0.52931	0.220584	5831.5	0.558163	0.953917	5497.5	0.497169
1.92E-05	6624.5	0.727447	4.07E-07	6829	0.740321	0.499947	5688.5	0.532049	0.814384	5451.5	0.488769	0.488932	5693	0.532871	0.251107	5811.5	0.554511	0.613762	5381.5	0.475986
0.000134	6506	0.711833	1.77E-05	6628	0.703616	0.664238	5626	0.520636	0.871493	5470.5	0.492239	0.688459	5617.5	0.519083	0.22947	5825.5	0.557067	0.857859	5466	0.491417
1.80E-05	6627	0.734295	3.13E-08	6951.5	0.762692	0.639794	5635	0.522279	0.873004	5555	0.50767	0.212277	5837.5	0.559259	0.349797	5756	0.544375	1	5513	0.5
8.96E-08	6903	0.782323	3.25E-07	6841	0.742513	0.590766	5653	0.525566	0.165614	5873.5	0.565833	0.136008	5900.5	0.570763	0.456747	5706.5	0.535336	0.488134	5693.5	0.532962
0.001711	6328.5	0.673484	8.50E-08	6904.5	0.754109	0.606711	5647	0.52447	0.157216	5881	0.567202	0.466501	5702.5	0.534606	0.40632	5729	0.539445	0.592141	5652.5	0.525475
0.000417	6430.5	0.695489	1.57E-07	6876	0.748904	0.803578	5448	0.48813	0.353737	5754.5	0.544102	1	5513	0.5	1	5513	0.5	0.555488	5666.5	0.528031
0.014504	6148.5	0.643809	2.38E-05	6611	0.700511	0.616108	5643.5	0.523831	0.22148	5831	0.558072	0.466269	5702.5	0.534606	0.681481	5620	0.51954	0.356306	5753	0.543828
1	5513	0.543097	0.258798	5806	0.553506	0.475748	5698.5	0.533875	0.470261	5701	0.534332	0.553955	5667	0.528123	0.647983	5632	0.521731	0.522654	5679.5	0.530405
0.591837	5373.5	0.51114	0.53706	5673.5	0.52931	0.935511	5534.5	0.503926	0.228364	5826.5	0.55725	0.484926	5694.5	0.533145	0.8471	5563.5	0.509222	0.323527	5770	0.546932
0.67809	5405	0.52657	0.579805	5657	0.526297	0.960028	5499.5	0.497535	0.376044	5743.5	0.542093	0.88052	5552.5	0.507213	0.61748	5643	0.52374	0.627559	5639.5	0.523101
0.28354	5234.5	0.485756	0.358727	5751.5	0.543554	0.723978	5421	0.483199	0.311305	5776.5	0.548119	0.294648	5785.5	0.549763	0.526091	5678	0.530131	0.676959	5621.5	0.519814
0.363263	5277	0.5	0.266498	5801	0.552593	0.940067	5493	0.496348	0.323252	5770	0.546932	0.458867	5705.5	0.535153	0.260242	5805.5	0.553415	0.871472	5555.5	0.507761
0.388317	5289	0.495617	0.084384	5066.5	0.484624	0.396396	5733.5	0.540267	1	5513	0.5	0.43073	5718	0.537436	0.522171	5679.5	0.530405	0.411103	5727	0.53908
0.337407	5264	0.495526	1	5513	0.5	0.776731	5587	0.513514	0.560675	5361.5	0.472334	0.460624	5705	0.535062	0.803477	5578	0.51187	0.799375	5579.5	0.512144
0.557413	5360.5	0.516709	0.986111	5508	0.499087	0.877444	5472.5	0.492604	0.872997	5471	0.49233	0.832072	5568.5	0.510135	0.283798	5791.5	0.550858	0.53673	5674	0.529401
0.106926	5095.5	0.464847	0.518398	5345.5	0.469412	0.414744	5725	0.538714	0.889761	5549.5	0.506665	0.488163	5693.5	0.532962	0.119685	5916.5	0.573685	0.416177	5724.5	0.538623
0.641049	5391.5	0.512053	0.595583	5651	0.525201	0.174557	5866	0.564463	0.869982	5470	0.492148	0.411617	5726.5	0.538988						
0.797467	5446	0.528488	0.458877	5705.5	0.535153	0.753429	5595	0.514974	0.850303	5463.5	0.490961	0.55828	5665.5	0.527849						
0.863723	5468	0.532597	0.054526	6011.5	0.591034	1	5513	0.5	0.854796	5465	0.491234	0.22646	5827.5	0.557432						
0.781396	5585.5	0.551132	0.228661	5825.5	0.557067	0.492735	5691.5	0.532597	0.758004	5432.5	0.485299	0.341929	5760	0.545106						
0.856067	5560.5	0.544649	0.1658	5872.5	0.56565	0.577341	5658	0.526479	0.559245	5361	0.472243	0.688746	5617.5	0.519083						
0.842507	5461	0.529766	0.016931	6133	0.613221	0.689803	5617	0.518992	0.996926	5511.5	0.499726	0.859318	5559.5	0.508492						
0.542046	5354.5	0.508309	0.002324	6303.5	0.644357	0.567017	5662	0.52721	0.584385	5370.5	0.473977	0.70147	5613	0.518262						
0.882008	5474	0.526205	0.000497	6416.5	0.664993	0.369355	5746.5	0.542641	0.944721	5494.5	0.496622	0.367386	5747.5	0.542823						
0.975371	5521.5	0.544649	0.000506	6415	0.664719	0.5789	5657.5	0.526388	0.232102	5824	0.556793	0.638268	5635.5	0.52237						
0.756338	5594	0.547936	0.000142	6500.5	0.680332	0.424145	5721	0.537984	0.261758	5805	0.553324	0.628522	5639	0.523009						
0.473458	5699.5	0.571585	6.41E-05	6550	0.689372	0.179572	5862	0.563733	0.381303	5741	0.541636	0.065446	5991.5	0.587381						
0.278135	5795	0.586925	0.000185	6483.5	0.677228	0.246591	5814	0.554967	0.338965	5762	0.545471	0.386903	5738	0.541088						
0.528437	5677	0.565285	3.04E-05	6595.5	0.697681	0.138578	5898	0.570307	0.81888	5573	0.510957	0.366946	5747.5	0.542823						
0.854762	5561	0.539354	9.98E-05	6522.5	0.68435	0.377508	5742.5	0.54191	0.789022	5583	0.512783	0.625887	5640	0.523192						
0.838128	5459.5	0.526936	0.000153	6495	0.679328	0.454323	5707.5	0.535519	0.451774	5709	0.535793	0.70553	5611.5	0.517988						
0.471803	5700	0.577885	0.001071	6362	0.65504	0.298214	5783.5	0.549397	0.471232	5700.5	0.53424	0.407642	5728.5	0.539354						
0.802108	5447.5	0.524196	0.000126	6508.5	0.681793	0.236904	5820.5	0.556154	0.466715	5702.5	0.534606	0.721485	5606	0.516983						
0.839588	5460	0.527575	1.74E-05	6628.5	0.703707	0.382834	5740	0.541454	0.500727	5688.5	0.532049	0.506465	5686	0.531592						
0.787316	5583.5	0.55168	6.29E-06	6685.5	0.714116	0.695665	5615	0.518627	0.777206	5587	0.513514	0.325826	5768.5	0.546658						
0.735825	5425	0.520909	5.26E-05	6563	0.691746	0.037478	6053.5	0.598703	0.853282	5561.5	0.508857	0.648087	5632	0.521731						
0.598543	5650	0.569668	6.51E-06	6683	0.71366	0.053647	6014	0.59149	0.638457	5635.5	0.52237	0.436534	5715.5	0.53698						
0.60653	5647	0.564007	1.47E-05	6637.5	0.705351	0.045809	6032	0.594777	0.640975	5634.5	0.522188	0.517417	5681.5	0.530771						
0.696924	5614.5	0.559533	0.000559	6408.5	0.663532	0.037208	6054	0.598795	0.615124	5644	0.523923	0.528617	5677	0.529949						
0.918546	5540	0.548576	3.04E-05	6595.5	0.697681	0.042852	6039	0.596056	0.718644	5607	0.517166	0.334642	5764	0.545836						
0.689468	5409	0.521549	0.000386	6434.5	0.66828	0.151338	5886	0.568115	0.769891	5589.5	0.51397	0.513595	5683	0.531045						

Figure C.1: Non-Parametric Analysis for Single-Interest Classification

APPENDIX

D

Non-Parametric Statistical Analysis on Sensitivity Analysis for Multi-Interest Classification: Case 1 Scenario

