Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Simen Sverdrup-Thygeson

An Artificial Immune System for Fake News Classification

Master's thesis in Computer Science
Supervisor: Pauline Catriona Haddow
June 2021

NTNU
Norwegian University of
Science and Technology

Simen Sverdrup-Thygeson

An Artificial Immune System for Fake News Classification

Master’s thesis in Computer Science
Supervisor: Pauline Catriona Haddow
June 2021

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Simen Sverdrup-Thygeson

An Artificial Immune System for Fake News Classification

Master's Thesis, Spring 2021

Artificial Intelligence Group
Department of Computer and Information Science
Faculty of Information Technology, Mathematics and Electrical Engineering

NTNU

Abstract

An explosive growth of misleading and untrustworthy news articles has been observed over the last years. These news articles are often referred to as *fake news* and have been found to severely impact fair elections and democratic values. As such, the need for accurate, adaptive and computationally effective classification models is clear.

The biological immune system is a network of biological processes which protects its host from foreign pathogens, distinguishing them from the host's own cells. The immune system is inherently adaptive, self-organising and robust, which has inspired several computational classification models. Such models are known as Artificial Immune Systems (AIS), which seek to utilize the underlying principles and properties of the biological immune system to produce similar levels of effectiveness on computational applications. One such application is e-mail spam detection, for which the nature of immune systems is inherently suited. Such applications, which apply AIS models to text classification, have shown promising potential. This thesis investigates whether an Artificial Immune System could be applied to the classification of fake news articles with similar success as found for e-mail spam detection.

An AIS fake news classification model was designed, based on various existing models – tuned to the needs of fake news classification. Further, comprehensive feature extraction strategies were implemented and analyzed, based on the linguistic writing styles of the news articles. Finally, the model was tested on three distinct fake news datasets. Notably, the accuracy scores obtained beat those of several state of the art models, suggesting there is considerable potential in this AIS application.

Sammendrag

Det har blitt observert en eksplosiv vekst av upålitelige nyhetsartikler de siste årene. Disse nyhetsartiklene blir ofte referert til som "fake news" og det har blitt vist at disse kan ha alvorlige innvirkninger på demokratiske prosesser. Behovet for nøyaktige, adaptive og effektive filtreringsmodeller har dermed blitt mer og mer tydelig.

Det biologiske immunforsvaret består av naturlige prosesser som beskytter kroppen mot fremmede smittestoffer. Disse smittestoffene kan være virus, bakterier, sopp eller parasitter. En sentral del av dette er å skille disse fremmedstoffene fra kroppens egne celler, et problem som er tett knyttet til klassifisering. Videre er immunforsvaret kjent for å være både tilpasningsdyktig, selvorganiserende og robust, noe som har inspirert mange forskere til å hente inspirasjon fra immunforsvaret når de designer klassifiseringsmodeller. Slike modeller er ofte kjent som kunstige immunsystemer (Artificial Immune System, AIS). Disse forsøker å etterape de bakenforliggende prosessene til biologiske immunsystemer, for å oppnå lignende grader av effektivitet. Et kjent applikasjonsdomene er filtrering av spam e-post, hvor AIS-modeller har oppnådd lovende resultater. Denne oppgaven har vært rettet mot å undersøke om et kunstig immunsystem kan brukes på klassifisering av *fake news*, med samme suksess som for e-post spam.

En klassifiseringsmodell ble designet fra bunnen av, basert på diverse eksisterende AIS-modeller. Modelldesignet har vært vridd spesifikt opp imot *fake news* klassifisering. Videre ble omfattende uthentingsstrategier implementert, for å hente ut tall (som videre kan brukes til klassifisering) fra nyhetsartiklene. Disse strategiene er basert på skrivestilen som er brukt i nyhetsartiklene. Til slutt ble modellen testet på tre ulike *fake news* datasett. De oppnådde resultatene er kompetitive med flere moderne klassifiseringsmodeller, selv om modellen har noe problemer med å forbedre nøyaktigheten over tid. Likevel indikerer disse resultatene at å bruke en AIS-modell på dette applikasjonsdomenet har betydelig potensial.

Preface

The following thesis is the result of a research conducted at the Norwegian University of Science and Technology in Trondheim, Norway. The work was conducted during the period of 15.01.2021 - 11.06.2021, as part of the M.Sc. degree in Computer Science.

I would like to thank Pauline Catriona Haddow for her excellent guidance throughout the project. Her late nights and weekends dedicated to reading through and commenting on my work has been invaluable and I am truly grateful to have had her as my supervisor.

I would also like to thank Eirik Baug and Andreas Norstein for great tips and input related to the MAIM algorithm. Additionally, thanks are extended to the bio-inspired computing research group CRAB, for intriguing discussions surrounding a variety of biologically-inspired artificial intelligence topics. Finally, I would like to thank my roommates, who made writing this thesis from home considerably less tedious.

Simen Sverdrup-Thygeson Trondheim, June 11, 2021

Abbreviations

NTNU = Norwegian University of Science and Technology

M.Sc. = Master of Science

ML = Machine Learning

NLP = Natural Language Processing

EA = Evolutionary Algorithm

ANN = Artificial Neural Network

AIS = Artificial Immune System

RR = Recognition Region

BoW = Bag-of-Words

BERT = Bidirectional Encoder Representations

ELMo = Embeddings from Language Models

TF = Term Frequency

TF-IDF = Term Frequency-Inverse Document Frequency

TE = Text Embedding

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter aims to introduce the motivation behind the selected research topic, as well as the identified research goals. Further, the literature review protocol, research methodology and thesis structure are presented.

1.1 Motivation

Over the last few years, an explosive growth of disinformation and untrustworthy news articles on digital media has been observed [Meel and Vishwakarma, 2019]. These misleading news articles are generally known as *fake news* and have been reported to erode public trust, freedom of speech and democracy [Zhou and Zafarani, 2018]. In a 2017 poll, 64% of U.S adults reported that fake news articles had caused them considerable confusion regarding the truthfulness of recent events [Volkova et al., 2017]. Additionally, the nature of modern social media tend to reinforce and amplify the false and/or biased information, often referred to as the *Echo Chamber Effect* [Jamieson and Cappella, 2008].

One of the most illustrative examples of the impact of fake new articles was the months leading up to the 2016 U.S. presidential selection. Reportedly, the fake news engagements on Facebook (defined as the total number of comments, shares and reactions) were 20% higher than that of mainstream news articles, throughout the final months leading up to election day [Berghel, 2017]. It is reasonable to assume that this massive engagement, to some extent, had an impact on the result of the presidential election. The skewed engagement ratio for deceitful news articles aligns well with research showing that fake news generally are retweeted by more users, and therefore spread far more swiftly than real news. This effect is observed to be even greater when the news articles are of a political nature [Vosoughi et al., 2018].

Needless to say, this problem has created a need for effective filtering and classification models, to accurately separate the fake and fact-based news articles. However, this has proven to be challenging, due to the nature of the fake news and their similarity to real news articles. Several approaches has been suggested, often combining the fields of linguistics, data mining and machine learning. Conversely, there has been few non-hybrid bio-inspired algorithms (hybrid algorithms referring to algorithms combining bio-inspired models with more traditional machine learning models) proposed to tackle the issue. Some approaches involve using a bio-inspired optimization technique to select and extract features, but the actual classification is usually performed by applying traditional machine learning algorithms, such as State Vector Machine, k-Nearest Neighbors and Random Forests [Zhou and Zafarani, 2019]. The proposed model takes a novel approach to this classification problem, adapting an Artificial Immune System (AIS) inspired model for the classification task using a wide variety of linguistic features.

1.2 Goals and Research Questions

The aim of this section is to present the goal for the research and model implementation, as well as the research questions which guided the literature review. The goal represents the overall objective of the thesis. The research questions seek to split the main goal into granular sub-goals – which are easier to evaluate in terms of fulfilment than the overall goal.

Goal *To investigate the applicability of an Artificial Immune System for the classification of fake news articles.*

The research questions aim to address two different sides of the overall goal, the AIS side and the fake news feature extraction side. This division is also reflected in the literature study in Chapter 3. As using an AIS to perform classification of fake news is a completely novel approach, the problem of finding suitable evaluation criteria is prominent. To combat this issue, the goal is divided into three research questions.

Research question 1 *How should the traditional design of an Artificial Immune system be adapted to enable fake news classification?*

The first research question is concerned with the AIS part of the proposed model. As the underlying base of the model is an AIS, the characteristics of this model should be tuned to reflect the challenges of classifying news articles. AIS models have typically been used to perform classification on benchmark classification datasets, therein not using extracted feature values (which may not be representative). The added challenge of presenting an AIS with high-dimensional and potentially unreliable features must therefore be considered and investigated thoroughly.

Research question 2 *Which feature extraction strategies are suitable for an AIS adopted for the classification of fake and real news articles?*

The second research question is concerned with which features that should be extracted from the news article samples. These extracted feature values is included in the feature vectors of the antibodies/antigens of the AIS and as such, a sophisticated feature extraction strategy is essential to the success of the proposed model. The goal of this research question is to gain knowledge of what such a feature extraction strategy should include. Furthermore, extensive pre-processing and feature extraction strategies may boost the accuracy of the model, at the expense of sacrificing efficiency and general applicability. As the overall goal is a model that rapidly could be applied to previously unseen news articles, it is preferred that the pre-processing and feature extraction of the news article are computationally efficient.

Research question 3 *How does the proposed model perform, in terms of accuracy, compared to other fake news classification methods?*

The third research question is related to the comparison to other fake news classification models. These classification models may employ non-bio-inspired techniques, but use the same datasets for training and testing as the proposed model. Although not similar (in terms of approach) to the proposed model, these models may serve as benchmarks for evaluation as the overall goal is the same. Such comparisons also establish the relative position of the proposed model in the research field.

1.3 Research Method

The aforementioned goal was chosen after a relatively brief literature study and the subsequent discovery that this may be a ”missing link” in the research area. Further, a more specific literature search was performed to confirm that this was actually the case. After establishing the need for research into such a model, the focus was shifted onto similar models for inspiration. This structured literature review (further specified in 1.4) formed the technical base on which the proposed model was designed. The relevant findings of the literature review is presented in Section 3.

After the structured literature review was performed, a model was proposed, based on the results of the literature review. Further, the model was implemented and experiments conducted in an iterative fashion, where the model was tweaked to reflect the experimental results. These tweaks apply both to fine-grained parameter tuning and more grand-scale changes to the nature of the model. Lastly, results of the proposed model were compared to the results of similar models – in line with the stated research questions.

1.4 Structured Literature Review

This section outlines the strategy used for the structured literature review, including sources, search words, inclusion criteria and evaluation criteria. The literature search was guided by the following questions. As the scope of the thesis narrowed, the search scope narrowed as well, focusing on articles highly relevant to the proposed model.

- What are potential application areas of AIS models?
- How can feature extraction/selection be applied to AIS models?
- How can AIS be applied to text classification?
- What attributes characterize fake news articles and not real news articles?
- How can classification of fake news articles be conducted?
- Which feature extraction techniques are applied to raw news articles to perform fake news classification, in existing models?

The sources chosen for the literature review were selected to cover the major publishing platforms within the scientific area of Computer Science in general and specifically bio-inspired AI. Articles of interest to the proposed model would, with a high probability, be published/available at least one of the platforms. Additionally, Google Scholar was used as the search engine for the literature search. As this search engine searches across various publishing platforms and sources for scholarly material and articles, this research strategy was likely to cover most relevant literature.

Publishing platforms and search engines used for literature search

- Google Scholar
- IEEE Explore
- ACM

- SpringerLink
- ResearchGate
- ScienceDirect

Additionally, Iris.ai was used to identify relevant literature. This is a machine learning system designed for reviewing massive collections of research papers and identifying relevant material through using Natural Language Processing (NLP). The system works by processing a scientific article chosen by the user and then identifying similar literature. The articles chosen for the Iris literature searches were an AIS-based spam detection model by Saleh et al. [Saleh et al., 2019] and Klyuev's proposed semantic approaches for classifying fake news articles [Klyuev, 2018].

Further, when deciding the specific search phrases that would be used for searching the stated sources, there was an emphasis on covering synonyms and words having relatively similar meaning in the context of the research area. Additionally, different word combinations were employed to cover more relevant material.

Keywords used for literature search (synonyms comma separated)

- AIS, Artificial Immune System, Immune System
- Fake News Classification, Fake News Detection, Fake News Filtering
- Fake News Benchmark, Fake News Dataset, Fake News Attributes, Fake News Structure
- Spam Filtering, Spam Detection, Spam Classification, Anti-Spam Filter, Spammer Detection
- Semantic, Linguistic, Style-based, NLP, Natural Language Processing, Text Mining, Text Analytics, Sentiment Mining

The inclusion criteria for relevant articles were quite broad. This is due to the novel nature of the proposed model, as to not exclude any literature that might be relevant. More recent articles were preferred when the research area was related to fast-moving disciplines such as fake news characteristics, while older and renowned papers were satisfactory for less dynamic topics such as AIS. The literature search pointed towards an increased research focus on AIS models adapted for spam detection around 2003-2012. To include these in the literature study, a generous requirement of the research being published after the year of 2000 was decided. Regarding the literature concerned with fake news detection, most of the literature is less than 5 years old. Therefore, the articles selected for this part of the model were given a stricter requirement of being published no earlier than 2015.

Inclusion Criteria

- For AIS-based spam-detection: the research presented was conducted no earlier than the year of 2000
- For fake news classification: the research presented was conducted no earlier than the year of 2015
- The literature's main research topic is either AIS adapted for text classification or related to the identification of fake news
- The literature seems relevant from only reading the abstract and conclusion
- The literature is peer reviewed and published on a recognized publishing platform

Lastly, quality criteria were established – in order to easily evaluate the quality and relevance of the collected literature. These criteria serve as a subset of the inclusion criteria, to compare collected articles and limit the scope further.

Quality Criteria

- The research conducted is quite recent, preferably less than 5 years old
- The literature's main research topic is either AIS adapted for spam classification or fake news detection by using a linguistic/semantic approach
- The article has a significant amount of citations (although this will, of course, depend on the recency of the article)
- The literature clearly presents the model proposed, preferably with pseudocode and/or illustrations
- The literature clearly states the results and compares them to similar models, preferably including standard deviations
- The literature contains a section dedicated to possible future work within the research topic

1.5 Preliminary Process Overview

Throughout the process of literature review, the research goal was developed iteratively. As the research progressed, model ideas were proposed, discarded and changed. Figure 1.1 shows the preliminary overview of this process. Model ideas and problem scopes are illustrated as rounded rectangles, while discoveries leading to the discardment of the related model idea are illustrated as grey rectangles.