Ka			Kc			Km			Ksb			Ksm			Kl			Kt		
P-Value	Rank Sum	A-Test	P-Value	Rank Sum	A-Test	P-Value	Rank Sum	A-Test	P-Value	Rank Sum	A-Test	P-Value	Rank Sum	A-Test	P-Value	Rank Sum	A-Test	P-Value	Rank Sum	A-Test
2.01E-11	7257	0.818481	4.35E-16	7629	0.886413	3.14E-14	7489.5	0.860939	4.13E-22	8029.5	0.959551	0.784757	5441.5	0.486943	0.950846	5496.5	0.496987	0.90025	5546	0.506026
5.18E-09	7032	0.777392	6.73E-15	7540.5	0.870252	3.17E-13	7410.5	0.846512	9.93E-21	7942.5	0.943663	0.649649	5394.5	0.47836	0.845782	5564	0.509313	0.629823	5638.5	0.522918
9.40E-12	7285.5	0.823685	1.08E-14	7525	0.867421	5.55E-13	7390.5	0.84286	1.12E-20	7939.5	0.943115	0.909624	5483	0.494522	0.574178	5659.5	0.526753	0.329688	5766.5	0.546293
9.03E-10	7106.5	0.790997	1.34E-12	7358	0.836925	7.56E-11	7207.5	0.809441	7.42E-20	7886	0.933346	0.538167	5352.5	0.47069	0.740337	5599.5	0.515796	0.53443	5674.5	0.529492
5.76E-09	7026.5	0.776388	0.001474	6339.5	0.650931	0.002956	6286	0.641161	2.18E-09	7067.5	0.783875	0.983117	5507	0.498904	0.963148	5525.5	0.502283	0.671648	5623.5	0.520179
3.69E-10	7141.5	0.797389	0.000186	6485	0.677502	0.000389	6436.5	0.668645	1.01E-10	7193.5	0.806885	0.328146	5258.5	0.453524	0.808393	5449.5	0.488404	0.955429	5498	0.497261
2.87E-06	6728.5	0.721969	0.000183	6485.5	0.677593	0.000668	6398.5	0.661706	3.39E-11	7235.5	0.814554	0.811577	5575.5	0.511413	0.737617	5425.5	0.484021	0.889638	5476.5	0.493335
2.85E-06	6729.5	0.722151	0.000337	6445	0.670197	0.000846	6381	0.65851	8.59E-10	7106.5	0.790997	0.450674	5316.5	0.464116	0.800698	5447	0.487947	0.958462	5499	0.497443
2.24E-07	6859.5	0.745891	1.18E-06	6776.5	0.730734	1.43E-05	6642.5	0.706264	7.20E-15	7536	0.86943	0.707549	5415	0.482104	1	5513	0.5	0.779853	5586	0.513331
4.07E-07	6829.5	0.740413	0.009254	6189.5	0.623539	0.01127	6172.5	0.620435	1.39E-08	6987	0.769175	0.262934	5221.5	0.446768	0.677553	5621.5	0.519814	0.399595	5732	0.539993
2.71E-05	6603.5	0.699142	0.102084	5938	0.577611	0.082287	5965	0.582542	4.87E-06	6700	0.716764	0.287222	5236	0.449416	0.860922	5559	0.5084	0.63255	5637.5	0.522736
1.99E-05	6622	0.70252	0.008878	6193.5	0.62427	0.011579	6170	0.619978	1.61E-08	6981.5	0.76817	0.398319	5293	0.459825	0.895816	5547.5	0.5063	0.750388	5596	0.515157
1.63E-07	6875	0.748722	0.055511	6011	0.590942	0.05363	6015	0.591673	9.69E-07	6785	0.732286	0.633218	5637.5	0.522736	0.779853	5440	0.486669	1	5513	0.5
0.001366	6344.5	0.651844	0.20081	5846	0.560811	0.143004	5894.5	0.569668	7.86E-06	6674.5	0.712107	0.588759	5654	0.525749	0.845777	5462	0.490687	0.996922	5511.5	0.499726
0.011261	6171.5	0.620252	0.20407	5843.5	0.560354	0.164365	5875	0.566107	7.67E-06	6676	0.712381	1	5513	0.5	0.40321	5295.5	0.460281	0.456747	5319.5	0.464664
0.019622	6119	0.610665	0.361412	5750.5	0.543371	0.373227	5745	0.542367	1.84E-05	6625	0.703068	0.888311	5550	0.506757	0.650933	5395	0.478451	0.823106	5454.5	0.489317
0.023365	6101	0.607378	0.874236	5554.5	0.507579	0.640988	5634.5	0.522188	0.0006	6401.5	0.662253	0.795043	5445	0.487582	0.905004	5481.5	0.494248	0.815651	5574	0.51114
0.09404	5947	0.579255	0.834935	5458.5	0.490047	0.932448	5535.5	0.504109	0.002135	6308.5	0.64527	0.900511	5546	0.506026	0.579053	5368.5	0.473612	0.775409	5438.5	0.486395
0.044354	6034	0.595142	0.725653	5421.5	0.483291	1	5513	0.5	0.011621	6167	0.61943	0.987726	5517.5	0.500822	0.576148	5367.5	0.47343	0.699524	5412.5	0.481647
0.234761	5821	0.556245	0.95848	5527	0.502557	0.754984	5594.5	0.514883	0.00148	6336.5	0.650383	0.947848	5530.5	0.503196	0.717329	5418.5	0.482743	0.850106	5463.5	0.490961
0.219234	5831.5	0.558163	1	5513	0.5	0.725653	5604.5	0.516709	0.000669	6393	0.660701	0.953953	5528.5	0.502831	0.888122	5550	0.506757	0.686635	5618	0.519175
1	5513	0.5	0.000669	4633	0.339299	0.011621	4859	0.38057	1	5513	0.5	0.353737	5271.5	0.455898	0.839713	5460	0.490321	0.966157	5501.5	0.4979
0.174066	5865.5	0.564372	0.121724	5112	0.426771	0.36769	5279	0.457268	0.041671	6040	0.596238	0.155828	5144	0.432615	0.680107	5405.5	0.480369	0.787041	5442.5	0.487126
0.069895	5983	0.585829	0.12613	5117	0.427684	0.339144	5265	0.454711	0.074986	5974	0.584186	0.279381	5231.5	0.448594	0.88512	5551	0.506939	0.698226	5614	0.518444
0.305077	5778	0.548393	0.110022	5099	0.424397	0.358359	5274.5	0.456446	0.08766	5955	0.580716	0.432597	5308.5	0.462655	0.284129	5791	0.550767	0.131147	5903.5	0.571311
0.121173	5915	0.573411	0.807792	5449.5	0.488404	0.940056	5533	0.503652	0.00152	6333.5	0.649836	0.323609	5256	0.453068						
0.050794	6019	0.592403	0.952349	5497	0.497078	0.821664	5572	0.510774	0.002362	6300.5	0.643809	0.258692	5219	0.446311						
0.029501	6077.5	0.603086	0.390396	5736	0.540723	0.339603	5761.5	0.54538	1.35E-05	6640	0.705807	0.322547	5255.5	0.452977						
0.0192	6120.5	0.610939	0.67945	5620.5	0.519631	0.605462	5647.5	0.524562	9.86E-05	6521.5	0.684167	0.255428	5217	0.445946						
0.014824	6144.5	0.615321	0.754611	5594.5	0.514883	0.552867	5667.5	0.528214	0.000556	6407	0.663258	0.170403	5156.5	0.434898						
0.05348	6013.5	0.591399	0.156222	5881.5	0.567294	0.141183	5895.5	0.56985	9.79E-06	6660.5	0.709551	0.327452	5258	0.453433						
0.087921	5955	0.580716	0.029379	6078.5	0.603269	0.038918	6049.5	0.597973	5.64E-08	6921	0.757122	0.225926	5198	0.442476						
0.045677	6031	0.594595	0.024709	6095.5	0.606373	0.035921	6058	0.599525	1.64E-07	6868	0.747443	0.306871	5247	0.451424						
0.01928	6120	0.610847	0.016604	6135	0.613587	0.019927	6118	0.610482	2.29E-08	6962.5	0.764701	0.809992	5576	0.511505						
0.015748	6139	0.614317	0.004481	6251	0.63477	0.006888	6215	0.628196	2.02E-09	7069.5	0.78424	0.877675	5553.5	0.507396						
0.006125	6224.5	0.629931	0.00331	6276	0.639335	0.00577	6230.5	0.631026	1.13E-09	7093	0.788532	0.961645	5526	0.502374						
0.001357	6344.5	0.651844	0.006393	6221	0.629291	0.007478	6208.5	0.627009	5.06E-09	7029	0.776844	0.993862	5515.5	0.500457						
0.007428	6207	0.626735	0.001601	6332.5	0.649653	0.0028	6290	0.641892	3.14E-10	7146	0.79821	0.558581	5360.5	0.472151						
0.03969	6046.5	0.597425	0.003103	6281	0.640248	0.005285	6238	0.632396	6.72E-10	7114.5	0.792458	0.500644	5337.5	0.467951						
0.046435	6029.5	0.594321	0.007823	6203.5	0.626096	0.007985	6202.5	0.625913	1.34E-08	6987.5	0.769266	0.858015	5466	0.491417						
0.002229	6305.5	0.644722	0.020154	6116.5	0.610208	0.020009	6118	0.610482	7.96E-08	6905.5	0.754291	0.820294	5453.5	0.489134						
0.060838	5999	0.588751	0.0044	6252.5	0.635044	0.009658	6186	0.6229	2.86E-09	7054	0.78141	0.691825	5409.5	0.481099						
0.043342	6037	0.59569	0.000765	6386.5	0.659514	0.003817	6265	0.637327	2.02E-11	7253	0.81775	0.918801	5486	0.495069						
0.01354	6153.5	0.616965	0.000441	6425.5	0.666636	0.002224	6308	0.645179	7.66E-12	7288.5	0.824233	0.46704	5323.5	0.465394						
0.080334	5967	0.582907	0.002464	6299.5	0.643627	0.005884	6229	0.630752	4.69E-10	7130	0.795289	0.434572	5309.5	0.462838						
0.003173	6278	0.639701	0.000305	6450	0.67111	0.001389	6344	0.651753	3.31E-12	7321	0.830168	0.580805	5369	0.473703						
0.006083	6223.5	0.629748	0.000681	6395	0.661066	0.002495	6299	0.643535	2.01E-11	7253	0.81775	0.532878	5350.5	0.470325						
0.007631	6205	0.62637	0.013921	6151.5	0.6166	0.024633	6097	0.606647	2.27E-08	6962.5	0.764701	0.634471	5389	0.477356						
0.013012	6157	0.617604	0.000846	6380	0.658327	0.002847	6288.5	0.641618	3.69E-11	7231	0.813733	0.593837	5374	0.474617						
0.055049	6010	0.59076	0.014567	6147	0.615778	0.025156	6095	0.606282	2.86E-08	6952.5	0.762874	0.574342	5366.5	0.473247						

Figure D.1: Non-Parametric Analysis for Multi-Interest Classification: Case 1 Scenario

APPENDIX

E

Non-Parametric Statistical Analysis on Sensitivity Analysis for Multi-Interest Classification: Case 2 Scenario

Ka			Kc			Km			Ksb			Ksm			Kl			Kt		
P-Value	Rank Sum	A-Test	P-Value	Rank Sum	A-Test	P-Value	Rank Sum	A-Test	P-Value	Rank Sum	A-Test	P-Value	Rank Sum	A-Test	P-Value	Rank Sum	A-Test	P-Value	Rank Sum	A-Test
8.82E-19	7489.5	0.918481	4.13E-22	8029.5	0.9864134	8.82E-19	7112	0.8609386	2.01E-11	5705.5	0.959551	0.463383	7816	0.686943	0.640919	5634.5	0.596987	0.4593	5705.5	0.606026
9.89E-18	7410.5	0.977392	9.93E-21	7942.5	0.970252	9.89E-18	6789	0.8465121	5.18E-09	5758	0.943663	0.775598	7744.5	0.67836	0.519679	5680.5	0.609313	0.345718	5758	0.622918
3.31E-17	7390.5	0.923685	1.12E-20	7939.5	0.9674215	3.31E-17	7005	0.8428598	9.40E-12	5675.5	0.943115	0.426455	7708	0.694522	0.519796	5680.5	0.626753	0.532374	5675.5	0.646293
8.30E-15	7207.5	0.990997	7.42E-20	7886	0.9369248	8.30E-15	6922.5	0.8094412	9.03E-10	5589	0.933346	0.753557	7532.5	0.67069	0.992308	5516	0.615796	0.771172	5589	0.629492
2.04E-05	6286	0.976388	2.18E-09	7067.5	0.9509313	2.04E-05	6858.5	0.8411614	5.76E-09	5639.5	0.783875	0.312379	6620.5	0.698904	0.759103	5593	0.602283	0.627152	5639.5	0.620179
2.39E-09	6436.5	0.897389	1.01E-10	7193.5	0.9775018	2.39E-09	6845	0.868645	3.69E-10	5698	0.806885	0.976959	7064.5	0.653524	0.460748	5705	0.588404	0.476943	5698	0.697261
3.71E-08	6398.5	0.821969	3.39E-11	7235.5	0.9775931	3.71E-08	6527	0.7617056	2.87E-06	5702.5	0.814554	0.150062	6943	0.511413	0.6431	5392.5	0.584021	0.466372	5702.5	0.693335
2.39E-07	6381	0.822151	8.59E-10	7106.5	0.9701972	2.39E-07	6496	0.7585099	2.85E-06	5699	0.790997	0.841287	6856	0.564116	0.785733	5584	0.587947	0.474516	5699	0.697443
2.28E-11	6642.5	0.845891	7.20E-15	7536	0.8307341	2.28E-11	6725.5	0.7062637	2.24E-07	5697.5	0.86943	0.562321	7251	0.582104	0.537648	5673.5	0.521335	0.478269	5697.5	0.613331
4.07E-07	6172.5	0.840413	1.39E-08	6987	0.8235391	4.07E-07	6624.5	0.7204346	4.07E-07	5688.5	0.769175	0.814384	6829	0.546768	0.488932	5693	0.519814	0.499947	5688.5	0.653999
1.77E-05	5965	0.799142	4.87E-06	6700	0.8776114	1.77E-05	6506	0.782542	2.71E-05	5626	0.716764	0.871493	6628	0.549416	0.688459	5617.5	0.5084	0.664238	5626	0.622736
3.13E-08	6170	0.750252	1.61E-08	6981.5	0.7242695	3.13E-08	6627	0.6199781	1.99E-05	5635	0.76817	0.873004	6951.5	0.559825	0.212277	5837.5	0.5	0.639794	5635	0.615157
3.25E-07	6015	0.748722	9.69E-07	6785	0.7909423	3.25E-07	6903	0.6916728	1.63E-07	5653	0.732286	0.165614	6841	0.522736	0.136008	5900.5	0.486669	0.590766	5653	0.6
8.50E-08	5894.5	0.751844	7.86E-06	6674.5	0.7608108	8.50E-08	6328.5	0.6696676	0.001366	5647	0.712107	0.157216	6904.5	0.5	0.466501	5702.5	0.490687	0.606711	5647	0.599726
1.57E-07	5875	0.720252	7.67E-06	6676	0.6603543	1.57E-07	6430.5	0.6661066	0.011261	5448	0.712381	0.353737	6876	0.456774	1	5513	0.460281	0.803578	5448	0.564664
2.38E-05	5745	0.710665	1.84E-05	6625	0.6433711	2.38E-05	6148.5	0.5423667	0.019622	5643.5	0.703068	0.22148	6611	0.450676	0.466269	5702.5	0.478451	0.616108	5643.5	0.589317
0.258798	5634.5	0.707378	0.0006	6401.5	0.6075785	0.258798	5513	0.5221877	0.023365	5698.5	0.762253	0.470261	5806	0.487582	0.553955	5667	0.494248	0.475748	5698.5	0.51114
0.53706	5535.5	0.679255	0.002135	6308.5	0.5900475	0.53706	5373.5	0.5	0.09404	5534.5	0.74527	0.228364	5673.5	0.506026	0.484926	5694.5	0.473612	0.935511	5534.5	0.506395
0.579805	5513	0.695142	0.011621	6167	0.5832907	0.579805	5405	0.5041088	0.044354	5499.5	0.61943	0.376044	5657	0.500822	0.88052	5552.5	0.47343	0.960028	5499.5	0.5
0.358727	5594.5	0.656245	0.00148	6336.5	0.5025566	0.358727	5234.5	0.5148831	0.234761	5421	0.550383	0.311305	5751.5	0.503196	0.294648	5785.5	0.482743	0.723978	5421	0.490961
0.266498	5604.5	0.658163	0.000669	6393	0.5	0.266498	5277	0.5167093	0.219234	5493	0.500701	0.323252	5801	0.502831	0.458867	5705.5	0.406757	0.940067	5493	0.519175
0.084384	4859	0.6	1	5513	0.4392988	0.084384	5289	0.538057	1	5733.5	0.5	1	5066.5	0.455898	0.43073	5718	0.390321	0.396396	5733.5	0.4979
1	5279	0.664372	0.041671	6040	0.4267714	1	5264	0.4572681	0.174066	5587	0.496238	0.560675	5513	0.432615	0.460624	5705	0.380369	0.776731	5587	0.487126
0.986111	5265	0.585829	0.074986	5974	0.4276844	0.986111	5360.5	0.4547115	0.069895	5472.5	0.484186	0.872997	5508	0.448594	0.832072	5568.5	0.406939	0.877444	5472.5	0.518444
0.518398	5274.5	0.548393	0.08766	5955	0.4243974	0.518398	5095.5	0.4564463	0.305077	5725	0.480716	0.889761	5345.5	0.462655	0.488163	5693.5	0.450767	0.414744	5725	0.571311
0.595583	5533	0.573411	0.00152	6333.5	0.4884039	0.595583	5391.5	0.5036523	0.121173	5866	0.449836	0.869982	5651	0.453068						
0.458877	5572	0.592403	0.002362	6300.5	0.4970782	0.458877	5446	0.5107743	0.050794	5595	0.543809	0.850303	5705.5	0.446311						
0.054526	5761.5	0.5	1.35E-05	6640	0.5407232	0.054526	5468	0.5453798	0.029501	5513	0.505807	0.854796	6011.5	0.452977						
0.228661	5647.5	0.510939	9.86E-05	6521.5	0.5196311	0.228661	5585.5	0.5245617	0.0192	5691.5	0.584167	0.758004	5825.5	0.445946						
0.1658	5667.5	0.515321	0.000556	6407	0.5148831	0.1658	5560.5	0.528214	0.014824	5658	0.563258	0.559245	5872.5	0.434898						
0.016931	5895.5	0.511399	9.79E-06	6660.5	0.5672936	0.016931	5461	0.5698503	0.05348	5617	0.609551	0.996926	6133	0.453433						
0.002324	6049.5	0.580716	5.64E-08	6921	0.5032688	0.002324	5354.5	0.597973	0.087921	5662	0.657122	0.584385	6303.5	0.442476						
0.000497	6058	0.594595	1.64E-07	6868	0.5063733	0.000497	5474	0.5995252	0.045677	5746.5	0.647443	0.944721	6416.5	0.451424						
0.000506	6118	0.610847	2.29E-08	6962.5	0.5135866	0.000506	5521.5	0.6104821	0.01928	5657.5	0.664701	0.232102	6415	0.511505						
0.000142	6215	0.614317	2.02E-09	7069.5	0.5347699	0.000142	5594	0.6281958	0.015748	5721	0.58424	0.261758	6500.5	0.507396						
6.41E-05	6230.5	0.629931	1.13E-09	7093	0.5393353	6.41E-05	5699.5	0.6310263	0.006125	5862	0.588532	0.381303	6550	0.502374						
0.000185	6208.5	0.651844	5.06E-09	7029	0.6292915	0.000185	5795	0.6270088	0.001357	5814	0.576844	0.338965	6483.5	0.500457						
3.04E-05	6290	0.626735	3.14E-10	7146	0.649653	3.04E-05	5677	0.6418919	0.007428	5898	0.59821	0.81888	6595.5	0.472151						
9.98E-05	6238	0.597425	6.72E-10	7114.5	0.6402484	9.98E-05	5561	0.6323959	0.03969	5742.5	0.592458	0.789022	6522.5	0.467951						
0.000153	6202.5	0.594321	1.34E-08	6987.5	0.6260957	0.000153	5459.5	0.6259131	0.046435	5707.5	0.569266	0.451774	6495	0.491417						
0.001071	6118	0.644722	7.96E-08	6905.5	0.6102082	0.001071	5700	0.6104821	0.002229	5783.5	0.654291	0.471232	6362	0.489134						
0.000126	6186	0.588751	2.86E-09	7054	0.6350438	0.000126	5447.5	0.6228999	0.060838	5820.5	0.68141	0.466715	6508.5	0.481099						
1.74E-05	6265	0.59569	2.02E-11	7253	0.6595142	1.74E-05	5460	0.6373265	0.043342	5740	0.61775	0.500727	6628.5	0.495069						
6.29E-06	6308	0.616965	7.66E-12	7288.5	0.5666362	6.29E-06	5583.5	0.645179	0.01354	5615	0.624233	0.777206	6685.5	0.465394						
5.26E-05	6229	0.582907	4.69E-10	7130	0.6436267	5.26E-05	5425	0.6307524	0.080334	6053.5	0.695289	0.853282	6563	0.462838						
6.51E-06	6344	0.639701	3.31E-12	7321	0.6711103	6.51E-06	5650	0.6517531	0.003173	6014	0.630168	0.638457	6683	0.473703						
1.47E-05	6299	0.629748	2.01E-11	7253	0.6610665	1.47E-05	5647	0.6435354	0.006083	6032	0.61775	0.640975	6637.5	0.470325						
0.000559	6097	0.52637	2.27E-08	6962.5	0.6165997	0.000559	5614.5	0.6066472	0.007631	6054	0.664701	0.615124	6408.5	0.477356						
3.04E-05	6288.5	0.597604	3.69E-11	7231	0.5583272	3.04E-05	5540	0.641618	0.013012	6039	0.613733	0.718644	6595.5	0.474617						
0.000386	6095	0.59076	2.86E-08	6952.5	0.6157779	0.000386	5409	0.606282	0.055049	5886	0.662874	0.769891	6434.5	0.473247						