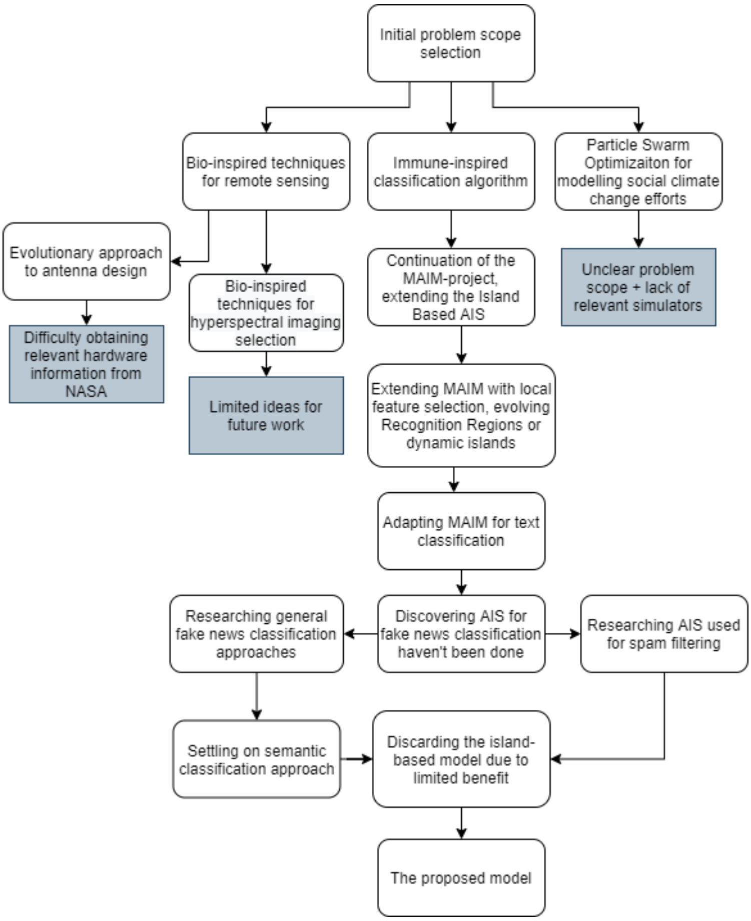

Initial problem scope selection
Bio-inspired techniques for remote sensing
Immune-inspired classification algorithm
Particle Swarm Optimizaiton for modelling social climate change efforts
Evolutionary approach to antenna design
Bio-inspired techniques for hyperspectral imaging selection
Continuation of the MAIM-project, extending the Island Based AIS
Unclear problem scope + lack of relevant simulators
Difficulty obtaining relevant hardware information from NASA
Limited ideas for future work
Extending MAIM with local feature selection, evolving Recognition Regions or dynamic islands
Adapting MAIM for text classification
Researching general fake news classification approaches
Discovering AIS for fake news classification haven't been done
Researching AIS used for spam filtering
Settling on semantic classification approach
Discarding the island-based model due to limited benefit
The proposed model

Figure 1.1: *Flowchart illustrating the preliminary process overview*

Initially, three separate problem scopes were considered. Two of these were based on previous bio-inspired Master's theses at NTNU. The first one was concerned with bio-inspired techniques for analyzing image data from satellites, known as remote sensing. This problem scope was further divided into applying evolutionary approaches to antenna design and the continuation of a previous Master's thesis focusing on hyperspectral imaging selection. Regarding the hyperspectral image selection, potential future work was limited and the problem scope was therefore dropped. The antenna design domain also proved problematic, due to difficulties obtaining hardware information from relevant parties, where NASA was the prevalent one due to previous collaboration. After some consideration, this problem scope was also dropped.

Secondly, a potential problem scope concerned with using Particle Swarm Optimization (PSO) for modelling social dynamics revolving around climate change efforts was considered. Although highly relevant and interesting, this problem domain was also dropped, due to an unclear problem scope and limited groundwork on which to build a model upon.

Lastly, the continuation of the work of the MAIM model [Baug et al., 2019], was considered. Also the product of a previous NTNU Master's thesis, MAIM is an Island Model-based AIS, using principles of population migration between separate isolated islands to boost the solution diversity and run-time efficiency of an AIS. Several improvements of the model was considered, but the choice was made to instead focus on applying an AIS to text classification. Upon further research, it became apparent that although several researchers had applied AIS models to spam classification, there was no published work describing the application of an AIS at fake news classification. After this discovery, the problem scope shifted to instead research and implement an AIS model for fake news classification.

Upon the shift in problem scope, a reconsideration was made as to whether to keep the Island Model architecture of MAIM, or to start fresh with a more traditional AIS as base. As the benefits of MAIM were mostly prevalent for datasets with fewer features, the impression was that the potential benefits of an Island Model architecture was not worth the additional work required for implementation.

Additionally, using an Island-based model also means having to utilize crossover operation between population members, instead of the more prevalent clonal selection [De Castro and Von Zuben, 2000] algorithm used in most of the AIS models adapted for spam classification. As these models serve as the main architectural inspiration of the proposed model, the choice was made for the proposed model to employ a variation of the clonal selection algorithm, with cloning and subsequent mutation rather than the aforementioned crossover operation. More about this in Chapter 3 and Chapter 4.

1.6 Thesis Structure

The subsequent sections will be presented as following: Chapter 2 is aimed at providing the reader with the required background knowledge needed to understand the proposed model. Additionally, the motivation behind the proposed model is presented – in light of the current state of the research field. It should be noted that some of the content is adapted from the research project conducted during the 2020 Fall semester.

Chapter 3 introduces the current state of the art. This applies to both the research area of fake news detection in general, as well as AIS adapted for fake news classification. The focus of the chapter is on relating the collected literature to the proposed model and present arguments for why the proposed model is a valuable contribution to the academic research. Similarly to Chapter 2, some of the content is adapted from the aforementioned research project.

Chapter 4 introduces the proposed model – as a natural continuation of Chapter 2 and 3. The architecture and behaviour of the model is presented in detail, with the aim of presenting the model in a reproducible fashion. Additionally, the reasoning behind and justification of the chosen architecture is explained.

Chapter 5 presents the results obtained by the proposed model. The section also includes the experimental setup, as well as a discussion of which parts of the proposed model may have contributed the obtained results.

In Chapter 6, a conclusion regarding the obtained results and findings is presented. Additionally, an evaluation of the goals and research questions is provided. Further, contributions to the research area are discussed, as well as possibilities for future work. Finally, bibliography and appendices are presented.

Chapter 2

Background Theory

This chapter aims to introduce the necessary background knowledge needed to understand the proposed model.

2.1 Background Theory

2.1.1 Bio-Inspired Computing

Within the field of Computer Science, biologically inspired (bio-inspired) computing is a field of study which aims to employ biological processes to solve a variety of computing problems. Examples of such biological processes are swarm behaviour, evolution and clonal selection. Bio-inspired models seek to adapt these processes to concrete computational problems. Applicable problem domains include optimization, classification, clustering and more. Additionally, bio-inspired algorithms have shown to be efficient at both discrete and continuous problem domains [Kar, 2016]. Needless to say, this makes such algorithms an exciting area of research in an increasingly data-driven world.

Most bio-inspired algorithms make use of the principle of natural selection for their computation, more specifically the "the survival of the fittest". This principle defines the biological concept of fitness as the degree of reproductive triumph, meaning that the individuals who are genetically most adapted to their environments tend to produce more offspring. Thus, subsequent generations will inherit the most successful attributes and the population as a whole moves towards improved adaptivity. This is the principle behind Darwinian evolution [Darwin, 1859]. For this strategy to work, a degree of variety between individuals is needed. This is achieved through re-combination of attributes (breeding) and mutation.

Mutation refers to change in the DNA sequence as it's being copied, cumulatively leading to a change in attributes [Collins, 2020]. This results in the population "trying out" different sets of attributes and how they perform in terms of leading to increased reproductivity.

Traditional machine learning is revolved around learning from experience by identifying emerging patterns in the data. These approached often use gradient descent to iteratively move toward solutions that minimize the loss function. In high-dimensional spaces, this calculation can be quite demanding and time-consuming. Comparatively, bio-inspired methods often employ a more stochastic approach, where successful solutions breeds more successful solutions and random mutation ensures that diversity is contained.

2.1.2 Evolutionary Algorithms

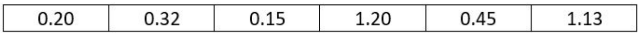
The proposed model is not technically an Evolutionary Algorithm (EA), but an introduction to EAs is nonetheless included as it illustrates key functions and components that the proposed model employs.

EAs aim to generate solutions to optimization problems based on the principles of evolution and natural selection. They do this by using the following components:

- Representations (of individuals)
- Population initialization
- Fitness function
- Parent selection
- Recombination (crossover)
- Mutation
- Survivor selection

First, a representation of the individuals is chosen. Usually, each individual represents an individual solution to a specific problem and are usually represented in memory as a bit string, an array of integers/decimal numbers or a tree.

The representation also includes the formulation of some fitness evaluation, to differentiate the solutions as to how well they solve the given problem. This fitness evaluation could for instance be the cost of traversing a graph a given way, as in the Travelling Salesman Problem [Bernhard and Vygen, 2008]. Another



0.20	0.32	0.15	1.20	0.45	1.13

Figure 2.1: *Representation of an EA problem solution, as a vector of floating numbers*

example is the structural integrity of a construction, if the problem at hand is to generate suggestions for the assembly of some construction. The fitness function can also be a more complex calculation, which includes several aspects of the problem space we want to optimize.

After the representation has been decided, the next component is the population initialization. The population consists of a (usually pre-stated) number of individuals. In accordance with Darwin's theory of natural selection, it is the population as a whole – and not the individuals – that evolve towards a greater adaptation [Darwin, 1859]. Several initialization strategies can be employed, for instance random start values or implementation of some external heuristics to enhance the fitness of the initial population. After the population has been initialized, five steps are iteratively repeated until the termination of the algorithm [Homayounfar, 2003]:

1. Fitness evaluation of every individual in the population. This fitness function will be closely tied to the computational problem which the algorithm is trying to solve.

2. Parent selection. The most fit individuals (according to their respective fitness values) are selected for reproduction. The heuristic for choosing these individuals is up for experimentation, but often *tournament selection* is used. This approach randomly selects k individuals from the population of size N. Thereafter, these individuals compete in a tournament of size k, where the one with best fitness is crowned the winner. This individual is then chosen for reproduction. This process is then repeated until the desired number of parent individuals is found, n. The advantages of this approach over simply selecting the best n individuals, is that there is less likeliness of quickly converging towards a local optimum in the search space – due to more diversity within the population [Fang and Li, 2010].

3. Recombination (or crossover) is performed between the n individuals selected for reproduction. Strategies for crossover vary greatly depending on solution representation and the nature of the problem. An usual strategy,

however, is *k-point crossover*. Illustrated in Figure 2.2 with k=2, this approach splits both parents at k points into *k+1* segments. The children are then created by combining these segments, concatenating every other segment from each parent.

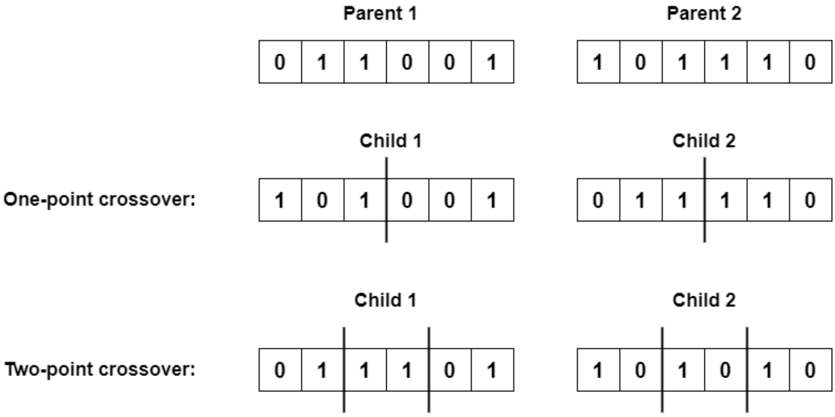


Figure 2.2: *1-point and 2-point crossover*

4. Mutation is then performed by stochastically changing values in the individual representations. Naturally, this heuristic is also dependent on the chosen representation of solutions. Building onto the example representation in Figure 2.1, a reasonable mutation heuristic would be to iterate through the vector elements with a mutation probability of p. Mutation would then be applied to a random number of the elements in each solution. This mutation could, for instance, be to increment or decrement the value with some random number from a probability distribution. This way, both the mutation rate and the mutation amount include a stochastic element, preventing the population from converging prematurely.

5. Lastly, the optional step of survivor selection. Some of the individuals may be directly transferred to the next generation, without being the products of crossover nor mutation. The reasoning behind this is to keep exceptionally good solutions across generations. However, this strategy should be used with care as this has the potential to severely limit diversity in the population.

It is important to note the importance of balancing the parameters which direct the algorithm. For instance, an inflated mutation rate (and amount) may lead to too much diversity and the algorithm might resemble a random search and diverge. On the other hand, a deflated mutation rate combined with too high selection pressure may lead to premature convergence at a local optimum. Tuning these parameters is central to the implementation of these types of algorithms [Del Ser et al., 2019].

2.1.3 Biological Immune Systems

The natural biological immune system is revolved around a set of immune cells called *lymphocytes*, which originate in the bone marrow. These cells are further divided into B- and T-cells. The B-cells fight viruses and bacteria, often referred to as *antigens*. They do this by producing Y-shaped proteins called antibodies. These antibodies are specific to a subset of antigens capable of hurting the host, known as pathogens [CTCA, 2017]. Their specificity is achieved through a specially designed receptor which binds to specific pathogens through chemical interaction. The antibody is said to *activate* when such a binding occurs. The strength of this interaction is known as the *affinity*, depending on the relative match between the antibody receptor and the antigen shape. When this affinity is high, the antigen is said to be within the *recognition region* (RR) of the antibody. An important aspect of this imperfect matching is that an antibody can be triggered to activate by a number of antigenic patterns – leading to enhanced noise tolerance [Secker et al., 2003].

T-cells are further divided into helper and killer T-cells. The helper T-cells trigger the B-cells into making antibodies and assist in the creation of killer T-cells. The killer T-cells are responsible for destroying cells that are infected by pathogens [CTCA, 2017]. Overall, the T-cells are responsible for alerting the rest of the immune system to threats and encouraging the B-cells to ramp up their response.

It is the manipulation of the populations of B- and T-cells which gives the biological immune systems their inherently dynamic and adaptive abilities. When a antibody activates (a binding takes place), an immune response is triggered and the cell starts a process of mutation and cloning. In a process known as *clonal selection*, the cloning rate is proportional to the affinity of the connection, while the mutation rate is inversely proportional to the affinity. This process creates significant selection pressure on the antibodies, leading to increased affinity for future connections and enhanced immune response efficiency [Secker et al., 2003].

Biological Negative Selection

Negative selection refers to the differentiation between self and non-self, considered one of the most central mechanisms in the biological immune system, protecting the body from self-reactive lymphocytes [Ji and Dasgupta, 2007]. During the T-cell maturing process, T-cells are first generated by a random genetic rearrangement process. Further, the T-cell undergo a selection procedure in the thymus where the T-cells recognizing/binding to self-cells are destroyed. This procedure is known as negative selection, protecting the body against T-cells which encourage attacks on the self-cells. Thereafter, the T-cells are deployed into the biological immune system, ready to attack external pathogens [Idris and Selamat, 2011].