Figure E.1: Non-Parametric Analysis for Multi-Interest Classification: Case 2 Scenario

APPENDIX

F

Non-Parametric Statistical Analysis on Sensitivity Analysis for Multi-Interest Classification: Case 3 Scenario

Ka			Kc			Km			Ksb			Ksm			Kl			Kt		
P-Value	Rank Sum	A-Test	P-Value	Rank Sum	A-Test	P-Value	Rank Sum	A-Test	P-Value	Rank Sum	A-Test	P-Value	Rank Sum	A-Test	P-Value	Rank Sum	A-Test	P-Value	Rank Sum	A-Test
4.42E-19	7489.5	0.718481	4.13E-22	8029.5	0.726413	2.18E-09	7112	0.660939	2.01E-11	5705.5	0.659551	0.463383	7816	0.486943	0.4593	5634.5	0.506026	0.4593	5705.5	0.496987
2.69E-18	7410.5	0.717392	9.93E-21	7942.5	0.770252	1.01E-10	6789	0.646512	5.18E-09	5758	0.643663	0.775598	7744.5	0.47836	0.345718	5757.5	0.522918	0.346847	5758	0.449313
1.11E-17	7390.5	0.723685	1.12E-20	7939.5	0.767421	3.39E-11	7005	0.64286	9.40E-12	5675.5	0.643115	0.484926	7708	0.494522	0.532374	5878	0.546293	0.160347	5675.5	0.436753
3.30E-15	7207.5	0.690997	7.42E-20	7886	0.736925	8.59E-10	6922.5	0.609441	9.03E-10	5589	0.633346	0.88052	7532.5	0.47069	0.771172	5848	0.529492	0.197208	5589	0.445796
5.76E-09	6286	0.676388	2.18E-09	7067.5	0.650931	7.20E-15	6858.5	0.641161	5.76E-09	5639.5	0.683875	0.294648	6620.5	0.498904	0.627152	5737.5	0.520179	0.388037	5639.5	0.452283
3.69E-10	6436.5	0.697389	1.01E-10	7193.5	0.677502	1.39E-08	7816	0.668645	3.69E-10	5698	0.606885	0.458867	7064.5	0.453524	0.476943	5730.5	0.497261	0.298214	5783.5	0.488404
2.87E-06	6398.5	0.621969	3.39E-11	7235.5	0.677593	1.33E-18	7744.5	0.661706	2.87E-06	5702.5	0.614554	0.43073	6943	0.511413	0.466372	5688.5	0.493335	0.236904	5820.5	0.484021
2.85E-06	6381	0.622151	8.59E-10	7106.5	0.670197	9.89E-18	7708	0.65851	2.85E-06	5699	0.610997	0.460624	6856	0.464116	0.474516	5679.5	0.497443	0.382834	5740	0.487947
2.24E-07	6642.5	0.645891	7.20E-15	7536	0.730734	3.31E-17	7532.5	0.706264	2.24E-07	5697.5	0.66943	0.832072	7251	0.482104	0.478269	5831.5	0.513331	0.695665	5615	0.5
4.07E-07	6172.5	0.640413	1.39E-08	6987	0.623539	8.30E-15	6620.5	0.620435	4.07E-07	5688.5	0.669175	0.488163	6829	0.446768	0.499947	5811.5	0.5	0.037478	6053.5	0.519814
2.71E-05	5965	0.699142	4.87E-06	6700	0.577611	2.04E-05	7064.5	0.582542	2.71E-05	5626	0.616764	0.871493	6628	0.449416	0.664238	5825.5	0.502736	0.053647	6014	0.5084
1.99E-05	6170	0.60252	1.61E-08	6981.5	0.62427	2.39E-09	6943	0.619978	1.99E-05	5635	0.66817	0.873004	6951.5	0.459825	0.639794	5756	0.505157	0.045809	6032	0.5063
1.63E-07	6015	0.648722	9.69E-07	6785	0.590942	3.71E-08	6856	0.591673	1.63E-07	5653	0.632286	0.165614	6841	0.522736	0.590766	5706.5	0.500324	0.037208	6054	0.486669
0.001366	5894.5	0.651844	7.86E-06	6674.5	0.560811	2.39E-07	7251	0.569668	0.001366	5647	0.612107	0.157216	6904.5	0.525749	0.606711	5729	0.499726	0.042852	6039	0.490687
0.011261	5875	0.620252	7.67E-06	6676	0.560354	2.28E-11	6829	0.566107	0.011261	5448	0.612381	0.353737	6876	0.501235	0.803578	5513	0.464664	1	5448	0.460281
0.019622	5745	0.610665	1.84E-05	6625	0.543371	4.07E-07	6628	0.542367	0.019622	5643.5	0.603068	0.22148	6611	0.506757	0.616108	5620	0.489317	0.681481	5643.5	0.478451
0.023365	5634.5	0.607378	0.0006	6401.5	0.5	1.77E-05	6951.5	0.522188	0.023365	5698.5	0.662253	0.470261	5806	0.487582	0.475748	5632	0.51114	0.647983	5698.5	0.494248
0.09404	5535.5	0.579255	0.002135	6308.5	0.490047	3.13E-08	6841	0.504109	0.09404	5534.5	0.64527	0.228364	5673.5	0.506026	0.935511	5563.5	0.486395	0.8471	5534.5	0.473612
0.044354	5513	0.5	0.011621	6167	0.483291	3.25E-07	6904.5	0.5	0.044354	5499.5	0.61943	0.376044	5657	0.500822	0.960028	5643	0.481647	0.61748	5499.5	0.47343
0.234761	5757.5	0.556245	0.00148	6336.5	0.502557	8.50E-08	6876	0.514883	0.234761	5421	0.5	0.311305	5751.5	0.5	0.723978	5678	0.490961	0.526091	5421	0.482743
0.219234	5878	0.558163	0.000669	6393	0.432213	0.266498	6611	0.516709	0.219234	5493	0.660701	0.323252	5801	0.502831	0.940067	5805.5	0.519175	0.260242	5493	0.496757
1	5848	0.512454	1	5513	0.339299	0.084384	5806	0.38057	1	5733.5	0.513455	1	5066.5	0.455898	0.396396	5679.5	0.4979	0.522171	5733.5	0.490321
0.174066	5737.5	0.564372	0.041671	6040	0.426771	1	5673.5	0.457268	0.174066	5587	0.596238	0.560675	5513	0.432615	0.776731	5578	0.487126	0.803477	5587	0.480369
0.069895	5730.5	0.585829	0.074986	5974	0.427684	0.986111	5657	0.454711	0.069895	5472.5	0.584186	0.872997	5508	0.448594	0.877444	5791.5	0.518444	0.877444	5472.5	0.506939
0.305077	5688.5	0.548393	0.08766	5955	0.424397	0.518398	5751.5	0.456446	0.305077	5725	0.580716	0.889761	5345.5	0.462655	0.488163	5693.5	0.531311	0.414744	5725	0.500767
0.121173	5679.5	0.573411	0.00152	6333.5	0.488404	2.98E-07	5801	0.503652	0.121173	5866	0.649836	0.869982	5651	0.453068						
0.050794	5831.5	0.592403	0.002362	6300.5	0.497078	9.63E-05	5066.5	0.510774	0.050794	5595	0.643809	0.850303	5705.5	0.446311						
0.029501	5811.5	0.603086	1.35E-05	6640	0.540723	0.000157	5513	0.54538	0.029501	5513	0.705807	0.854796	6011.5	0.452977						
0.0192	5825.5	0.610939	9.86E-05	6521.5	0.519631	3.14E-06	5508	0.524562	0.0192	5691.5	0.684167	0.758004	5825.5	0.445946						
0.014824	5756	0.615321	0.000556	6407	0.514883	1.92E-05	5345.5	0.528214	0.014824	5658	0.663258	0.559245	5872.5	0.434898						
0.05348	5706.5	0.591399	9.79E-06	6660.5	0.567294	0.016931	5651	0.56985	0.05348	5617	0.709551	0.996926	6133	0.453433						
0.087921	5342	0.580716	5.64E-08	6921	0.603269	0.002324	5705.5	0.597973	0.087921	5662	0.757122	0.584385	6303.5	0.442476						
0.045677	5513	0.594595	1.64E-07	6868	0.606373	0.000497	6011.5	0.599525	0.045677	5746.5	0.747443	0.944721	6416.5	0.451424						
0.01928	5342	0.610847	2.29E-08	6962.5	0.613587	0.000506	5825.5	0.610482	0.01928	5657.5	0.764701	0.232102	6415	0.511505						
0.000142	5632	0.614317	2.02E-09	7069.5	0.63477	0.000142	5872.5	0.628196	0.015748	5721	0.78424	0.261758	6500.5	0.507396						
6.41E-05	5563.5	0.629931	1.13E-09	7093	0.639335	6.41E-05	6133	0.631026	0.006125	5862	0.788532	0.381303	6550	0.502374						
0.000185	5643	0.651844	5.06E-09	7029	0.629291	0.000185	6303.5	0.627009	0.001357	5814	0.776844	0.338965	6483.5	0.500457						
3.04E-05	5678	0.626735	3.14E-10	7146	0.649653	3.04E-05	6416.5	0.641892	0.007428	5898	0.79821	0.81888	6595.5	0.472151						
9.98E-05	5805.5	0.597425	6.72E-10	7114.5	0.640248	9.98E-05	6415	0.632396	0.03969	5742.5	0.792458	0.789022	6522.5	0.467951						
0.000153	5679.5	0.594321	1.34E-08	6987.5	0.626096	0.000153	6500.5	0.625913	0.046435	5707.5	0.769266	0.451774	6495	0.491417						
0.001071	5578	0.644722	7.96E-08	6905.5	0.610208	0.001071	6550	0.610482	0.002229	5783.5	0.754291	0.471232	6362	0.489134						
0.000126	5791.5	0.588751	2.86E-09	7054	0.635044	0.000126	6483.5	0.6229	0.060838	5820.5	0.78141	0.466715	6508.5	0.481099						
1.74E-05	6265	0.59569	2.02E-11	7253	0.659514	1.74E-05	6595.5	0.637327	0.043342	5740	0.81775	0.500727	6628.5	0.495069						
6.29E-06	6308	0.616965	7.66E-12	7288.5	0.666636	6.29E-06	6522.5	0.645179	0.01354	5615	0.824233	0.777206	6685.5	0.465394						
5.26E-05	6229	0.582907	4.69E-10	7130	0.643627	5.26E-05	6495	0.630752	0.080334	6053.5	0.795289	0.853282	6563	0.462838						
6.51E-06	6344	0.639701	3.31E-12	7321	0.67111	6.51E-06	6362	0.651753	0.003173	6014	0.830168	0.638457	6683	0.473703						
1.47E-05	6299	0.629748	2.01E-11	7253	0.661066	1.47E-05	5647	0.643535	0.006083	6032	0.81775	0.640975	6637.5	0.470325						
0.000559	6097	0.62637	2.27E-08	6962.5	0.6166	0.000559	5614.5	0.606647	0.007631	6054	0.764701	0.615124	6408.5	0.477356						
3.04E-05	6288.5	0.617604	3.69E-11	7231	0.658327	3.04E-05	5540	0.641618	0.013012	6039	0.813733	0.718644	6595.5	0.474617						
0.000386	6095	0.59076	2.86E-08	6952.5	0.615778	0.000386	5409	0.606282	0.055049	5886	0.762874	0.769891	6434.5	0.473247						

Figure F.1: Non-Parametric Analysis for Multi-Interest Classification: Case 3 Scenario

APPENDIX

G

The Java WordNet Library (JWNL) Interface

```
<?xml version="1.0" encoding="UTF-8"?>
- <jwnl_properties language="en">
    <version language="en" number="2.0" publisher="Princeton"/>
  - <dictionary class="net.didion.jwnl.dictionary.FileBackedDictionary">
    - <param value="net.didion.jwnl.dictionary.morph.DefaultMorphologicalProcessor" name="morphological_processor">
      - <param name="operations">
          <param value="net.didion.jwnl.dictionary.morph.LookupExceptionsOperation"/>
        - <param value="net.didion.jwnl.dictionary.morph.DetachSuffixesOperation">
            <param value="|s=|ses=s|xes=x|zes=z|ches=ch|shes=sh|men=man|ies=y|" name="noun"/>
            <param value="|s=|ies=y|es=e|es=|ed=e|ed=|ing=e|ing=|" name="verb"/>
            <param value="|er=|est=|er=e|est=e|" name="adjective"/>
          - <param name="operations">
              <param value="net.didion.jwnl.dictionary.morph.LookupIndexWordOperation"/>
              <param value="net.didion.jwnl.dictionary.morph.LookupExceptionsOperation"/>
            </param>
          </param>
        - <param value="net.didion.jwnl.dictionary.morph.TokenizerOperation">
          - <param name="delimiters">
              <param value=" "/>
              <param value="-"/>
            </param>
          - <param name="token_operations">
              <param value="net.didion.jwnl.dictionary.morph.LookupIndexWordOperation"/>
              <param value="net.didion.jwnl.dictionary.morph.LookupExceptionsOperation"/>
            - <param value="net.didion.jwnl.dictionary.morph.DetachSuffixesOperation">
                <param value="|s=|ses=s|xes=x|zes=z|ches=ch|shes=sh|men=man|ies=y|" name="noun"/>
                <param value="|s=|ies=y|es=e|es=|ed=e|ed=|ing=e|ing=|" name="verb"/>
                <param value="|er=|est=|er=e|est=e|" name="adjective"/>
              - <param name="operations">
                  <param value="net.didion.jwnl.dictionary.morph.LookupIndexWordOperation"/>
                  <param value="net.didion.jwnl.dictionary.morph.LookupExceptionsOperation"/>
                </param>
              </param>
            </param>
          </param>
        </param>
      </param>
      <param value="net.didion.jwnl.princeton.data.PrincetonWN17FileDictionaryElementFactory" name="dictionary_element_factory"/>
    - <param value="net.didion.jwnl.dictionary.file_manager.FileManagerImpl" name="file_manager">
        <param value="net.didion.jwnl.princeton.file.PrincetonRandomAccessDictionaryFile" name="file_type"/>
        <param value="c:\program files (x86)\wordnet\2.0\dict" name="dictionary_path"/>
      </param>
    </dictionary>
```

Figure G.1: The Java WordNet Library (JWNL) used to create an interface between ProAdDCS and WordNet

APPENDIX

H

List of Topic, Thematic Subject and Number of Document Description of the Topics for TREC-2001 Filtering Track

Topic	Subject	**Number of Documents in**	
		Test Set	Training Set
R1	STRATEGY/PLANS	23651	597
R2	LEGAL/JUDICIAL	11563	351
R3	REGULATION/POLICY	36463	821
R4	SHARE LISTINGS	7250	146
R5	ANNUAL RESULTS	22813	352
R6	INSOLVENCY/LIQUIDITY	1871	42
R7	SHARE/CAPITAL	17876	403
R8	BONDS/DEBT ISSUES	11202	251
R9	LOANS/CREDITS	5625	612
R10	CREDIT RATINGS	5625	212
R20	MARKET SHARE	1074	38
R21	ADVERTISING/PROMOTION	2041	39
R29	MONETARY/ECONOMIC	26402	630
R32	CONSUMER PRICES	5492	140
R41	INDUSTRIAL PRODUCTION	1658	35
R50	LEADING INDICATORS	5104	149
R58	ECCOMPETITION/SUBSIDIARY	1991	41
R68	LABOR ISSUES	16770	419
R79	WELFARE/SOCIAL SERVICE	1818	42

Table H.1: List of Topic, Thematic Subject and Number of Document Description of the Topics Involved in the Experiment

References

[1] N. F. M. Azmi, J. Timmis, and F. Polack, "Towards a principled design of bio-inspired solutions to adaptive information filtering," in *15th IEEE Int. Conf. on Engineering of Complex Computer Systems (ICECCS)*, 2010, pp. 315 – 316.