2.1.4 Classification

Classification refers to the task of categorisation. Given a set of vectors, containing a set of attributes, the task is to assign a *class* to each vector. These vector attributes are often referred to as *features* and usually consist of numerical values [Alpaydin, 2010]. These features describe central traits of the element which is being classified and can be represented in a n-dimensional space, where n is the number of features. Supervised learning refers to machine learning algorithms where the model is given a *training set* containing a number of such vectors, with their correct class assigned. The algorithms then use these examples to construct models, which aim to find a curve in the feature space that divides the examples into their respective classes. An example of this can be seen in Figure 2.3. Future (unseen) vectors lacking an assigned class can then be classified simply by

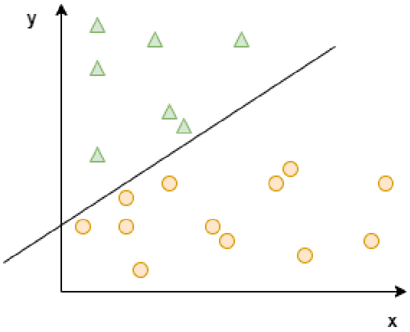


Figure 2.3: *Simple classification example for a 2-dimensional feature space. The line separates the elements into two distinct classes*

plotting them in the feature space and see which category it fits into [Alpaydin, 2010]. This strategy is the fundamental logic behind classification, which the proposed model is based on.

2.1.5 Pre-processing and Feature Extraction

In terms of classification problems considering high-dimensional data, pre-processing and feature extraction are central concepts. These concepts refer to the "cleaning" of the input data and the creation of new feature values based on the given input data, respectively. This data cleaning includes the removal of unreliable and noisy data which obscures the classification accuracy, as well as normalizing feature values.

Feature extraction refers to creating new features from the data input. The reasoning behind this strategy is that the data input can be high-dimensional – which means that training the model with the raw data can be very slow. By creating new features by combining original ones, the feature space dimensionality can be significantly reduced [Levine, 1969]. Additionally, some types of input data can be very unfit for classification directly. Raw text is an example of this, which is highly relevant for the proposed model. To perform classification in the proposed model, feature values are created from the textual data input, through pre-processing and feature extraction operations. Finally, the output of the feature extraction will be the data which the model use for training and testing.

2.1.6 Cross-Validation

k-fold cross-validation is a commonly used validation approach, which generally results in less biased estimates of the performance (accuracy) of a model, than a simple training/testing-split. The scheme starts by randomly shuffling the dataset and then splitting it into k parts of roughly equal length. Further, one of the k parts are removed from the remaining parts. The model is then trained using the remaining *k-1* dataset parts as training data. Then, the model is tested on the dataset part initially removed. The current model iteration has never been exposed to this data before and the results will therefore be an accurate estimation of the model's performance on unseen data. Further, the accuracy obtained on the current testing set is saved and the model is discarded. This process is repeated k times, each round with a different part of the dataset used for testing. Finally, the accuracy scores obtained are averaged – to produce a final accuracy score [James et al., 2013].

An important problem to consider in terms of cross-validation testing, is that of *knowledge leakage*. This refers to inadvertently providing the model with information from the testing set, in the training phase. This exposes the model to data which it isn't supposed to see until the testing phase and thus the model may adapt to this data in a way that it otherwise wouldn't, resulting in overly optimistic results when the model is tested.

2.1.7 Artificial Immune Systems

Artificial Immune Systems (AIS) aim to replicate some of the functionality and inner workings of biological immune systems. It is important to distinguish between bio-inspiration and bio-plausible AIS models. The immune system is highly complex and there is a trade-off between realism and computational benefits when it comes to modelling it. Exactly where this trade-off point is located is still an open question. The relevant AIS models for this thesis take inspiration from biological immune systems, with a relatively high-level view at the natural complexity. This means that the modelling complexity is drastically reduced (compared with biological immune systems), while the natural processes which provide considerable computational benefits are taken advantage of.

Generally, AISs have had success in a number of fields, including malware/anomaly detection, combinatorial optimisation, clustering, classification and more [Hart and Timmis, 2008]. In this introduction to AISs, the focus will be on classification-based AIS, which is both the most common type as well as the relevant type for the proposed model.

Antigens

In the AIS, antigens are the single data entries in the dataset, fed as input to the classifier. The model will then perform classification of these data entries. The antigens can be represented in a variety of ways, but the common representation (and the one used in the proposed model) is a vector of length n, consisting of n features. These feature values can be thought of as the coordinates of the antigen in an n-dimensional feature space. It is the task of the antibodies to classify these antigens, based on their respective position in this feature space [Read et al., 2012].

Antibodies and Recognition Regions

In classification based AISs, the antibodies have a similar representation as the antigens. The only difference is that the antibodies have two additional elements – the class and the *Recognition Region* (RR) radius (if a hypersphere shape is used for the RR). The class is the label of the antibody, collected from the data entry and used to further predict the class of the antigens. The antibodies share the feature space with the antigens and will "search" their local n-dimensional space for antigens. This local space is known as the antibody's *recognition region.* This region can be modelled as a variety of different geometrical shapes, although hyperspheres are the go-to option for most applications [Hart, 2005]. A hypersphere is the generalization of a three dimensional sphere, which is a set of all the points within a specific distance (referred to as the radius) from the center. In AISs, this radius is dynamic and may increase or decrease for each individual antibody throughout the training phase of the model.

The antibody will bind to all antigens within its recognition region, with a connection strength known as the *affinity.* The affinity is usually calculated as the inverse Euclidean distance from the antibody to the antigen in question [Read et al., 2012]. This means that the closer the antigen and antibody is in the feature space, the stronger the affinity will be.

The antibody will then try to assign its class to the antigens within its recognition region. In the case that antigens are within several recognition regions, a heuristic is needed to predicting the class which the antigen belongs to. Such heuristics can be a k-Nearest Neighbors vote (where the connected antibodies vote their individual class) or by summing all the affinities for each class and selecting the class with the most cumulative affinities [Dudek, 2012]. These antigen class predictions are the output of a classification AIS. In Figure 2.4, an illustration of the antibodies and class prediction is presented.

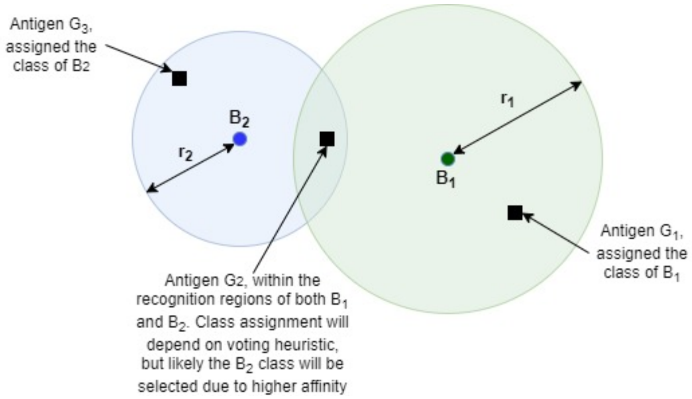

Antigen G3, assigned the class of B2
B2
r2
r1
B1
Antigen G1, assigned the class of B1
Antigen G2, within the recognition regions of both B1 and B2. Class assignment will depend on voting heuristic, but likely the B2 class will be selected due to higher affinity

Figure 2.4: *Antibody (B_x) and antigen (G_x) interactions in a 2-dimensional AIS feature space (r_x is the RR radius of antibody x)*

Clonal Selection

Most AIS models use the *Clonal Selection Principle* for training the model [Read et al., 2012]. This principle describe the basic functionality of the immune response to antigenic stimuli [De Castro and Von Zuben, 2000]. It is based on the idea that only the cells that recognize antigens are allowed to proliferate, thus creating selection pressure within the antibody population. This selection pressure will then lead the population as a whole towards increased antigenic adaption.

The CLONALG algorithm is a well known algorithm for implementing the principle of clonal selection in AIS models, employing evolutionary concepts in its implementation. The basic functionality of the algorithm is to calculate the affinity between all the antigens and antibodies and then perform cloning of the antibodies with the highest cumulative affinity to antigens. This cloning is performed proportionally to the affinity, i.e. the antibodies with the highest affinity values will have a higher chance of being the subject of cloning. Then, the newly cloned antibodies will be mutated, at a rate inversely proportional to their respective affinity. Lastly, the highest affinity antibody clones is copied into a memory set, and the lowest affinity antibodies will be replaced by randomly generated alternatives [De Castro and Von Zuben, 2000]. The memory set will then be used to classify previously unseen antigens.

Computational Negative Selection

The biological negative selection mechanism in the natural immune system was introduced in Section 2.1.3. Research has been conducted into mimic this mechanism in artificial immune systems. This research has primarily focused on binary classification, where the model is only distinguishing between two classes [Ji and Dasgupta, 2007]. Although some work has been conducted into generalising the mechanism to multi-class classification, such as MINSA [Markowska-Kaczmar and Kordas, 2008], the main focus has been on binary classification. As the proposed model mainly employs binary classification, this section will mainly consider binary negative selection models.

These models generally employ a population of antibodies with the same class, spread throughout the feature space. If an antigen falls within the recognition region of an antibody, it is classified according to that antibody's class. If not, it is classified as the alternative class. This removes the need for voting heuristics when an antigen is covered by several antibody recognition regions, as all the antibodies share the same class. The approach could also limit the amount of training data needed, as all the training data potentially could be single-class [Ji and Dasgupta, 2007]. It will then become the goal of the AIS to detect antigens which differ from the established "norm". The usual terms for this norm is "self", while antigens that the system wants to remove/recognize (connect its antibodies to) are referred to as "non-self". For practical applications, this norm can be swapped to match the dataset available and the classification results wanted. I.e. for spam classification the antibodies could either be spam samples or non-spam samples.

It is important to note that if single-class training data is used, the antibody population should be relatively large and features should be selected carefully in order to prevent clustering of the antibodies throughout the feature space. This is because these single-class antibodies must be an accurate representation of all the potential data samples (antigens), in order to not classify incorrectly. Further, the volume of the recognition region will approach zero as the number of dimensions increases – in comparison to the exponentially growing feature space. This problem is known as the "curse of dimensionality" [Baug et al., 2019]. Single-class negative selection algorithms are especially vulnerable to this problem, due to the aforementioned problem of covering the feature space to a satisfactory degree [Ji and Dasgupta, 2007]. Thus, the number of features for single-class negative selection models should be kept relatively low.

2.1.8 Text Analytics and Pre-Processing

The field of text analytics is revolved around extracting meaningful insights and sentiments from raw text data. With the growing amount of available text data on the Internet, this has become increasingly valuable. A wide range of analytic models exists for this purpose, ranging from highly complex linguistic natural language processing (NLP) tools to statistical approaches requiring less pre-processing.

Tokenization

Within the field of text analytics, tokenization refers to the process of splitting a text into smaller parts – called tokens. These tokens are usually single words, although they could also be groups of words or even single characters. As for single word tokenization, there's also a question of correct parsing. Consider the sentence "Mr. O'Malley isn't entertained by the boys' stories". Here, there are three apostrophes, all used in different ways. There are several way of parsing this sentence, one could consider the apostrophes as whitespaces and divide text where they appear or one could ignore them and only split at whitespaces and punctuation dots. These parsing choices could impact model performance further down the road and should be considered accordingly [Oda and White, 2005].

Stop Word Removal

Stop words are words which doesn't add any significant meaning to a sentence. As these words don't carry any symbolic weight, they can be removed without the sentence losing its meaning. Examples of such words are "a", "the", "an", "about", "by" etc. By removing these words in the pre-processing phase, the volume of data is significantly reduced. Simultaneously, noise is removed from the data, as the stop words generally provide little information that can be used for classification or clustering [Mahmoud and Mahfouz, 2012]. In a worst-case scenario, the inclusion of these words might even confuse the model into performing poor classifications with a high degree of certainty.

However, improper removal of stop words might result in changing the meaning of a text document. This means that the stop word selection should be considered carefully. For instance, if "the", "not" an "was" were to be considered stop words and removed from the sentence "The book was not good", the meaning of the sentence would be changed drastically, to "book good". The sentiment of the sentence would then change completely, from negative to positive – which might lead to poor classification results.

Regular Expressions

Regular expressions (regex) are patterns used to match certain character combinations. These can be very general or highly specific. For instance, the general regular expression "/ab*c*/" matches an "a" followed by zero or more "b"s and zero or more "c"s. In other words, this expression also matches single "a"s – occurring in a wide range of words. Comparatively, the pattern "/met/" only matches words where that exact character combination occurs, such as "metronome", "meter" or "metropolitan". This way, regular expressions can be used to detect certain words or word-combinations throughout a document [Oda and White, 2005].

Bag-of-Words

Bag-of-words (BoW) is an extensively used technique within the field of text analytics [Secker et al., 2003]. The technique is a way of representing a text in a simplified way, disregarding word order and grammar – but keeping multiplicity of each word. This works by iterating through the processed document, split at whitespaces (and possibly removed punctuation marks), and counting the occurrences of each word. For instance, consider the text: "Mark liked the movie. Samantha also enjoyed the movie.". A BoW representation of this sentence would be:

{"Mark":1, "liked":1, "the":2, "movie":2, "Samantha":1, "also":1, "enjoyed":1};

The BoW technique could also be combined with stop word removal, as a way to disregard high occurrences of words like "a" and "the".

Term Frequency

A common way of using BoW for classification, is to calculate term frequencies. The simplest method of calculating term frequency is simply to divide the occurrences of a word by the total number of words in the document. This frequency provides an estimate of the importance of the word, in the document.

A more sophisticated term frequency calculation is the Term Frequency - Inverse Document Frequency (TF-IDF) method. The reasoning behind this method is to deal with the issue of high frequencies of words carrying little useful information. Words like "the" and "a" is likely to have high term frequencies, yet they are of little use for classification purposes. The TF-IDF method works by diminishing the weight of words occurring very frequently across all the document in the set, while increasing the weight of terms which (generally) occur rarely. TF-IDF is calculated as:

$$TF - IDF = TF(t,d) \cdot IDF(t,D) = \frac{f_{t',d}}{\sum_{t \in d} f_{t,d}} \cdot \log\left(\frac{N}{1+n_{t'}}\right)$$

where $f_{t',d}$ is the raw count of term t' in the document d, N is the total number of documents in the document set and $n_{t'}$ is the number of documents where term t' occurs. It should be noted that several alternative methods of calculation are possible as well.

N-grams

N-grams are a way of tweaking BoW to include more information about the text [Saleh et al., 2019]. Consider a simple sentence such as "The child asked for ice cream.". If one simply removes punctuation and splits the sentence at whitespaces, "ice" and "cream" would be two separate tokens. These words carry a significantly different meaning when appearing separate, than when they appear together. This meaning might be lost if single-word BoW is used in the tokenizer. One way to solve this is using N-grams. These are tokens consisting of N words each, capturing the spatial information of the words. If N-grams with N=2 (called a *bigram* model) is applied to the sentence above, the N-grams would be constructed as: ["The child", "child asked", "asked for", "for ice", "ice cream"].

White- and Blacklisting

Within the field of textual classification, white- and blacklisting refers to the practice of explicitly including (whitelisting) or excluding (blacklisting) some textual document from a document set based on the occurrence of certain words [Secker et al., 2003]. These lists may be constructed based on pre-existing knowledge, or during runtime. For instance, word like "Viagra", "free", "buy" and "cash" are much more probable to be included in spam e-mails than in non-spam e-mails and as such, they could be added to a blacklist of words in an anti-spam filter [Saleh et al., 2019]. Correspondingly, words occurring regularly in non-spam e-mails could be added to a spam filter whitelist.