[2] C. Fellbaum, *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[3] U. Hanani, B. Shapira, and P. Shoval, "Information filtering: Overview of issues, research and systems," *User Modelling and User-Adapted Interaction*, vol. 11, pp. 203–259, 2001.

[4] L. N. de Castro and J. Timmis, *Artificial Immune System: A New Computational Intelligence Approach*. Springer, 2002.

[5] S. Stepney, R. Smith, J. Timmis, A. Tyrrell, M. Neal, and A. Hone, "Conceptual frameworks for artificial immune systems," *Unconventional Computing*, vol. 1, no. 3, pp. 315–338, 2006.

[6] J.Timmis, E. Hart, A. Hone, M. Neal, A. Robins, S. Stepney, and A. Tyrrell, "Immuno-engineering," in *2nd IFIP Int. Conf. on Biologically Inspired Collaborative Computing*, 2008, pp. 3–17.

[7] L. N. de Castro and J. Timmis, "Artificial immune systems as a novel soft computing paradigm," *Soft Computing*, pp. 526–544, 2003.

[8] L. N. de Castro and F. J. V. Zuben, "The clonal selection algorithm with engineering applications," in *Conf. on Genetic and Evolutionary Computation*, 2000, pp. 36–37.

[9] A. Secker, A. A. Freitas, and J. Timmis, "AISEC: An artificial immune system for e-mail classification," *Evolutionary Computation*, vol. 1, pp. 131–138, 2003.

[10] N. Prattipati and E. Hart, "Evaluation and extension of the AISEC e-mail classification system," in *Int. Conf. on Artificial Immune Systems*, 2008, pp. 154–165.

[11] N. Nanas and A. de Roeck, "A review of evolutionary and immune-inspired information filtering," *Natural Computing*, 2009.

[12] J. Kim and P. J. Bentley, "Towards an artificial immune system for network intrusion detection: An investigation of dynamic clonal selection," in *Congress on Evolutionary Computation (CEC)*, 2002, pp. 1015–1020.

[13] D. R. Tauritz and I. G. Sprinkhuizen-Kuyper, "Adaptive information filtering algorithms," in *Advances in Intelligent Data Analysis*, 1999, pp. 513–524.

[14] C. Lanquillon and I. Renz, "Adaptive information filtering: Detecting changes in text streams," in *Int. Conf. on Information and Knowledge Management*, 1999, pp. 538–544.

[15] J. Mostafa, S. Mukhopadhyay, W. Lam, and M. Palakal, "A multilevel approach to intelligent information filtering: Model, system and evaluation," *ACM Transaction on Information Systems*, vol. 15, no. 4, pp. 368–399, 1997.

[16] N. Nanas and A. de Roeck, "Multimodal dynamic optimization: From evolutionary algorithms to artificial immune systems," in *Int. Conf. on Artificial Immune Systems*, 2007, pp. 13–24.

[17] A. A. Freitas and J. Timmis, "Revisiting the foundations of artificial immune systems for data mining," *IEEE Transaction on Evolutionary Computation*, pp. 521 – 540, 2007.

[18] D. L. Chao and S. Forrest, "Information immune system," *Genetic Programming and Evolvable Machines*, vol. 4, pp. 311–331, 2003.

[19] J. M. Jose, H. Joho, and C. van Rijsbergen, "Adaptive information retrieval," *Information Processing and Management*, vol. 44, pp. 1819–1833, 2008.

[20] A. Secker, A. A. Freitas, and J. Timmis, "AISIID: An artificial immune system for interesting information discovery on the web," *Applied Soft Computing*, vol. 8, no. 8, pp. 885–905, 2008.

[21] N. Nanas, A. de Roeck, and V. Uren, "Immune-inspired adaptive information filtering," in *Int. Conf. on Artificial Immune Systems*, 2006, pp. 389–394.

[22] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., 1986.

[23] J. Rocchio, "Relevance feedback in information retrieval," in *SMART Retrieval System-Experiments in Automatic Document Processing*, 1971, p. 313323.

[24] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave information tapestry," *Communication of the ACM*, vol. 35, no. 12, pp. 61–70, 1992.

[25] D. W. Oard and G. Marchionini, "A conceptual framework for text filtering," University of Maryland, Tech. Rep., 1996.

[26] K. Aas, "A survey on personalised information filtering systems for the World Wide Web," in *Int. Conf. on Advances in Infrastructure for e-Business, e-Education, e-Science, and e-Medicine on the Internet*, 1997.

[27] N. Nanas, "Information filtering for knowledge management," The Open University, Tech. Rep., 2001.

[28] N. J. Belkin and W. B. Croft, "Information filtering and information retrieval: Two side of the same coin," *Communication of the ACM*, pp. 29–38, 1992.

[29] M. Höfferer, B. Knaus, and W. Winiwarter, "Adaptive information extraction from online messages," in *Conf. on Intelligent Multimedia Information Systems and Management*, 1994, pp. 314–327.

[30] N. J. Belkin, "Helping people find what they don't know," *Communication of the ACM*, pp. 58–61, 2000.

[31] T. W. Yan and H. Garcia-Molina, "The SIFT information dissemination system," *ACM Transactions on Database Systems*, pp. 529–565, 1999.

[32] T. W. Malone, K. R. Grant, F. A. Turbak, S. A. Brobst, and M. D. Cohen, "The information Lens: An intelligent system for information sharing in organizations," in *SIGCHI Conf. on Human Factors in Computing Systems*, 1986, pp. 1–8.

[33] M. Morita and Y. Shinoda, "Information filtering based on user behavior analysis and best match text retrieval," in *17th Ann. Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 1994, pp. 272–281.

[34] J. M. Yanga and K. F. Li, "Recommendation based on rational inferences in collaborative filtering," *Knowledge-Based Systems*, pp. 105–114, 2009.

[35] D. M. Pennock, E. Horvitz, S. Lawrence, and C. L. Giles, "Collaborative filtering by personality diagnosis: A hybrid memory and model-based approach," in *16th Conf. on Uncertainty in Artificial Intelligence (UAI-2000)*, 2000, pp. 473–480.

[36] R. Kass and T. Finin, "Rules for the implicit acquisition of knowledge about the user," in *Proc. of the AAAI*, 1987, pp. 295–300.

[37] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Machine Learning*, vol. 23, pp. 69–101, 1996.

[38] J. Scanlan, J. Hartnett, and R. Williams, "DynamicWEB: Adapting to concept drift in COBWEB," in *21st Australasian Joint Conf. on Artifical Intelligence*, 2008.

[39] H. Lieberman and A. Letizia, "Letizia: An agent that assists web browsing," in *Int. Joint Conf. of Artificial Intelligence*, 1995.

[40] T. Joachims, D. Freitag, and T. Mitchell, "Webwatcher: A tour guide for the world wide web," in *15th Int. Joint Conf. on Artificial Intelligence*, 1997, pp. 770–777.

[41] M. Pazzani, J. Muramatsu, and D.Billsus, "Syskill and webert: Identifying interesting web sites," in *National Conf. on Artificial Intelligence*, 1996.

[42] J. Fink, A. Kobsa, and A. Nill, "User-oriented adaptivity and adaptability in the AVANTI project," *Designing for the Web: Empirical Studies*, 1996.

[43] A. Wexelblat and P. Maes, "Footprints: History-rich web browsing," in *Conf. on Computer-Assisted Information Retrieval (RIAO)*, 1997, pp. 75–84.

[44] N. Nanas, V. S. Uren, and A. de Roeck, "Nootropia: A user profiling model based on a self-organising term network," in *Int. Conf. on Artificial Immune Systems*, 2004, pp. 146–160.

[45] D. H. Widyantoro, T. R. Ioerger, and J. Yen, "An adaptive algorithm for learning changes in user interests," in *8th Int. Conf. on Information and Knowledge Management*, 1999, pp. 405–412.

[46] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[47] Y. W. Seo and B. T. Zhang, "A reinforcement learning agent for personalized information filtering," in *5th Int. Conf. on Intelligent User Interfaces*, 2000, pp. 248–251.

[48] B. T. Zhang and Y. W. Seo, "Personalized web document filtering using reinforcement learning," *Applied Artificial Intelligence*, vol. 15, pp. 665–685, 2001.

[49] D. R. Tauritz and I. G. Sprinkhuizen-Kuyper, "Adaptive information filtering using evolutionary computation," *Information Sciences*, vol. 122, pp. 121–140, 2000.

[50] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, pp. 59–69, 1982.

[51] T. Kohonen, S. Kaski, K. Lagus, J. Salojrvi, J. Honkela, V. Paatero, and A. Saarela, "Self organization of a massive document collection," *IEEE Transaction on Neural Network*, vol. 11, pp. 574–585, 2000.

[52] C. Ding, J. C. Patra, and F. C. Peng, "User modeling for personalized web search with self-organizing map," *American Society for Information Science and Technology*, pp. 494–507, 2007.

[53] C. Lemnaru, A. A. Firte, and R. Potolea, "Static and dynamic user type identification in adaptive e-learning," in *IEEE Int. Conf. on Intelligent Computer Communication and Processing (ICCP)*, 2011, pp. 11–18.

[54] D. O. Hebb, *The Organizational Behaviour*. John Wiley and Sons, 1949.

[55] Q. Chen, J. Jin, and H. Chen, "A dynamic hebbian learning algorithm for constructing e-learner communities," in *5th Int. Conf. on Natural Computation (ICNC)*, 2009, pp. 3–6.

[56] M. McElligott and H. Sorensen, "An emergent approach to information filtering," *UCC Computer Science*, 1993.

[57] M. M. Elligott and H. Sorensen, "An evolutionary connectionist approach to personal information filtering," in *Int. Conf. on Neural Network (INNC)*, 1994, pp. 141–146.

[58] J. Bollen and F. Heylighen, "Dynamic and adaptive structuring of the world wide web," in *Flexible Hypertext Workshop*, 1997, pp. 13–17.

[59] R. A. Goldsby, T. J. Kindt, B. A. Osborne, and J. Kuby, *Immunology*. W.H. Freeman, 2003.

[60] Y. Ishida, *Immunity-Based System: A Design Perspective*. Springer, 2004.

[61] I. R. Cohen, "Discrimination and dialogue in the immune system," *seminars in Immunology*, vol. 12, pp. 215–219, 2000.

[62] ——, *Tending Adam's Garden: Evolving the Cognitive Immune Self*. Academic Press, 2000.

[63] H. Atlan and I. R. Cohen, "Immune information, self-organization and meaning," *International Immunology*, vol. 10, no. 6, pp. 711–717, 1998.

[64] C. Janeway, *Immunobiology: The Immune System in Health and Disease*. Garland Science, 2005.

[65] S. A. Hofmeyr, "An interpretative introduction to the immune system," in *Design Principles for the Immune System and Other Distributed Autonomous Systems*, 2000.

[66] S. A. Hofmeyr and S. Forrest, "Architecture for an artificial immune system," *Evolutionary Computation*, vol. 8, no. 4, pp. 443–473, 2000.

[67] D. Dasgupta and F. Gonzalez, "Artificial immune system in intrusion detection," *Enhancing Computer Security With Smart Technology*, pp. 165–208, 2006.

[68] R. Medzhitov and C. A. Janeway, "How does the immune system distinguish self from nonself?" *seminars in Immunology*, pp. 185–188, 2000.

[69] C. A. Janeway and R. Medzhitov, "Innate immune recognition," *Annual Review Immunology*, 2002.

[70] J. D. Farmer, N. H. Packard, and A. S. Perelson, "The immune system, adaptation and machine learning," *Physica 22D*, pp. 187–204, 1986.

[71] D. Dasgupta, *Artificial Immune Systems and Their Applications*. Springer-Verlag, 1998.

[72] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri, "Self-nonself discrimination in a computer," in *IEEE Computer Society Symposium on Research in Security and Privacy*, 1994, pp. 202–212.

[73] S. Forrest, S. A. Hofmeyr, and A. Somayaji, "Computer immunology," *Communication of the ACM*, pp. 88–96, 1997.

[74] D. Dasgupta, "Immunity-based intrusion detection system: A general framework," in *22nd Nat. Information Systems Security Conf.*, 1999, pp. 147–160.

[75] P. D. Williams, K. P. Anchor, J. L. Bebo, G. H. Gunsch, and G. B. Lamont, "CDIS: Toward a computer immune system for detecting network intrusions," in *Fourth Int. Symp. Recent Advances in Intrusion Detection*, 2001, p. 117133.

[76] R. Greaves, "Computational modelling of Treg networks in experimental autoimmune encephalomyelitis," Master's thesis, Department of Computer Science, University of York, 2011.

[77] L. Albergante, "A petri net model of liver response to visceral leishmaniasis: Self-regulation and complex interplay in the vertebrate immune system," Ph.D. dissertation, Department of Mathematics, Universit degli Studi di Milano, Italy, 2010.

[78] Y. Liu, "A neuro-immune inspired computational framework and its applications to a machine visual tracking system," Ph.D. dissertation, Department of Electronics, University of York, 2009.

[79] P. S. Andrews and J. Timmis, "Inspiration for the next generation of artificial immune systems," in *Int. Conf. on Artificial Immune Systems*, ser. 3627, 2005, pp. 126–138.

[80] P. S. Andrews, "An investigation of a methodology for the development of artificial immune systems: A case-study in immune receptor degeneracy," Ph.D. dissertation, Department of Computer Science, University of York, 2009.

[81] M. Read, "Statistical and modelling techniques to build confidence in the investigation of immunology through agent-based simulation." Ph.D. dissertation, Department of Computer Science, University of York, 2011.

[82] S. Stepney, R. Smith, J. Timmis, A. Tyrrell, M. Neal, and A. Hone, "Conceptual frameworks for artificial immune systems," in *3rd Int. Conf. on Artificial Immune Systems*, 2004, p. 5364.

[83] J. Twycross and U. Aickelin, "Towards a conceptual framework for innate immunity," in *Int. Conf. on Artificial Immune Systems*, 2005, pp. 112–125.

[84] H. Bersini, "Immune system modeling: The OO way," in *Int. Conf. on Artificial Immune Systems*, 2006, pp. 150–163.

[85] I. R. Cohen, "Real and artificial immune systems: Computing the state of the body," *Nature Review Immunology*, pp. 569–574, 2007.

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

[87] N. K. Jerne, "Towards a network theory of the immune system," *Annual Immunology*, vol. 125C, no. 1-2, pp. 373–389, 1974.

[88] P. Matzinger, "An innate sense of danger," *Seminars in Immunology*, pp. 399–415, 1998.

[89] ——, "The danger model: A renewed sense of self," *Science Magazine*, pp. 301–305, 2002.

[90] Z. Grossman and W. E. Paul, "Adaptive cellular interactions in the immune system: The tunable activation threshold and the significance of subthreshold responses," in *National Academy of Science*, 1992, p. 1036510369.

[91] J. Kim and P. Bentley, "A model of gene libraries evolution in the dynamic clonal selection algorithm," in *Int. Conf. on Artificial Immune Systems*, 2002, pp. 182–189.

[92] L. N. de Castro and F. J. V. Zuben, "aiNet: An artificial immune network for data analysis," in *Data Mining: A Heuristic Approach*, 2001, pp. 231–259.

[93] J. Timmis and M. Neal, "A resource limited artificial immune system for data analysis," *Knowledge Based Systems*, pp. 121–130, 2001.

[94] J. Timmis, "Artificial immune systems: A novel data analysis technique inspired by the immune network theory," Ph.D. dissertation, Department of Computer Science, University of Wales, 2000.

[95] J. Greensmith, U. Aickelin, and S. Cayzer, "Introducing dendritic cells as a novel immune-inspired algorithm for anomaly detection," *4th Int. Conf. on Artificial Immune Systems*, pp. 153–167, 2005.

[96] N. D. L. Owens, J. Timmis, A. Greensted, and A. Tyrrell, "Modelling the tunability of early t cell signaling events," in *Int. Conf. on Artificial Immune Systems*, 2008, pp. 12–23.

[97] A. S. Perelson and G. F. Oster, "Theoretical studies of clonal selection: Minimal antibody repertoire size and reliability ofself non-self discrimination," *Journal of Theoretical Biology*, vol. 81, pp. 645–670, 1979.

[98] C. McEwan, E. Hart, and B. Paechter, "Boosting the immune system," in *Int. Conf. on Artificial Immune Systems*, 2008.

[99] R. E. Schapire, "A brief introduction to boosting," in *Proc. of the 16th Int. Conf. on Artificial Intelligence*, 1999, pp. 1–6.

[100] A. A. Freitas and J. Timmis, "Revisiting the foundation of artificial immune systems: A problem-oriented perspective," in *Proc. of 2nd Int. Conf. on Artificial Immune Systems (ICARIS)*, 2003, pp. 249–260.

[101] E. Hart and J. Timmis, "Application areas of AIS: The past, the present and the future," *Applied Soft Computing*, vol. 8, no. 1, pp. 191–201, 2008.

[102] J. Timmis, P. Andrews, N. Owen, and E. Clark, "An interdisciplinary perspective on artificial immune system," *Evolutionary Intelligence*, pp. 5–26, 2008.

[103] C. McEwan and E. Hart, "Representation in the (Artificial) Immune System," *Mathematical Modelling and Algorithms*, pp. 125–149, 2009.

[104] S. Cayzer and U. Aickelin, "A recommender system based on idiotypic artificial immune networks," *Mathematical Modelling and Algorithms*, vol. 4, pp. 181–198, 2005.

[105] D. A. Kimbrell and B. Beutler, "The evolution and genetics of innate immunity," *Nature Reviews: Genetics*, vol. 2, pp. 256–267, 2001.

[106] L. V. Parijs and A. K. Abbas, "Homeostasis and self-tolerance in the immune system: Turning lymphocytes off," *Science*, vol. 280, pp. 243–248, 1998.

[107] J. T. Opferman and S. J. Korsmeyer, "Apoptosis in the development and maintenance of the immune system," *Nature Immunology*, vol. 4, no. 5, pp. 410–415, 2003.

[108] J. A. M. Borghans and R. J. D. Boer, "Diversity in the immune system," in *Design Principle for the Immune System and Other Distributed Autonomous Systems*. OUP, 2001, pp. 161–183.