Word Embeddings

As mentioned, the AIS models generally operates on real-valued numbers as feature values. Therefore, ways of extracting real-valued numbers from the article text samples are central to the proposed model. Strategies like term frequency, lexicon lookups (black-/whitelisting) and TF-IDF were mentioned as ways of doing this. Another way is by using word embeddings. Word embedding refer to placing similar (in terms of symbolic meaning) words close in the representational feature space. This feature space can have different number of dimensions, but (importantly) this number is equal for all words, i.e. the resulting output vectors are fixed-length. For instance, the words "happy" and "excited" should be relatively near in the representational feature space (the Euclidean distance between them is relatively small), as the words have a similar symbolic meaning. This is a complicated issue, subject to considerable recent research. Google's BERT (Bidirectional Encoder Representations from Transformers) [Devlin et al., 2019] and AllenNLP's ELMo (Embeddings from Language Models) [Peters et al., 2018] are examples of recent developments concerned with this problem. These will be explained further in Section 3.2.3.

2.1.9 Fake News Classification

Fake news classification models can employ a variety of strategies, but the usual division is the partitioning into four distinct strategies [Zhou and Zafarani, 2018]:

1. Fact-based strategies, concerned with fact-checking the information stated in the news articles. As the amount of manual labour for this strategy can be immense, models adapting this strategy usually try to implement some sort of automatic fact-checking against a knowledge base.

2. Semantic-based (linguistic-based) strategies, concerned with differentiating real and fake news articles based on their writing style.

3. Propagation-based strategies, concerned with studying the different propagation patterns of fake and real news articles through social media. This includes who, when and how many people share, comment or react to the news articles.

4. Source-based approaches, concerned with classifying news articles based on the credibility of the source(s). The sources can be the publishing media, the reported speaker or the references provided.

As the proposed model seeks to employ the semantic approach to the classification task, the other strategies are not explained in detail. The intuition behind the semantic-based strategies is that fake news articles are written in a different style than real news articles. Often, the authors want to promote an emotional reaction in their readers – potentially leading to distrust or enragement towards some entity. A 2018 study [Zhou and Zafarani, 2018] showed that fake news articles, compared to real news articles, have a higher degree of informality, diversity, subjectivity and are written with a higher grade of emotion.

One approach to measure the degree of informality is to count the occurrences of swear words. For diversity, a measure that can be used is the percentage of unique verbs. For subjectivity, the amount of reporting verbs could be used. These are verbs which change direct speech into reported speech, i.e. "I have seen the new film" into "I told her (that) I had seen the new film". In this example, "tell/told" is the reporting verb. Lastly, emotional writing can be measured by the use of emotional and strong words such as "lie", "steal" and "kill" – and their various forms. Such analytic characteristics are the groundwork which semantic-based classification models seek to make use of.

Chapter 3

State of the Art

This chapter presents the current state of the art in the research area related to the proposed model. The chapter is split into two distinct parts. Section 3.1 will introduce the current state of the art in AIS used for classification, while Section 3.2 will look at current development within the research field of fake news classification. Section 3.2 will focus primarily on semantic based approaches, as the other approaches briefly introduced in 2.1.9 are of less relevance to the proposed model.

3.1 Artificial Immune Systems

The goal of this section is to introduce the current state of the art within the field of Artificial Immune Systems (AIS) and the various approaches to central AIS characteristics such as affinity calculation, mutation, reproduction strategies and class prediction. These central attributes have a significant impact on the model performance and, as such, the selected approaches vary greatly in recent contributions.

Additionally, adapting an AIS for fake news classification is a novel approach and, as such, the closest problem scope in the literature would be AISs used for spam detection. Thus, section 3.1.9 will focus on contributions focusing on AISs adapted for spam classification.

3.1.1 Affinity Calculation

As affinity values are critical to the functioning of AIS models, several different methods of calculation have been proposed. The most common affinity measures make use of Euclidean distance to calculate the distance between an antibody and an antigen in an n-dimensional feature space. The Euclidean distance is defined as:

$$d(p,q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + ... + (q_n - p_n)^2}$$

for n dimensions. q_x and p_x are the feature values at index x, for the antibody and antigen.

In AISLFS [Dudek, 2012], the authors employ local feature selection in their model, which means that each antibody only calculates distance and affinity to antigens for some selected features. The distance is calculated as:

$$d(\boldsymbol{y}_k, \boldsymbol{x}_j, \Omega_k) = \left(\sum_{i \in \Omega_k} |\boldsymbol{y}_{k,i} - \boldsymbol{x}_{j,i}|^p \right)^{1/p}$$

with $p = 1$ if the Manhattan distance metric is used and $p = 2$ if the Euclidean distance metric is used (the model was run with both alternatives, scoring mostly evenly). Further, the affinity is calculated as:

$$a(\boldsymbol{y}_k, \boldsymbol{x}_j, \Omega_k) = \begin{cases} 0, & \text{if } d(\boldsymbol{y}_k, \boldsymbol{x}_j, \Omega_k) > r_k(\Omega_k) \textit{ or } r_k(\Omega_k) = 0 \\ 1 - \frac{d(\boldsymbol{y}_k, \boldsymbol{x}_j, \Omega_k)}{r_k(\Omega_k)}, & \text{otherwise} \end{cases}$$

Where Ω_k is the set of selected features for the kth antibody and $d(\boldsymbol{y}_k, \boldsymbol{x}_j, \Omega_k)$ is the distance between the kth antibody y_k and the jth antigen x_j. This affinity calculation provides an upper limit for affinity values. The affinity will always be between 0 and 1 and will increase linearly as the distance decreases/RR radius increases. This affinity calculation also rewards large antibody radii, but as the RR radii are calculated deterministically instead of randomly mutated in AISLFS, this is less of an issue. Additionally, the antibodies of AISLFS employ local feature selection. Antigens which lie close to an antibody in the feature space (the Euclidean distance is low $\implies$ the affinity value is high) may still not be within the antibody RR, as the the antibody might not consider the feature values which place the antigen close. The local feature selection in AISLFS is further explained in Section 3.1.7.

In VALIS [Karpov et al., 2018], the authors calculate the affinity as:

$$W_{bg} = B\left(\frac{d(b,g)}{r}\right)$$

with

$$B(x) = \begin{cases} 1, & \text{if } x \leq 1 \\ 0, & \text{otherwise} \end{cases}$$

In other words, the Euclidean distance is calculated and divided by the RR radius. If the resulting value is smaller or equal to 1 (the the antigen is within the antibody RR), the affinity is set to 1. If not, the affinity will be 0, i.e. a step function is used. The use of a step function for affinity calculation will not by itself encourage antibodies to move towards antigens in the feature space (which is preferred, as the classification accuracy of the model then likely would increase). If an antigen is within the RR of the antibody, the affinity score between the antibody and the antigen won't increase as the antibody moves towards the antigen. Instead, it will remain 1 until the antigen is no longer within the antibody RR.

Additionally, the use of RR radius in the denominator encourages large antibody RR radii, which might lead to antibodies connecting to antigens of different classes. However, this simplistic affinity measure is counteracted by a sophisticated fitness function which punishes the clustering of antibodies and connection to different-class antigens.

In MAIM [Baug et al., 2019], the affinity measure is defined as:

$$W_{bg} = \begin{cases} \frac{1}{d(b,g)}, & \text{if } d(b,g) \leq \text{r} \\ 0, & \text{otherwise} \end{cases}$$

where r is the RR radius. In other words, affinity is calculated as the inverse Euclidean distance. This encourages antibodies to place themselves as close as possible to antigens in the feature space. Additionally, by not using a step function for affinity calculation (such as VALIS), the affinity values will reflect how close the antibody and antigen in question is, in the feature space. As the affinity values are used extensively in the fitness calculations, this floating number affinity measure has a significant impact on the model. However, this affinity calculation also allows for high affinity values when $d(b, g)$ approaches 0, which might skew the fitness calculations. The affinity calculation doesn't have an upper roof for affinity values, such as VALIS and AISLFS. The authors don't mention the impact or any means of handling this, nor how the affinity is calculated when $d(b, g)$ is 0.

3.1.2 Negative Selection

The biological inspiration and computational strategy of negative selection was introduced in section 2.1.3 and 2.1.7, respectively. When applied to AIS models, the principle of negative selection alter the way antibodies are initialized and evolved, i.e. how they move around in the feature space. In the MINSA model

[Markowska-Kaczmar and Kordas, 2008], the authors attempt to extend the negative selection principle to perform multi-class classification. They do this by defining several antibody sets, which each have some amount of same-class antibodies performing negative selection. In MINSA, these antibodies are generated randomly. This initialization scheme is in contrast to most models, which use antigens to initialize antibodies. This means that antibody feature vectors are initialized to those of various antigens from the training set. The reasoning behind this antibody initialization scheme is to cover the feature space sufficiently for each class. To do this using only antigen-initialized antibody feature vectors, a large number of antigens for each class would be needed. By instead initializing antibody feature vectors randomly and relying on antibody evolution to perform accurate classification, the need for comprehensive training data is reduced.

The overall aim of the antibody evolution in MINSA is to maximize the average affinity towards non-self antigens (antigens with a different class than the antibody) and minimize the average affinity towards self antigens (antigens with the same class as the antibody). If the antibody doesn't connect to any of the self antigens (Euclidean distance to antigen > antibody RR radius), it is added to a set of effective antibodies. If the antibody connects to one or more self antigens, it is mutated and presented with the antigens again. The set of effective antibodies will further be used to detect non-self antigens in the subsequent testing phase. Additional antibodies are continuously generated, so that the model is encouraged to cover a larger portion of the feature space – as the effective antibody set will grow rapidly.

A slightly different method of antibody initialization is used by Laurentys et al., in their negative selection AIS [Laurentys et al., 2010]. The authors use a fixed minimum coverage percentage, i.e. to which degree the antibodies cover the feature space. New antibodies will be generated randomly until this minimum coverage is achieved. In order to prevent antibody clustering, each new antibody is presented to the existing antibodies – to check for overlap. A significant characteristic of the model is that it also includes a limit on how many antibodies there can be in the antibody set. This characteristic means that the algorithm parameters for antibody set size and coverage ratio must be selected carefully, depending on the number of features in the datasets used. Further, the existence of a limit on antibody count and the avoidance of antibody RR overlap are meant to deal with the issue of MINSA where a disproportionately large amount of antibodies were needed to perform accurate classification. However, as the authors don't test their model on any benchmark datasets, the full effect of these strategies are hard to measure and compare to that of MINSA.

3.1.3 Mutation Strategies

The antibody mutation strategy may vary based on whether the feature vector consists of bits or real-value numbers. However, the usual strategy involves adding or subtracting a random normalized value to the feature value, with some probability. In VALIS [Karpov et al., 2018] and MAIM [Baug et al., 2019], this probability is equal to $1/(1+n_f)$, where n_f is the length of the feature vector. In VALIS, the RR radius is mutated by using a random multiplier from a log-normal distribution, while the feature values are mutated by adding/subtracting a random number from a log-uniform distribution. This is difference in distribution function is attributed to the large effect of mutating RR radius, compared to mutating feature values. Additionally, in VALIS, each antibody is put through the mutation process again, until at least one mutation occur. This ensures that the feature vector and/or RR radius of the antibody clones are not equal to those of their parent. In MAIM, this strategy of repeating the mutation process is not adopted, as individually evolving antibody populations on isolated islands provide enough diversity to the main antibody population.

In AISLFS, a different mutation strategy is used. Unlike in most other AIS models, the feature values are not changed [Dudek, 2012]. Rather, mutation only affects the feature subset, randomly including or excluding one of the features. As a result, each clone will have a different feature subset than its parent. This makes for a RR shape strategy where each antibody has an individual geometrical RR shape, depending upon which feature values are included or not. This strategy also renders the antibodies static in the feature space, as the feature values themselves aren't changed. This means that the model relies on accurate training samples to place the antibodies strategically in the feature space (antibodies are initialized from training set antigens).

3.1.4 Fitness Calculation

Although most AIS model don't explicitly employ fitness functions, they usually have some measure of antibody performance, used to evolve the antibody population. Ji et al. utilize a fitness function in their model, providing a fitness score for each antibody based on the antibody RR volume, number of connections and degree of overlap with other antibody RRs [Ji and Dasgupta, 2004]. The reasoning behind including degree of overlap in the fitness function is to promote diversity in the population and avoid antibody clusters in the feature space.

In AISLFS [Dudek, 2012], the fitness of each antibody is measured by looking at the antibody's avidity. The avidity is defined as the combined affinity between the antibody and all the same-class antigens within its recognition region (RR), introduced in Section 3.1.1. Although relatively simple, this fitness measure contribute to producing good accuracy results. However, this fitness evaluation must be considered in light of the nature of the model, where only a subset of feature values are considered at each antibody.

The authors of VALIS [Karpov et al., 2018] employ an explicit fitness function in their model, serving as the core of the model. The fitness is calculated for each antibody separately, based on all the available antigens in the training set. This process repeats each generation. The VALIS fitness function is based on several parameters, including affinity to antigens, relative accuracy (connected to same-class or different-class antigens) and sharing factor. The role of the sharing factor is the same as of the overlap calculation in the previously introduced model by Dasgupta et al. [Ji and Dasgupta, 2004]. This sharing factor is included to promote diversity and punish antibodies connected to the same antigens. If several antibodies are connected to the same antigen, they will only receive a portion of the "reward" and thus antibodies are encouraged to explore the feature space further.

3.1.5 Selection Strategies

The usual choice of selection strategy in AIS models is the one used in Clonal Selection Algorithm (CLONALG) [De Castro and Von Zuben, 2000]. As mentioned in Section 2.1.7, this strategy is based on the process of natural clonal selection in the biological immune system. This process is revolved around a loop, in which the antibodies are exposed to a (single) random antigen. Each iteration, the affinity between each antibody and the antigen is calculated. Further, the n antibodies with highest affinity with the antigen are selected. These antibodies are then cloned proportional to their affinity (higher relative affinity means more clones) and then mutated inversely proportional to their affinity (lower relative affinity means less mutation). Finally, the affinities of all the clones are calculated and the best-performing ones replace the worst-performing antibodies. Models utilising the CLONALG strategy for mutation includes AIRS [Watkins et al., 2004] and AISEC [Secker et al., 2003], among others.

Although being one of the most popular algorithms used to implement AIS models, CLONALG is said to be sub-optimal for classification tasks [Sharma and Sharma, 2011]. Sharma and Sharma instead propose an altered version of CLONALG, named CLONAX. The algorithm differs from CLONALG in that in the memory set of antibodies (the set used for classification), the antibodies are valued for their accuracy. This means that antibodies which connect to few or no antigens of different classes are favoured. To find antibodies which satisfy this criteria, antibody clones are put through a filtering process. The authors compare their CLONAX algorithm to CLONALG on 5 different datasets, showing that CLONAX generally produces more stable results across all the datasets.

In AISLFS [Dudek, 2012], every antibody generates n clones each iteration, which then undergo mutation inversely proportional to their fitness. The antibody clones are evaluated according to how many same-class antigens they bind to and the best-performing one will replace the parent. This replacement happens regardless of whether the clone performs better than its parent, to boost diversity and prevent premature convergence. By rewarding clones which bind to a large amount of antigens, an optimization process is created, where the number of antigens within the antibody RRs are maximized and the number of features used is minimized. Additionally, apoptosis (death) of antibodies is proposed. This elimination process kills off redundant antibodies by iteratively removing antibodies until removing more will decrease the accuracy of the classifier. However, this process includes sequential backward selection loops and therefore significantly increases the time complexity of the algorithm.