[109] I. R. Cohen, "The creation of immune specificity," in *Design Principle for the Immune System and Other Distributed Autonomous Systems*. OUP, 2001, pp. 151–159.

[110] G. M. Edelman and J. A. Gally, "Degeneracy and complexity in biological systems," *National Academy of Sciences of the United States of America*, vol. 98, no. 24, pp. 13 763–13 768, 2001.

[111] M. Neal and B. C. Trapnell, "Go dutch: Exploit interactions and environments with artificial immune systems," in *In Silico Immunology*, J. Timmis and D. Flower, Eds. Springer, 2007, pp. 313–330.

[112] D. Keil and D. Goldin, "Modeling indirect interaction in open computational systems," in *Int. Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*. IEEE, 2003, pp. 371– 376.

[113] P. Maes, "Agents that reduce work and information overload," *Communications of the ACM*, vol. 37, no. 7, pp. 30–40, 1994.

[114] B. M. Sarwar, "Sparsity, scalability, and distribution in recommender systems," Ph.D. dissertation, University of Minnesota, 2001.

[115] M. Neal, "Meta-stable memory in an artificial immune network," in *Int. Conf. on Artificial Immune Systems*, 2003, pp. 168–180.

[116] J. Kim and P. J. Bentley, "Immune memory and gene library evolution in the dynamic clonal selection algorithm," *Genetic Programming and Evolvable Machines*, vol. 5, pp. 361–391, 2004.

[117] S. Cayzer, J. Smith, J. A. R. Marshall, and T. Kovacs, "What have gene libraries done for AIS?" in *Int. Conf. on Artificial Immune Systems*, 2005, pp. 86–99.

[118] S. Cayzer and J. Smith, "Gene libraries: Coverage, efficiency and diversity," in *Int. Conf. on Artificial Immune Systems*, 2006, pp. 136–149.

[119] L. N. de Castro and F. J. V. Zuben, "Learning and optimization using the clonal selection principle," *IEEE Transactions on Evolutionary Computation*, pp. 239–251, 2002.

[120] J. Kim and P. Bentley, "Immune memory in the dynamic clonal selection," in *Int. Conf. on Artificial Immune Systems*, 2002, pp. 59–67.

[121] J. Kelsey and J. Timmis, "Immune inspired somatic contiguous hypermutation for function optimisation," in *Int. Conf on Genetic and Evolutionary Computation (GECCO)*, 2003, pp. 207–218.

[122] R. Bekkerman, "Automatic categorization of email into folders: Benchmark experiments on enron and sri corpora," University of Massachusetts - Amherst, Tech. Rep., 2004.

[123] D. Zhou, E. Manavoglu, J. Li, C. L. Giles, and H. Zha, "Probabilistic models for discovering e-communities," in *Proc. of the 15th Int. Conf. on World Wide Web (WWW)*, 2006, pp. 173 – 182.

[124] A. McCallum, A. Corrada-Emmanuel, and X. Wang, "Topic and role discovery in social networks with experiments on enron and academic email," *Journal of Artificial Intelligence Research*, vol. 30, pp. 249–272, 2007.

[125] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, p. 8083, 1945.

[126] A. Vargha and H. D. Delaney, "A critique and improvement of the common language effect size statistics of mcgraw and wong," *Journal on Educational and Behavioral Statistics*, vol. 25, no. 2, p. 101132, 2000.

[127] B. W. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochimica et Biophysica Acta*, vol. 405, p. 442451, 1975.

[128] J. W. Tukey, *Exploratory Data Analysis*. Addison Wesley, 1977.

[129] P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen, and H. Nielsen, "Assessing the accuracy of prediction algorithms for classification: An overview," *Bioinformatics*, pp. 412–424, 2000.

[130] Matlab, *MATLAB Documentation*, 2012.

[131] N. F. M. Azmi, F. Polack, and J. Timmis, "Immune inspired adaptive information filtering: Focusing on profile adaptation," in *Bio-Inspired Models of Networks, Information and Computing Systems*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, 2011, vol. 103, pp. 242–247.

[132] N. Nanas, M. Vavalis, and L. Kellis, "Immune learning in a dynamic information environment," in *8th Int. Conf. on Artificial Immune Systems*, 2009, pp. 192–205.

[133] T. Mitchell, *Machine Learning*. Mc Graw Hill, 1977.

[134] Y. H. Li and A. K. Jain, "Classification of text documents," *The Computer Journal*, 2000.

[135] T. S. Guzella and W. M. Caminhas, "A review of machine learning approaches to spam filtering," *Expert Systems with Applications*, vol. 36, pp. 10 206–10 222, 2009.

[136] Y. Song, A. Kolcz, and C. L. Giles, "Better naive bayes classification for high-precision spam detection," *Software: Practive and Experience*, vol. 39, pp. 1003–1024, 2009.

[137] S. Haykin, *Neural networks: A Comprehensive Foundation*. Prentice Hall, 1997.

[138] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, pp. 179–211, 1990.

[139] C. J. van Rijsbergen, *Information Retrieval*. Butterworths, 1979.

[140] ——, "A new theoretical framework for information retrieval," in *ACM SIGIR Conf. on Research and Development in Information Retrieval*, vol. 21, 1986, pp. 23–29.

[141] A. Saltelli, K. Chan, and E. M. Scott, *Sensitivity Analysis*. John Wiley and Sons, 2000.

[142] H. C. Frey and S. R. Patil, "Identification and review of sensitivity analysis methods," *Risk Analysis*, vol. 22, no. 3, pp. 553–578, 2002.

[143] J. T. Yao, "Sensitivity analysis for data mining," in *22nd International Conference of NAFIPS, Chicago, USA*, 2012, pp. 272–277.

[144] H. R. Maturana and F. J. Varela, *Autopoiesis and Cognition: The Realization of the Living*. Springer, 1980.

[145] J. Timmis, "Artificial immune systems : Today and Tomorrow," *Natural Computing*, vol. 6, pp. 1–18, 2007.

[146] B. H. Ulutas and S. K. Konak, "A review of clonal selection algorithm and its applications," *Artificial Intelligence Review*, vol. 36, pp. 117–138, 2011.

[147] N. F. M. Azmi, J. Timmis, and F. Polack, "Profile adaptation in adaptive information filtering: An immune inspired approach," in *IEEE Int. Conf. on Soft Computing and Pattern Recognition (SoCPaR)*, 2009, pp. 414–419.

[148] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[149] A. A. Haidar and L. M. Rocha, "Adaptive spam detection inspired by a cross-regulation model of immune dynamics: A study of concept drift," in *Int. Conf. on Artificial Immune Systems*, 2008, pp. 36–47.

[150] ——, "Adaptive spam detection inspired by the immune system," in *Artificial Life XI Int. Conf. on the Simulation and Synthesis of Living System*. MIT Press, 2008, pp. 1–8.

[151] T. S. Guzella, T. A. Mota-Santos, J. Uchoa, and W. M. Caminhas, "Identification of spam messages using an approach inspired on the immune system," *BioSystems*, vol. 92, pp. 215–225, 2008.

[152] T. Oda and T. White, "Developing an immunity to spam," in *Conf. on Genetic and Evolutionary Computation (GECCO)*, 2003, pp. 231–242.

[153] ——, "Increasing the accuracy of a SPAM-detecting artificial immune system," in *Congress on Evolutionary Computation*, 2003, pp. 390–396.

[154] ——, "Immunity form spam: An analysis of an artificial immune system for junk email detection," in *Int. Conf. on Artificial Immune System*, 2005, pp. 276–289.

[155] J. Sobecki and L. Szczepanski, "Wiki-News interface agent based on AIS method," *Agents and Multi-Agent Systems: Technologies and Applications*, pp. 258–266, 2007.

[156] S. Sarafijanovic and J. Y. L. Boudec, "Artificial immune system for collaborative spam," *Studies in Computational Intelligence*, vol. 129, pp. 39–51, 2008.

[157] D. L. Chao and S. Forrest, "Generating biomorphs with an aesthetic immune system," in *Artificial Life VIII Int. Conf. on the Simulation and Synthesis of Living System*. MIT Press, 2002, pp. 89–92.

[158] A. M. Acilar and A. Arslan, "A collaborative filtering method based on artificial immune network," *Expert System with Applications*, vol. 36, pp. 8324–8332, 2008.

[159] X. Yue, A. Abraham, Z. X. Chi, Y. Y. Hao, and H. Mo, "Artificial immune system inspired behaviour-based anti-spam filter," *Soft Computing*, vol. 11, pp. 729–740, 2007.

[160] X. Yue and L. Quan-Zhong, "Immune-inspired collaborative technology for rating-based recommendation system," in *Int. Conf. on Network and Parallel Computing*, 2007, pp. 897–902.

[161] P. A. D. de Castro, F. O. de Franca, H. M. Ferreira, and F. J. V. Zuben, "Applying biclustering to perform collaborative filtering," in *Int. Conf. on Intelligent Systems Design and Applications*, 2007, pp. 421–426.

[162] ——, "Evaluating the performance of a biclustering algorithm applied to collaborative filtering a comparative analysis," in *Int. Conf. on Intelligent Systems Design and Applications*, 2007, pp. 65–70.

[163] G. B. Bezerra, T. V. Barra, H. M. Ferreira, H. Knidel, L. N. de Castro, and F. J. V. Zuben, "An immunological filter for spam," in *Int. Conf. on Artificial Immune Systems*, 2006, pp. 446–458.