As outlined in Chapter 2, crossover is a computational way to mimic biological breeding. Most AIS models don't utilize crossover, but rather depend solely on cloning and mutation to boost diversity. The AIRS [Watkins et al., 2004], AISLFS [Dudek, 2012] and CLONALG [De Castro and Von Zuben, 2000] models are examples of state of the art AIS models utilising only mutation. However, VALIS [Karpov et al., 2018] and MAIM [Baug et al., 2019] use crossover in their models. In VALIS, uniform crossover is employed on both the antibody feature values and the antibody RR radius – recombining successful antibody parents. However, the authors don't present significant arguments as for why they chose to do so and subsequently don't debate the effect of the crossover operation on the results.

Another notable strategy adopted in VALIS is that the selection of antibody for crossover is carried out through the use of tournament selection, where the antibodies compete based on their individual fitness – calculated from the fitness function. Crossover is only performed between antibodies of the same class, which means that each tournament is ran with a predetermined amount of same-class antibodies. Finally, the tournament winner is chosen for reproduction with another tournament winner with the same class.

In MAIM [Baug et al., 2019], the architecture of the model is largely based on that of VALIS, but built on top of an Island Model topology. Antibody populations evolve on isolated islands, with periodical migration in between them. The island topology calls for a crossover operation, in order to reap the benefits of this topological choice. If no crossover operation was used, the island migration would have less effect.

3.1.6 RR Radius Initialization

Another interesting strategy of AISLFS, is the RR radius calculation. Each generation, the RR radius is set to the largest possible value, without including any antigens of a different class in the RR. As each antibody only considers a subset of the feature values, the RR radius is only considered for the individually selected feature values for each antibody. Notably, the RR radius is re-calculated in the same way each iteration. This means that the antibody radii are not mutated randomly throughout the runtime as is usual in AIS models, but rather they are calculated in a deterministic manner.

In VALIS [Karpov et al., 2018] (and MAIM [Baug et al., 2019]), a different RR initialization scheme is used. Here, the RR radius of each antibody is initialized to the Euclidean distance to a random same-class antigen in the training set. The reasoning behind this strategy is to increase the probability that all antigens are covered by at least one antibody, i.e. the antigen is within at least one antibody's RR. The downside of this strategy is that the antibodies might (at least initially) cover several antigens of a different class, especially if the antigens are not particularly clustered corresponding to their class, in the feature space. If the model don't allow enough room for mutation and local search, this could lead to poor classification accuracies. However, in MAIM, the authors saw significant improvements in accuracy after this initialization scheme was implemented, replacing a random initialization (where the RR radii were set to a random floating number >0).

3.1.7 Recognition Region Strategies

Although spherical (referred to as hyperspheres when number of dimensions >3) RRs generally are used in AIS applications, there are alternative geometrical RR shapes. The optimal RR shape may be very problem dependent [Hart, 2005]. Subsequently, Hart et al. show that both the accuracy and convergence of AISs may be affected by the RR shape choice. This is due to the difference in how many antigens each antibody connects to, as well as the ratio of how many of these antigens have the same class as the antibody. The placement of the antigens in the feature space may encourage a RR shape different than the traditional hypersphere.

Another approach is that of previously mentioned AISLFS [Dudek, 2012], where each antibody performs individual (local) feature selection. Local feature selection refers to antibodies individually selecting their own subset of the feature space. This means that each antibody clone performs mutations on their feature vectors, flipping each feature value on or off with a predetermined probability. If an antibody feature value is switched off, it will not be considered when calculating affinity, i.e. the corresponding feature value in the antigen which is compared to the antibody will also not be considered. This strategy results in unique RR shapes for each antibody.

For high-dimensional feature spaces, using only a subset of the features may boost performance significantly, without decreasing accuracy [Dudek, 2012] [Baug et al., 2019]. By having unique RR shapes for each antibody, a accuracy boost may be achieved, as well as reducing the effect of the aforementioned "curse of dimensionality". Additionally, feature selection is typically carried out globally in ML models, determining a single feature set for the entire model. This approach is good for simplicity and efficiency, but does not take into account that different features have variable importance in different regions of the feature space. With local feature selection, the model is tuned to adapt better to the varying importance of each feature. It is important to note that these benefits are mostly prevalent for high-dimensional datasets, i.e. datasets with large amounts of features.

3.1.8 Class Prediction

Class prediction strategies refer to the strategies used to assign class labels to antigens, after the training phase of the model is complete. In AISLFS [Dudek, 2012], the authors present three different methods of class prediction. The first one works somewhat similar to a k-Nearest Neighbors approach, simply summing up how many antibodies which cover the antigen, for each class. For instance, if an antigen is covered by two antibodies with class A and one with class B, the antigen is assigned class A. Although performing surprisingly well, this approach is outperformed by the two other approaches proposed. The second approach relies on summing up the affinities of all the antibodies covering the antigen, for each class:

$$v_{ag,k} = \sum_{ab \in B_k} W_{ab,ag}$$

where k is the class, B_k is the set of antibodies with class k and $W_{ab,ag}$ is the affinity between antigen ag. The third approach is defined as:

$$v_{ag,k} = s_{n_k}(ag)$$

where n_k is the number of antibodies with class k and s_{n_k} is the algebraic sum defined recursively as:

$$s_{n_k}(ag) = W_{ab_k,ag} + s_{n_k-1}(ag) - W_{ab_k,ag} \cdot s_{n_k-1}(ag)$$

In other words, the approach includes applying the probabilistic OR operation to all the affinities between the antigen ag and the connected antibodies with class k. The second and third approaches score roughly the same in terms of accuracy results, alternating on producing the best scores for different datasets.

In VALIS [Karpov et al., 2018], the vote of antigen ag to class k is calculated as:

$$v_{ag,k} = \sum_{ab \in B_k} W_{ab,ag} \cdot A_{ab}$$

where B_k is the set of antibodies with class k, $W_{ab,ag}$ is the affinity between antigen ag and an antibody with class k and A_{ab} is the weighted accuracy of antibody ab. In other words, this approach is similar to that of AISLFS, except for the weighted accuracy component. The weighted accuracy is defined as the total affinities of an antibody to same-class antigens, divided by the total affinities from the antibody to all of the antigens within its RR – further explained in 4.2.7. The inclusion of the weighted accuracy in the voting scheme is not further explained by the authors, but it provides each antibody with a "credit" – indicating its trustworthiness. If an antibody has a low weighted accuracy, this means that it's connected to at least one antigen of a different class. This means that the antibody cannot necessarily be trusted and its vote should subsequently bear less weight. Finally, the antigen is assigned the class which corresponds to the highest cumulative vote tally.

3.1.9 AIS-Based Spam Detection

Applications such as spam and fake news classification may be a niche in which AIS models perform superior to other machine learning techniques. This lack of an AIS niche has been a common critique of AIS models [Baug et al., 2019]. Generally, AIS models are outperformed by established state of the art ML algorithms for multi-class classification tasks, but promising results has been found for AIS models performing binary text classification [Secker et al., 2003] [Idris and Selamat, 2011].

The nature of spam and fake news differ in a lot of ways, but are still eligible for comparison as the overall architecture of the AIS model might remain the same. The main difference is related to the features used to construct antibodies and antigens, as well as the pre-processing necessary to obtain these features.

As the exact features used for spam classification aren't necessarily relevant for fake news classification, this section will look at spam-classifying AIS models as a whole, investigating model architecture and features used.

Representation and Affinity Calculation

The AISEC model is an AIS adapted for classifying spam e-mails [Secker et al., 2003]. The model adopts an algorithm similar to negative selection. Antibodies are single-class (spam samples) and are represented as two-part feature vectors – for each e-mail sample. Each field contains a bag-of-words of the subject and sender field of the e-mails, respectively. As the feature vector don't contain numerical values representing various aspects of the text, but rather all the words included in the stated fields, the affinity cannot be calculated as the Euclidean distance. Instead, the affinity strategy adopted is word-matching against a pre-existing list of words likely to be related to spam. Further, for every word in the antigen vector fields, the AISEC algorithm checks for an exact (case insensitive) match in the antibodies' respective feature vectors [Secker et al., 2003]. The affinity is then calculated as the number of words matching, divided by the length of the feature vector, i.e. the blacklisted words' term frequency.

Oda and White [Oda and White, 2005] adopt a different strategy for feature representation, using regular expressions as antibody features. This means that the antibodies have individual combinations of regular expression rules, matching (connecting to) different antigen samples. This means that antibodies either connected to a given antigen or not (depending on the regular expression match) – strength of connection was not considered. The model use negative selection and as such, all the antibodies seek to connect to spam-labelled antigens. Each antibody also has two weights, counting the cumulative number of spam and non-spam messages matched, respectively. These weights are used to iteratively remove poor-performing antibodies from the population.

An alternative affinity measure is presented in Mahmoud and Mahfouz' SMS spam filtering model [Mahmoud and Mahfouz, 2012]. In their model, the authors use spam and non-spam SMSs as antibodies. The affinity is calculated as:

$$f_j = \frac{S_f/TS}{(S_f/TS) + (H_f/TH)}$$

where S_f and H_f represent the number of matches with labelled antibody spam and non-spam SMSes, respectively. The affinity is calculated for each token in the antigen SMS. Additionally, TS and TH represent the total number of spam and non-spam (SMS) antibodies, respectively. If this affinity exceeds a predetermined threshold (the RR radius), the antibody connects to the antigen.

In the 2019 paper [Saleh et al., 2019], Saleh et al. propose a spam e-mail spam classification model, loosely based on the negative selection AIS architecture. Similarly to AISEC [Secker et al., 2003], the authors employ a blacklisting strategy. However, the model is not an AIS in the traditional sense. Instead of using data samples as antibodies, evolved through mutation or crossover operations, they use three subsequent filters which they refer to as detectors. Each antigen is fed through these detectors, which perform lookups to spam and non-spam databases to determine the class of the antigen. The detectors themselves are not evolved as the model is trained, but the spam and non-spam databases are continually updated with words and N-grams from the labelled antigen samples.

Feature Extraction

The antibody representation of AISEC [Secker et al., 2003] was presented 3.1.9. Antibodies are single-class and represented as two-part feature vectors, containing a bag-of-words of the subject and sender field of the e-mails, respectively. Conversely, the actual message field of the e-mails aren't included in the feature vector. The problem of such a simple strategy is that it does not take into account the message field (leading to easy avoidance by the spammers), as well as the need for an updated blacklist. The authors state that an increase in accuracy may be achieved by adapting more complex antibody features such as measuring term importance with inverse document frequency (TF-IDF) or including the message field text into the feature vector. They also propose that a more elaborate affinity function may further improve the performance of the model.

As mentioned, Oda and White [Oda and White, 2005] used regular expressions as feature representations for the antibodies in their e-mail spam classification model. These regular expressions were constructed by using randomly recombined information from an heuristic library. The heuristic library consisted of regular expressions, selected specifically for their high occurrence in spam e-mails. These regular expressions include matches for specific HTML-formatting and certain words appearing fairly close in the e-mails, in addition to matching individual words. The authors argue that this results in more useful antibodies, capable of achieving good accuracy results with fewer antibodies. The testing performed supports this statement, as the model produced promising results with relatively small antibody sets. With antibody features represented as regular expressions, evolution of the antibody population is performed by randomly adding a regular expression from the library to an antibody, with some probability.

Mahmoud and Mahfouz used spam and non-spam SMSs as antibodies in their SMS spam filtering model [Mahmoud and Mahfouz, 2012]. Upon antibody initialization, the message fields of the SMSs are tokenized and run through a stop word removal filter. Tokenization allows for easier comparison with antigen samples. This comparison is performed by using the aforementioned affinity calculation. Further, the affinity is then used to calculate the *Spam Score*:

$$Spam\ Score = \frac{\sum_{f=0}^{n} f_j}{NT}$$

where n is the number of antibodies connecting to the antigen, f_j is the affinity and NT is the number of tokens (words) in the antigen sample. The Spam Score is the deciding classification factor, labelling the antigen as spam if it surpasses some predetermined threshold. Arguably, this model is more robust than AISEC, due to considering the message field and not just e-mail subject and sender fields.

The model of Saleh et al. [Saleh et al., 2019] illustrates the relative feature simplicity of modern AIS models adapted for spam detection. The features tend to consist of relatively simple word counting strategies or lexicon lookup strategies. As mentioned, the model uses three filters (detectors) as antibodies, which each antigen is fed through in order to determine the class. These detectors use lexicon look ups for N-grams and single words appearing in the message field. By using N-gram lookups, the model can determine that the antigen is indeed spam with a high degree of confidence, if there is a match. This due to the low probability of a random N-gram match. The downside of this approach is that going through the text body looking for both single and multiple word matches in the token databases requires extra computation. As the detector filters themselves are static, the actual training of the model is this process of constructing the lexicons. These spam and non-spam lexicons are constructed from labelled e-mail samples, as the model is trained. Keywords and N-grams are added to the corresponding lexicon if their TF-IDF frequency surpasses a predetermined threshold.

3.2 Semantic-Based Fake News Classification

This section is dedicated to providing the reader with a look at the current state of the art within the field of fake new classification using semantic approaches, in context of relevance to AIS-based fake news classification. The emphasis will be on the linguistic features used and not on the nature of the machine learning models employed – as these are of less relevance to the proposed model.

When it comes to the identification of fake news articles based on their writing style, proposed models range greatly in terms of complexity. Some models aim to use relatively simple word count and term frequency techniques, while others employ complex linguistic models to identify degree of subjective bias, moral foundation and psycholinguistic cues (writing characteristics providing insight into the emotional state of the author) [Klyuev, 2018]. As such, this section is divided into different strategies implemented in various fake news classification models.

3.2.1 Term Counting Strategies

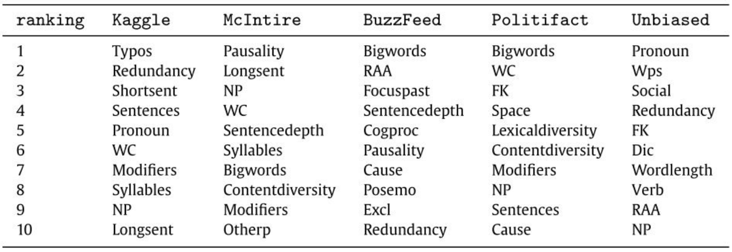
The authors of the 3HAN model [Singhania et al., 2017] found that simple word count based models, which don't consider word ordering or semantics, performed suprisingly well. This suggests that vocabulary and word usage patterns are quite distinguishable when comparing fake and real news articles – indicating good classification potential for a model lacking complex semantic pre-processing.

In their study [Rashkin et al., 2017], Rashkin et al. conducted an analytic study on the language used in news articles. The authors compared the linguistic characteristics of real news with that of satire, hoaxes and propaganda. Using frequency analysis of words belonging to specific linguistic groups, the authors identified clear characteristics of the news articles labeled as untrustworthy. Perhaps not surprisingly, the findings indicated that swear words appeared 7 times as frequently in fake news articles as in trusted ones. More interestingly, second person pronouns – (”you”, ”your” and ”yours”), appeared 6.7 times as frequent, and modal and action adverbs appeared 2.6 and 2.2 times as frequent, respectively. Such significant differences are excellent for precise separation and are, as such, promising for the proposed model. However, some external linguistic lexicon is needed to place the words into the correct groupings.

Volkova et al. tested a wide variety of linguistic features in their models [Volkova et al., 2017]. The models were used to perform binary classification on Twitter news posts. In their models, they found that incorporating linguistic features into the models boosted the classification result significantly. These linguistic features included bias cues, subjectivity cues, psycholinguistic cues and moral foundation cues. Bias was identified through the use of assertive verbs (aggressive action verbs, such as ”persuaded”), factive verbs (presupposing truth) and implicative verbs (implying truth). The frequency of these words were used as feature values in the models. Concerning subjectivity, the authors made use of external lexicons to identify strongly subjective words and opinion words. The psycholinguistic cues and moral foundation cues were extracted using NLP techniques and are further discussed in Section 3.2.3.

Rashkin et al. also found that the inclusion of numbers were more than half as likely in fake news articles as in real ones. This characteristic was also found in a study by Meel and Vishwakarma, where the authors ground this finding on fake news articles lacking concrete numerical data on which to base the articles [Meel and Vishwakarma, 2019]. Meel and Vishwakarma also extend this characteristic to textual data, i.e. proper nouns such as the name of a person or a place. These are concrete textual facts that fake news articles seem to often lack. However, Horne and Adali conclude exact opposite, stating that fake news articles use significantly more proper nouns than real ones [Horne and Adali, 2017].

In their benchmark study, Gravanis et al. looked at which linguistic features and word embedding strategies could lead to high accuracy results for fake news classification [Gravanis et al., 2019]. The authors found that the best performing classification was obtained using the union of all features proposed from three different papers. These features include big words, word count, sentence depth, pausality and pronouns, among others. However, this feature set consisted of almost 60 features in total – which might lead to reduced efficiency. Subsequent feature selection could possibly decrease the amount of features and therein run time. Additionally, the authors performed feature ranking to determine the top ten features for each dataset. These features are shown in Figure 3.1. Due to the variability of usefulness of the various features (for the different datasets), it is hard to precisely identify the best-performing features overall. However, some features such as total word count, pronoun usage, sentence length and frequency of big (long) words appear to work well across several of the datasets and were as such considered to be relevant for the proposed model.



ranking	Kaggle	McIntire	BuzzFeed	Politifact	Unbiased
1	Typos	Pausality	Bigwords	Bigwords	Pronoun
2	Redundancy	Longsent	RAA	WC	Wps
3	Shortsent	NP	Focuspast	FK	Social
4	Sentences	WC	Sentencedepth	Space	Redundancy
5	Pronoun	Sentencedepth	Cogproc	Lexicaldiversity	FK
6	WC	Syllables	Pausality	Contentdiversity	Dic
7	Modifiers	Bigwords	Cause	Modifiers	Wordlength
8	Syllables	Contentdiversity	Posemo	NP	Verb
9	NP	Modifiers	Excl	Sentences	RAA
10	Longsent	Otherp	Redundancy	Cause	NP

Figure 3.1: *Top performing features for various fake news datasets* [Gravanis et al., 2019]

In their 2019 paper, Ozbay and Alatas adapted two bio-inspired optimization models for classifying fake news articles [Altunbey Ozbay and Alatas, 2019]. Notably, their models produced promising results with relatively simplistic features. During the pre-processing phase, the authors performed tokenization, stop word removal and word stemming (reducing words to their base form). Term frequencies were then calculated using a predetermined set of terms. However, they don't specify which words they compute the term frequencies for, other than stating how many they used for the various datasets. Nevertheless, the obtained results hint at the effectiveness of stop word removal and word stemming in the pre-processing phase, for fake news classification.

Grammatical Quality

Grammatical quality of writing might also be indicative of the truthfulness of news articles [Klyuev, 2018]. Burgoon et al. used Flesch-Kincaid grade level as a feature in their classification model, to assess the grammatical quality [Burgoon et al., 2003]. Flesch-Kincaid grade level provides an indication of how difficult an english text is to understand, in the form of a score between 1-100. The score is calculated according to a formula, using the total number of words, sentences and syllables in the text. The authors found that truthful news articles in general had higher Flesch-Kincaid scores, in addition to more complex sentence structures (longer sentences) and more syllables per word – than deceitful articles.

Another measure of the grammatical quality is the percentage of grammatically incorrect sentences, as well as the lack of advanced (uncommonly used) words – indicating a less advanced vocabulary. Syntactic analysis tools (tools analysing the linguistic structures) such as Stanford CoreNLP can be used for this purpose [Klyuev, 2018]. CoreNLP can be used to generate a score of the grammatical quality of a text – which further can be used as a numerical feature value.

3.2.2 Differential Weighting

News article headlines are often designed to shock, surprise or tempt the reader to continue reading [Volkova et al., 2017]. The authors of the 3HAN model [Singhania et al., 2017] aimed to take advantage of this characteristic, by weighting the headline more than the remaining article, when extracting features. 3HAN is based on three distinct levels, concerned with the words, sentences and headline of the articles, respectively. These levels are combined into a news vector, by processing articles in a bottom-up manner. The authors justify this division by claiming that the headlines of news articles is known as a distinguishing feature. Subsequently, a relatively small subset of words and sentences (those in the headline) carry a disproportional value for classification and should therefore be weighted more.

The authors of 3HAN further maintain that this difference of importance means that the relative importance of a news article can be modelled as an inverse pyramid (where the width illustrates relative importance), with the headline at the top. They infer this from the repetitive nature of fake news articles, where the text corpus is often repeated with slightly different semantics. This is supported by Horne and Adali, who conclude that fake news articles use repetitive content in the corpus and rather pack long sentences into the headline [Horne and Adali, 2017]. Considering the nature of *clickbait* articles, this is inherently intuitive, as the fake news authors wish to attract readers by using shocking headlines.

Considering the 3HAN model [Singhania et al., 2017], this means that the headline and initial paragraph should be considered as more important (in terms of classification) than the rest of the news article. The authors implement this weighting by assigning attention weights to sentences and words depending on their placement in the article. The authors attribute the high classification accuracy to this differential importance weighting. Additionally, they conclude that their headline premise is valid, as the model which include the headline weighting scores better than the one without (although the difference is less than 1%).

In practice, this differential weighting strategy could be implemented into the proposed model by multiplying term frequencies found in the headline by some factor – so that these carry more weight than if they're found in the article corpus. Feature values of antibodies and antigens would then be changed correspondingly and would therein change position in the feature space – potentially simplifying classification.

3.2.3 Use of External NLP Tools

The Stanford CoreNLP software was introduced in Section 3.2.1. The software can also be used for a wide variety of text analytics. In the 3HAN model [Singhania et al., 2017], the authors used CoreNLP to tokenize sentences and perform word stemming, followed by grouping of the words. Used in this way, the software can aid the pre-processing phase of feature extraction – using built-in libraries for word stemming etc. Furthermore, Stanford CoreNLP also includes tools for performing sentiment analysis of text corpus. Sentiment analysis is based around extracting the expressed opinion and subjective state of the text, i.e. determine if the text is inherently positive or negative. As shown by Meel and Vishwakarma, multiple models have found success in using sentiment analysis to aid fake news classification [Meel and Vishwakarma, 2019]. One of these is the fake news detection model by Bhutani et al., where the authors found that extracting sentiment from the news articles improves classification accuracy [Bhutani et al., 2019]. The authors further hypothesized that the sentiments found in the articles could serve as a pivotal deciding factor for classification, as the sentiments extracted from fake news articles generally were more negative than those extracted from real ones.

In the aforementioned models of Volkova et al., LIWC was used for identifying psycholinguistic cues [Volkova et al., 2017]. LIWC (Linguistic Inquiry and Word Count) is a language psychology analysis framework [Pennebaker et al., 1999]. The psycholinguistic cues include the use of persuasive and biased language in the text. For example, LIWC picks up on rhetorical questions and emotional language to identify these persuasive cues. The authors also used LIWC to extract moral foundation cues, which refer to the ethical principles on which the news articles are built on. The accuracy results improved significantly when these cues were used to aid classification – leading to the hypothesis that fake news articles appeal to different ethical foundations than verified news. In practice, for the proposed model, LIWC could be used to compute a category in which the article is believed to belong to (in terms of moral foundation and psychological state of the author). Further, some hash-function could map these categories to real-valued numbers – which could then be used to aid classification.

In the benchmark study by Gravanis et al., the accuracy score was improved when additional word embedding features, generated with *word2vec* [Mikolov et al., 2013], were used – boosting the accuracy across all the testing datasets [Gravanis et al., 2019]. The authors used a pre-trained model (trained with Google News articles) to generate a fixed-length vector for each news article. This allows numeric representation of each news article, with the hope of capturing the symbolic meaning of each article – so that similar articles are placed in clusters in the feature space. These results highlight the promising capabilities of applying text embedding models to the classification task.

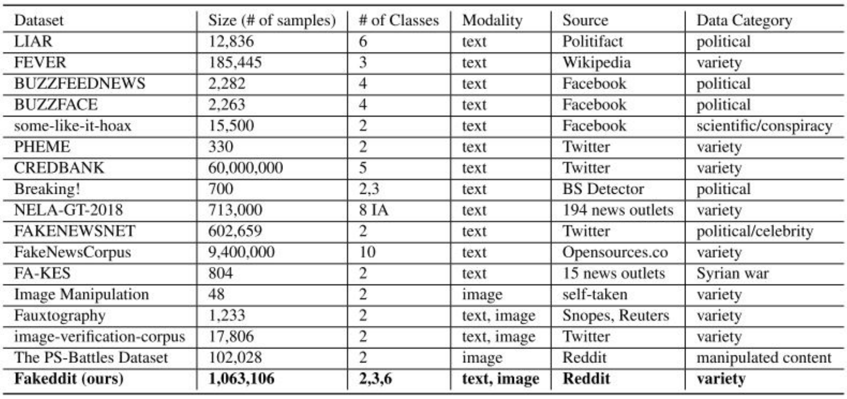
More recent developments in the field of text embedding includes the ”Bidirectional Encoder Representations” (BERT) [Devlin et al., 2019] and ”Embeddings from Language Models” (ELMo) [Peters et al., 2018] models. These models build context-dependent vectors rather than vectors representing the meaning of each word (like word2vec does). This means that the models evaluate the setting in which each word appears, when the meaning of the word is computed – leading to much more accurate vector representations. For instance, the word ”watch” has vastly different meanings in the two sentences: ”I looked at my watch” and ”I love to watch movies”. Word2vec would assign ”watch” the same vector in both these instances, while ELMo and Bert would not. BERT is also capable of representing entire sentences as fixed-length floating number vectors, which is highly relevant to the proposed model.

The authors of Fakeddit [Nakamura et al., 2020] introduced a model which classifies Reddit news submissions. In their model, the authors used BERT to generate sentence embeddings for each submission title. This way, the authors were able to represent the submission title as fixed-length vectors, which were then used directly for classification. The results obtained further support the usefulness of text embeddings for fake news classification.

The authors of FakeBERT proposed an approach to fake news classification using solely BERT embeddings [Kaliyar et al., 2021]. BERT processes the input sentence from both left-to-right and right-to-left, where other word embedding models generally only does one of the two. The authors theorize that this bidirectional training approach is excellent for modelling the relevant information of fake news articles, as both semantic and long-distance dependencies in input sentences are captured. The authors used a sequence length of 512 tokens as input to their pre-trained BERT model, which is the maximum available input sequence. However, they don't specify how they extract these 512 tokens from the full news article samples (which likely are longer than 512 tokens), other than that they extract embeddings for a sentence or a set of words.

3.2.4 Relevant Datasets

As fake news is a relatively recent phenomenon, there are no clear-cut benchmark datasets used across the various research articles. Some authors use manually labelled websites from sites such as PolitiFact [Wang, 2017] [Singhania et al., 2017], while other use labelled Twitter news posts [Gravanis et al., 2019]. Additionally, many researchers construct their own datasets for evaluating their models. This inconsistency makes for difficult comparisons between models. The sample size, source and number of classes may also vary between the different datasets, as can be seen in Table 3.2.



Dataset	Size (# of samples)	# of Classes	Modality	Source	Data Category
LIAR	12,836	6	text	Politifact	political
FEVER	185,445	3	text	Wikipedia	variety
BUZZFEEDNEWS	2,282	4	text	Facebook	political
BUZZFACE	2,263	4	text	Facebook	political
some-like-it-hoax	15,500	2	text	Facebook	scientific/conspiracy
PHEME	330	2	text	Twitter	variety
CREDBANK	60,000,000	5	text	Twitter	variety
Breaking!	700	2,3	text	BS Detector	political
NELA-GT-2018	713,000	8 IA	text	194 news outlets	variety
FAKENEWSNET	602,659	2	text	Twitter	political/celebrity
FakeNewsCorpus	9,400,000	10	text	Opensources.co	variety
FA-KES	804	2	text	15 news outlets	Syrian war
Image Manipulation	48	2	image	self-taken	variety
Fauxtography	1,233	2	text, image	Snopes, Reuters	variety
image-verification-corpus	17,806	2	text, image	Twitter	variety
The PS-Battles Dataset	102,028	2	image	Reddit	manipulated content
Fakeddit (ours)	**1,063,106**	**2,3,6**	**text, image**	**Reddit**	**variety**

Figure 3.2: *Overview of various fake news datasets and their attributes* [Nakamura et al., 2020]

Small and single-sources datasets might not serve as accurate representations of news articles found "in the wild". This highlights the difficulty of dataset selection, for fair and realistic model testing. This section will introduce relevant datasets for fake news classification, along with an assessment of suitability with the proposed model. Only the Liar, FakeNewsNet and Fakeddit datasets from Table 3.2 will be discussed in detail. The reasoning behind this is that many of the datasets listed were only used by the creators or were focused on other aspects of classification, like image recognition. As mentioned in Section 3.2.1, Gravanis et al. determined the top features for a variety of datasets in their benchmark study [Gravanis et al., 2019]. Referring to Figure 3.1, the difference in feature performance between the datasets supports the authors' claim that single-sourced datasets could be biased and therefore unfit for training.

The Kaggle fake news dataset is a large-scale dataset originally posted as a classification competition on the Kaggle website, not to be confused with the dataset referred to in Figure 3.1. The dataset includes binary class labels (reliable and unreliable), article text, title and author. The article samples in the dataset are real-world news articles, propagated during the 2016 U.S. presidential election. The dataset has been used in several models since its release in 2018, which makes the dataset fit for comparing accuracy results with other models.

The Fakeddit dataset [Nakamura et al., 2020] is a massive-scale dataset, consisting of more than 1 million individual samples – gathered from Reddit. These samples not only contain the article corpus, but also images, metadata and comment data. Although not particularly relevant to the proposed model (which only looks at the headline and article text), these are advantages that might make it suited for the role of a fake news classification benchmark dataset, in the future. The Fakeddit dataset features multi-class labelling. The authors additionally set up the dataset to allow for 2-class, 3-class and 6-class classification – enabling both binary and fine-grained (more than 2 classes) classification. The 6-class labels include true, satire/parody, misleading content, manipulated content (mostly applying to images), false connection and imposter content. The downside of the Fakeddit dataset is that it was released very recently (May 2020), and thus there are few models tested on the dataset yet. This means that there are few models with which to compare classification accuracy to.

FakeNewsNet is a dataset released in 2018, featuring 602,659 short Twitter statements labelled as real or fake [Shu et al., 2018]. The dataset contains a rich set of features, including contextual metadata as well as the tweet texts. The samples are collected from PolitiFact and GossipCop. Similar to Fakeddit, the FakeNewsNet dataset is also quite recent, but multiple researchers have nevertheless used the dataset to test their models. The recency of the datasets indicates that the language used in the text samples are up-to-date with modern writing styles, making it relevant for a model adapted to modern social media.

Lastly, the Liar dataset is a 2017 dataset consisting of 12,836 short political statements [Wang, 2017]. Like the Fakeddit, the dataset features six different class labels, ranging from ”pants-fire” to ”true”. The samples are fact-checked twice and the authors argue that the samples are as unbiased as possible, due to sampling from various sources and balancing the amount of samples per political party affiliation. Additionally, there is roughly the same amount of samples per class label (except for the ”pants-fire” class). Metadata is also included for the samples, including subject, context, speaker, political party etc.

The authors of Liar also tested several state of the art ML models on their dataset. The accuracies of these models were quite poor, generally achieving an accuracy of around 0.20. This is likely due to the nature of using six separate classes, where fine-grained classification is difficult. Needless to say, differentiating between ”mostly-true” and ”half-true” is difficult even for humans and cannot be directly compared with the accuracy scores of models which test on binary class datasets.

Chapter 4

Model and Architecture

This chapter aims to explain the architecture of the proposed model, including the algorithmic approach, relevant parameters and the graphical visualisation tool used for plotting solutions. The source code for the proposed model can be found at: www.github.com/SimenSverdrup/An-AIS-for-Fake-News-Classification

4.1 Datasets

The investigations into relevant datasets (see Section 3.2.4) led to the discovery of three datasets which are generally accepted as valid benchmarks. These are the Kaggle Real-World dataset [Kaliyar et al., 2021], FakeNewsNet (FNN) dataset [Shu et al., 2018] and the Liar dataset [Wang, 2017]. These datasets have been used in multiple models and were released quite recently – which gives the datasets relevance to today's writing styles.

The accuracy results obtained (by the various models) on the three datasets vary significantly. The Liar and FNN datasets are objectively difficult to classify, with Liar accuracies typically spanning the 0.15-0.25 range (for 6-class classification) and FakeNewsNet accuracies spanning the 0.55-0.65 range. Comparatively, accuracies for the Kaggle dataset typically span the 0.80-0.95 range. It was theorized that this variation in dataset difficulty would test the robustness of the proposed model, as well as the breadth of possible applications. The Kaggle dataset would test the model's performance on a typical fake news dataset, Liar would test the proposed model's ability to perform fine-grained classification and FNN would test the model's ability to perform difficult binary classification.

By testing on three distinct datasets with different sample sources, it was believed that the general applicability of the model would be thoroughly tested and the results would subsequently bear weight in terms of general applicability. The performance of classification models may vary greatly depending on which datasets are used for testing and results obtained using single datasets for evaluation should generally be taken with a grain of salt [Bozarth and Budak, 2020]. As such, it was decided that the proposed model should use three distinct datasets for model training and accuracy testing.

4.2 Model Architecture

This section presents the architecture of the proposed model, excluding the features employed (these can be found in Section 4.3). Subsection 4.2.1 introduces the overall structure of the model, to provide an overview of the model. Further, Subsection 4.2.2 presents the model parameters, which may impact the results. The remaining subsections cover the main elements of the model.

4.2.1 Model Structure

The algorithm of the underlying AIS takes its main inspiration from VALIS [Karpov et al., 2018]. However, some attributes have been changed in order to tailor the algorithm towards fake news classification. To address the challenge of rapidly growing feature spaces when text embedding features were used (more about this in Section 4.3.4), a scheme for local feature selection amongst the antibodies was implemented – inspired by that of AISLFS [Dudek, 2012]. Further, the crossover strategy from VALIS was replaced by a mutation scheme – to better evaluate the value of the extracted features. The overall structure of the proposed model can be seen in Figure 4.1.

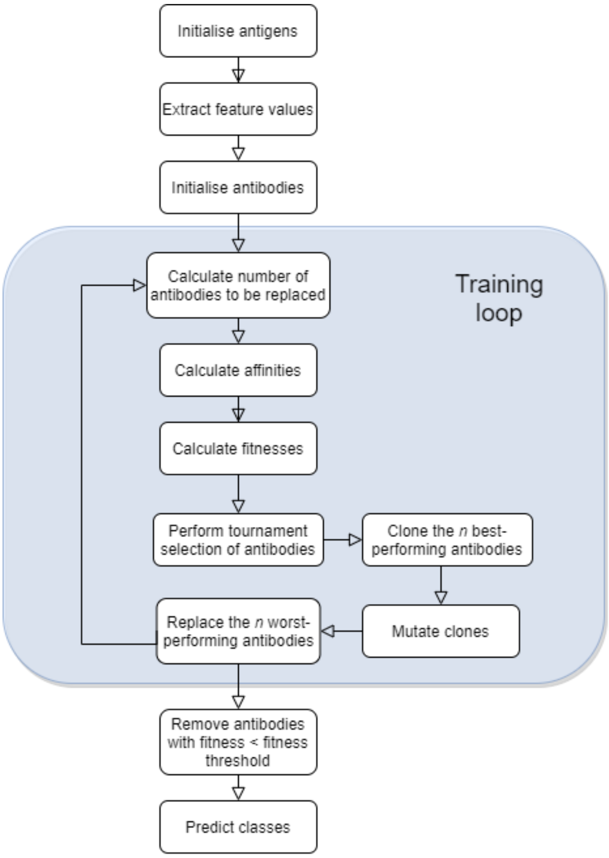

Initialise antigens
Extract feature values
Initialise antibodies
Training loop
Calculate number of antibodies to be replaced
Calculate affinities
Calculate fitnesses
Perform tournament selection of antibodies
Clone the n best-performing antibodies
Mutate clones
Replace the n worst-performing antibodies
Remove antibodies with fitness < fitness threshold
Predict classes

Figure 4.1: *The overall structure of the proposed model*

4.2.2 Model Parameters

The relevant parameters of the proposed model are presented in Table 4.1, together with their respective abbreviations. These abbreviations are used in the following subsections.

Model Parameters	
Parameter	**Explanation**
Cross-validation split (k)	The number of splits used for k-fold cross-validation
Datasets (D)	The set of relevant datasets
Classes (C)	The set of classes for the selected dataset
Total population (P)	The total antigen population of the current training set (before antibodies are initialized)
Generations (G)	The number of generations (iterations)
Feature vector mutation probability (p_{fv})	The probability of a *single* antibody feature value mutation
RR radius mutation probability (p_r)	The probability of mutation on the antibody recognition region radius
Antibody ratio (AB_{ratio})	The ratio of antibodies, compared to the current antigen training set (determining the degree of data reduction by the model)
Antibody initialization split ($AB_{init-split}$)	The ratio of antibodies initialized with antigen featue values vs initialized with random feature values
Antigens (AG)	The set of antigens for the current training split k
Antibodies (AB)	The set of antibodies for the current generation and training split k
Features (F)	The set of features used for the selected dataset
Antibody replacement ratio (AB_{rr})	The maximum ratio of antibodies being replaced (decreases each generation)
Antibody removal threshold (AB_{rt})	The minimum antibody fitness value
Antibody clones (AB_c)	The number of clones each selected (for reproduction) antibody produces
Tournament size (T)	The number of antibodies to compete in each tournament (statically set to 1/10 of the number of antibodies in the antibody set)
Antibody features used (AB_F)	The feature subset used by the antibody

Table 4.1: The parameters of the proposed model

4.2.3 Antibody and Antigen Structures

The structures of the antigens and antibodies in the proposed model are mostly standard (as introduced in Section 2.1.7). Antigens consist of a feature vector and a class label. Antibodies are similarly structured, but also include an additional value stating the radius of the RR region, as well as a vector identifying which features the given antibody uses. A hypersphere was selected as the geometrical RR shape, due to its simplicity and efficiency. Additionally, plotting of the feature values (with the implemented solution plotting tool, see Section 5.1) did not result in any clear alternatives. As an important part of the model is the feature extraction, an alternative RR shape might also skew the results. This is due to features being weighted differently in terms of inclusion into the RRs. Non-hyperspherical RRs would stretch further in some dimensions (i.e., for some features), than others – leading to an unfair evaluation of the features. If maximal accuracy was the only goal of the proposed model, this would be okay, but as the scope of the thesis includes evaluating applicability of features as well, this is sub-optimal.

As mentioned, each antibody employs feature subset selection, inspired by AISLFS [Dudek, 2012]. This selection was implemented through each antibody having a vector which keeps track of which features the antibody uses for calculating affinity to antigens. This means that the antibody might lie close to a given antigen in the feature space, but the antibody might not consider (use) the feature which puts them close to each other. The antigen might then still fall outside the RR of the antibody. Further, the antibody would then not connect to the antigen and subsequently not vote on the antigen class in the class prediction phase, thus not negatively impacting the classification. The vector indicating which features the antibody uses is further referred to as the *feature selection vector*.

The structure of the antibodies can be seen in Figure 4.2. It is important to distinguish the feature vector and the *feature selection vector*. The feature vector contain all the feature values (floating numbers) of the antibody, while the *feature selection vector* contains booleans indicating which features the antibody will actually use. As the length of the feature vectors of antibodies and antigens is static, the structure of the antigens is the same as for the antibodies, but without the RR radius and *feature selection vector*. It is important to note that the antigens don't mutate and thus they stay the same throughout the execution of the algorithm.

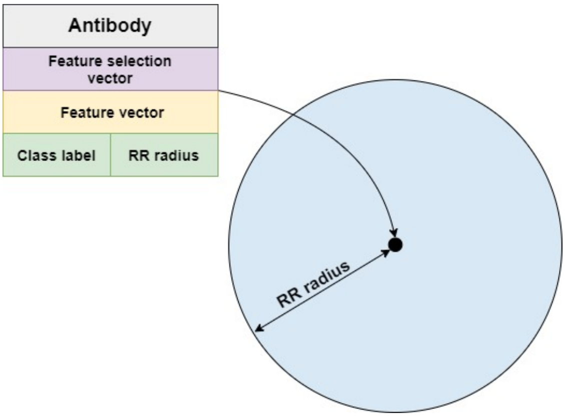


Figure 4.2: *Structure of an antibody with spherical RR, in a 2-dimensional feature space*

Negative selection was ultimately discarded. This was due to the benefits of using this approach being unclear. The antibody fitness and selection scheme would likely not work optimally with negative selection, as antibodies are only slightly punished for including different-class antigens in their RR. It was therefore theorized that the model would struggle with differentiating between self and non-self antigens, especially when the feature values are extracted during runtime (and therein might not be accurately representative of antigen classes). As the proposed model includes a large amount of features, there was also uncertainty tied to how negative selection might work with high-dimensional feature spaces.

4.2.4 Normalization

As the recognition region shape used in the model is an hypersphere, normalization of feature values was necessary – in order to not weigh features unfairly. The initial normalization scheme included Z-Score normalization, as in VALIS [Karpov et al., 2018]. This strategy changes each feature value as:

$$x' = \frac{x - \bar{x}}{\sigma}$$

where x is the old value, x' is the new value, $\bar{x}$ is the average value for x across all the data samples and σ is the standard deviation. However, after some preliminary testing, it was determined that Min-Max normalization resulted in slightly better results. This approach re-scales each value as:

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

where x is the old value, x' is the new value, $min(x)$ is the minimum feature value across all the data samples and $max(x)$ is the maximum feature value across all the data samples. It's important to note that the normalization was carried out for each feature value separately. In other words, when calculating $min(x)$ and $max(x)$, these denote the minimum and maximum value across all the data samples' ith feature vector index, where $i \in [0, length(feature\ vector)]$. Thus, the feature values at the ith and *(i+1)*th indices are considered independent when performing normalization.

Further, some extracted features resulted in negative feature values (see Subsection 4.3.4). In order to normalize these feature values, a simple approach of adding the minimum feature value (before normalization) to all the values was implemented. This approach finds the minimum (most negative) feature value across all the data samples, at a given feature vector index. Then, the absolute value of this feature value is added to all the feature values at this index. This ensures that each feature value has a minimum value of 0. Finally, the Min-Max normalization scheme is used on the new feature values. This normalization strategy results in all the new values being linearly mapped to a number $\in [0, 1]$.

4.2.5 Initialization

The proposed model is initialized with an initial antigen population. For the benchmark (not fake news) classification datasets (see Section 5.2.1), the antigen feature values are simply the normalized (see Section 4.2.4) feature values of the datasets. For the fake news datasets, the antigen feature values are calculated through feature extraction, as explained in depth in Section 4.3. The antigen class labels are set to the class labels of their respective sample in the dataset.

The size of the antibody set is based on the antibody ratio (AB_{ratio}) parameter, which state the ratio between antibodies and antigens in the current training set. For reference, VALIS [Karpov et al., 2018] and AISLFS [Dudek, 2012] use a constant value of 1.0 as AB_{ratio}, meaning that the amount of antibodies is set equal to the number of antigens in the training set. This allows for more fine-grained distinction (which some datasets require), at the cost of no data reduction (the amount of antibodies is equal to number of input samples). This lack of data reduction is acceptable when model testing is conducted with traditional classification benchmark datasets (with sample sizes ranging from 150-800), but is infeasible for the large scale fake news datasets used for testing the proposed model (with thousands of samples). As the feature values are normalized between [0-1] in the proposed model, there is also a disproportional "return of investment" for large antibody population sizes. This is due to not needing additional antibodies to cover antigen outliers – as these would be normalized into the [0-1] range. When small numbers of features are used, it's unnecessary to employ a large antibody population – as a smaller antibody population would still cover the feature space adequately. As such, the (AB_{ratio}) parameter is not defined statically, so that it can be tuned according to the characteristics of the datasets which are used for testing.

Further, the antibody initialization approach consists of two mixed initialization schemes, that of AISLFS [Dudek, 2012] and that of MAIM [Baug et al., 2019], respectively. These two initialization schemes result in two antibody sets, which are then combined into one. This approach was used to reap the benefits of both initialization schemes, without sacrifice, as both schemes showed promising results during the model refinement testing (see Section 5.2.1). For the first antibody set, the feature values and class of the antibodies are set to those of randomly selected antigens from the current training set (see Section 4.2.11). The initialization scheme was implemented so that several antibodies aren't initialized with the same antigen's feature values. This initialization approach is similar to that of VALIS Karpov et al. [2018] and AISLFS [Dudek, 2012], among others.

The second antibody set are initialized similarly to the initialization scheme of MAIM [Baug et al., 2019]. This initialization scheme revolves around randomly assigning feature values and classes to the antibodies. The range of possible feature values are defined as the span from 10% less than the minimum feature value (at a given index) in the training set, to 10% more than the maximum value. Adding this randomly initialized antibody set boosted model performance on the Wine and Diabetes datasets especially – indicating that the model may benefit from this initialization scheme when larger numbers of features are used (the Iris dataset includes 4 features, compared to 8 and 13 for Diabetes and Wine, respectively). The $AB_{init-split}$ parameter was introduced to denote the ratio between the two initialization schemes, i.e. an $AB_{init-split}$ of 0.9/0.1 means that 90% of the antibodies are initialized with the AISLFS approach and 10% with the MAIM approach.

Two different RR radius initialization schemes were considered, that of AISLFS and that of VALIS. In the AISLFS approach, the RR radius of each antibody is initialized to be as large as possible, without including any antigens of a different class in the RR. In the VALIS approach, the RR radius is set to include at least one (random) antigen of the same class, in the RR. This RR radius is used regardless of whether this also allows antigens of a different class inside the RR. After model refinement testing of the two approaches (see Section 5.2.1), it was decided to use the VALIS initialization scheme. Finally, the *feature selection vector* (identifying which features the antibody will use) is initialized to initially consider all the features.

4.2.6 Affinity Calculation

Various possible methods of affinity calculation were presented and discussed in Section 3.1.1. The affinity measure selected for the proposed model was inspired by the affinity measure in AISLFS [Dudek, 2012], where the affinities always remain between 0 and 1. Comparatively, the affinity calculation of MAIM [Baug et al., 2019] would result in the affinities (and therein fitnesses) approaching infinity when the Euclidean distance approached zero. As the proposed model uses an initialization scheme where a part of the antibodies are initialized with the feature values of random antigens (thus having an Euclidean distance of zero to these antigens), this affinity measure would cause problems. As such, the antibody-antigen affinities of the proposed model are inspired by that of AISLFS, calculated as:

$$a(ab_k, ag_j) = \begin{cases} 0, & \text{if } d(ab_k, ag_j) > r_{ab_k} \\ 1 - \frac{d(ab_k, ag_j)}{r_{ab_k}}, & \text{otherwise} \end{cases}$$

where $d(ab_k, ag_j)$ is the Euclidean distance between antibody k and antigen j and r_{ab_k} is the RR radius of antibody k. The Euclidean distance is calculated as:

$$d(ab_k, ag_j) = \sqrt{\sum_{i \in F'} (ab_k[i] - ag_j[i])^2}$$

where F' is the feature set of antibody k (a subset of the total feature set F) and $ab_k[i]$ and $ag_j[i]$ refers to the feature value at index i in the antibody and antigen feature vector, respectively. It is important to note that only the values of features which are included in the antibody *feature selection vector* are used in the Euclidean distance calculation.

Additionally, it should be noted that r_{ab_k} is in the denominator in the affinity calculation. Thus, the problem of antibodies employing excessively large RR radii to increase affinity and therein fitness must be considered. This problem is avoided due to the subsequent fitness calculation, which punishes antibodies which include different-class antigens in their RR, as well as sharing same-class antigens with several other antibodies.

4.2.7 Fitness Evaluation

The fitness calculation is a crucial part of the underlying AIS, serving as the backbone of the proposed model. The fitness function was adapted from VALIS, due to its flexibility and proven efficiency. Additionally, the fitness function's ability to encourage exploration of the feature space, without introducing too much diversity, was considered. The fitness calculation consists of three main components and is calculated as:

$$F(ab) = \frac{Weighted\,Accuracy(ab) \cdot Sharing\,Factor(ab)}{Total\,Affinity(ab)}$$

where ab is the antibody for which the fitness is being calculated. The components are explained further in the following subsections.

Total Affinity

Total Affinity refers to the summed affinities between the antibody and all the (connected) antigens, regardless of their class. As the affinity is 0 between an antibody and any antigen outside its RR (as explained in 4.2.6), the sum cover the entire antigen set G:

$$Total\,Affinity(ab) = \sum_{ag \in G} W_{ab,ag}$$

where ab is the antibody in focus, ag is an antigen, G is the set of all the antigens in the current training set and $W_{ab,ag}$ is the affinity between antibody ab and antigen ag. As this component is in the denominator of the fitness function, antibodies are punished for connecting to a large amount of antigens. Although seeming somewhat counter-intuitive for covering the feature space, this encourages antibodies to "specialize" and locate a smaller subset of antigens which are covered by few or no antibodies. Thus, the feature space coverage is in fact improved.

Weighted Accuracy

The role of the *Weighted Accuracy* is to measure the degree to which the antibody connects to antigens of the same class. The summed affinity between each antibody and its (connected) antigens with the same class label is divided by the antibody's *Total Affinity.* Additionally, Laplacian smoothing is employed to prevent overfitting. This smoothing is altered slightly from VALIS, as VALIS uses $\alpha = 1$ and the proposed model uses $\alpha = 1.5$. The reasoning behind this choice was that preliminary testing with $\alpha = 1$ resulted in high-variance results. Additionally, as the fake news datasets are very large compared to the classification benchmark datasets, it was reasoned that overfitting would be less of a problem for the proposed model (than for VALIS). As such, the *Weighted Accuracy* is calculated as:

$$Weighted\,Accuracy(ab) = \frac{\alpha + \sum_{ag \in G_{ab}} W_{ab,ag}}{k + \sum_{ag \in G} W_{ab,ag}}$$

where k is the number of classes in the dataset and G_{ab} is the set of antigens connected to ab with the same class as ab.

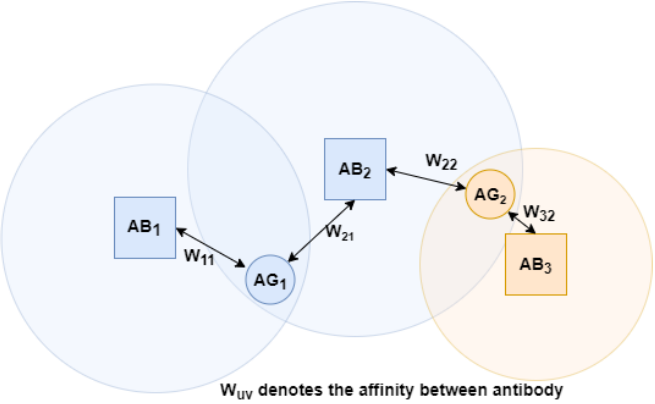
Sharing Factor

The *Sharing Factor* component of the fitness function plays a critical role in the model. The component stimulates the antibodies to continually search their local feature space for antigens, by only giving a portion of the "reward" to each antibody connecting to the same antigen. When several antibodies cover the same antigens, the *Sharing Factor* quickly drops and the antibodies are encouraged to move or reduce their RR. As the worst-performing antibodies are replaced every round, the model would quickly converge towards local optima if the *Sharing Factor* didn't encourage the antibodies to avoid crowding high-concentration (in terms of antigens) regions of the feature space. The authors of MAIM [Baug et al., 2019] (which adapt a similar fitness function) found that this was indeed the case, as the accuracy was severely degraded when the *Sharing Factor* was not used in the fitness function. As such, the *Sharing Factor* is calculated as:

$$Sharing\,Factor(ab) = \sum_{ag \in G} \frac{W_{ab,ag}^2}{\sum_{ab \in B_{ag}} W_{ab,ag}}$$

where B_{ag} is the set of antibodies connecting to antigen ag and the remaining notations are the same as in the previous subsections. The *Sharing Factor* is calculated by squaring the affinity from the antibody to one of its connected antigens and dividing by the sum of all the affinities of the antigen (the affinity between the antigen and all its connected antibodies). This provides a measure of the portion that the antibody's affinity to the antigen make up, compared to all of the antigen's affinities. Finally, this fraction is summed up for all the antigens within the RR of the antibody – resulting in the *Sharing Factor* of that specific antibody.

It should be noted that the affinity in the numerator is squared. This gives the affinity a greater impact on the fitness score – to not give the *Sharing Factor* too much influence on the fitness. As the class prediction scheme (see Subsection 4.2.10) is somewhat based upon several antibodies covering the same antigens, it is not optimal that each antigen is covered by only one antibody. By squaring the affinity in the numerator, antigen sharing is not completely discouraged – striking a balance between discouraging overcrowding and encouraging local search. Overall, the fitness calculation was adapted from VALIS, with the only tweak being the slightly altered *Weighted Accuracy.* The fitness calculation scheme is illustrated in Figure 4.3.



W_{uv} denotes the affinity between antibody u and antigen v.

True classes illustrated with blue color for class X and with orange color for class Y.

$$\text{Sharing factor of } AB_2 = \frac{(W_{21})^2}{W_{21} + W_{11}} + \frac{(W_{22})^2}{W_{22} + W_{32}}$$

$$\text{Total affinity of of } AB_2 = W_{21} + W_{22}$$

$$\text{Weighted accuracy of } AB_2 = \frac{1.5 + W_{21}}{2 + W_{21} + W_{22}}$$

Figure 4.3: *The fitness calculation strategy of the proposed model, with k=2*

4.2.8 Antibody Selection

After the antibody fitnesses has been calculated, antibody selection is carried out, based on fitness values. Each generation, n antibodies are selected for reproduction. Tournament selection strategy (see Subsection 2.1.2) is used to find these n antibodies. Tournaments of a static size of 1/10 of the total antibody set are created, by randomly picking antibodies from the antibody set. Each tournament will produce a winner (by having the highest fitness score in that particular tournament), which is allowed to reproduce. If the winner of a tournament has already been allowed to reproduce that generation, the antibody with the second-highest fitness score is chosen as the winner instead. Implementations were made to ensure multiple versions of the same antibody don't compete in the same tournament.

The reasoning behind using tournament selection instead of simply selecting the n antibodies with the highest fitness scores (like VALIS does), is to introduce a greater degree of diversity into the antibody population. Through tournament selection, even antibodies with a relatively low fitness score may be allowed to reproduce, if its tournament contestants have even lower fitness scores. This results in more feature space exploration, which therein prevents premature convergence.

Further, each of the n selected antibodies produces m clones, for a total of $n * m$ new clones. The clones are then mutated and their fitnesses calculated. Then, the n best clones replace the n poorest performing (in terms of fitness) antibodies in the population, regardless of whether their fitness is better or worse than these antibodies. By replacing the worst antibodies with new clones regardless of their fitness scores, the diversity of the antibody population will decrease more slowly than if the fitness scores were considered. This is due to new mutations being introduced to the population at every generation. It is worth noting that this selection approach only gives the new antibodies one generation to "prove their worth" in the antibody population. If a new antibody is one of the n antibodies with the worst fitness score in the next generation, it's discarded after only one generation. This halts further local search surrounding the antibody parent, until a new antibody clone potentially mutates to similar feature space coordinates. This is a necessary sacrifice, in order to make the model run efficiently. More exhaustive local search would quickly increase both time and space complexity drastically, especially for high-dimensional feature spaces.

As the new clones replace current antibody individuals regardless of performing better or worse, some heuristic to aid the population (as a whole) to steadily improve was needed. If no such heuristic was used, the model could replace a significant portion of the antibody population all the way to the last generation – leading to large variation in accuracy results. To solve this problem, a *reproduction ratio* calculation was introduced – similar to that of VALIS. This *reproduction ratio* is based on the current generation number, compared to the total number of generations. The reproduction rate will gradually decrease, in relation to a distribution, as the model progresses through the generations. This means that the value for n (the number of antibodies to be replaced) progressively decreases. The *reproduction ratio* is calculated as:

$$AB_{rr}(g) = ratio_{max} \cdot \left(\frac{2}{n_{ab}}\right)^{\frac{3}{2} \cdot \frac{g}{g_{total}}}$$

where g is the current generation number, g_{total} is the total number of generations, $ratio_{max}$ is the maximum ratio of antibodies to be replaced and n_{ab} is the total number of antibodies. Further, the number of antibodies to be replaced is calculated as:

$$n = floor\left(AB_{rr} \cdot n_{ab}\right)$$

The proposed model's method of *reproduction ratio* differs from that of VALIS, as VALIS utilize a static value of 0.5 for $ratio_{max}$. Not using a static $ratio_{max}$ value allows for experimentation with different values of the maximum replacement ratio, as presented in Section 5.2.1. Additionally, VALIS raise $(\frac{2}{n_{ab}})$ to the power of $\frac{g}{g_{total}}$, instead of $\frac{3 \cdot g}{2 \cdot g_{total}}$. This change (in the proposed model) lead to the number of new antibodies declining more slowly across the generations. This change was implemented to introduce more diversity into the antibody population, which otherwise would be less diverse than that of VALIS – as the proposed model doesn't utilize crossover between antibodies.

As antibodies are continuously replaced with randomly mutated clones throughout the generations, some poor-performing antibodies might still be present in the antibody set at the last generation – leading to degraded accuracy. To solve this problem, the proposed model employs apoptosis of antibodies – inspired by AISLFS [Dudek, 2012] (as mentioned in Section 3.1.3). After the generation loops are complete, the fitnesses of the antibodies are calculated on the training set. The fitnesses are compared to the antibody removal threshold and antibodies scoring lower than this are removed from the antibody set. This fitness threshold is generally set quite low, as the training set isn't necessarily representative of the entire antigen population. The apoptosis process serves mainly as a tool to remove antibodies which are added late in the process, that don't connect to any antigens.

4.2.9 Mutation and Feature Subset Selection

As mentioned, the proposed model includes local feature selection amongst the antibodies, inspired by AISLFS [Dudek, 2012]. As opposed to AISLFS, the proposed model does not rely solely on mutating which features are used by each antibody (feature subset selection), but rather the features values, feature subset selection and RR radii are all mutated randomly, for each antibody. The inclusion of local feature selection combined with feature value and RR radius mutation is attributed to the large amount of features and difficulty of classification (the Liar and FNN datasets are inherently hard to perform classification on). Additionally, each antigen sample is less representative of an "average antigen" with that particular class label – as the feature values are extracted rather than provided "as-is", in a dataset. The authors of AISLFS only test their model on benchmark classification datasets, with given feature values and a relatively small amount of features.

Although being inspired by VALIS [Karpov et al., 2018], the proposed model doesn't employ crossover for reproduction. The reasoning behind this was that crossover is uncommon in AIS models generally and the literature study revealed no spam-classifying AISs which utilized crossover – only mutation. Additionally, as the features extraction was of an experimental nature, the impression was that crossover operations would introduce too much diversity into the population – which in turn would make it difficult to give a fair comparison of the efficiency of the various features (and their correlation). Further, changes were implemented in the model to introduce additional diversity into the antibody population – somewhat substituting the effect of crossover (see Section 4.2.8).

Instead of using crossover, each antibody selected for reproduction produces clones, which are subsequently run through a mutation process. First, the antibody *feature selection vector* is mutated. This is done by iterating through the vector and randomly flipping a boolean value with a probability of $\frac{1}{2*size(F)}$, where $size(F)$ denotes the amount of feature values in the feature vector. It should be noted that $size(F)$ is used rather than $size(F')$, i.e. the antibody's feature subset. Further, the strategy of VALIS where the antibodies are put through the mutation process repeatedly until at least one mutation occurs, is not used for the *feature selection vector*. This (and the low mutation probability) is due to the fact that including/excluding a given feature may have drastic impact on the antibody fitness, to a much larger extent than altering a feature value or RR radius slightly. As such, a mutation of the *feature selection vector* is not forced upon antibody clones.

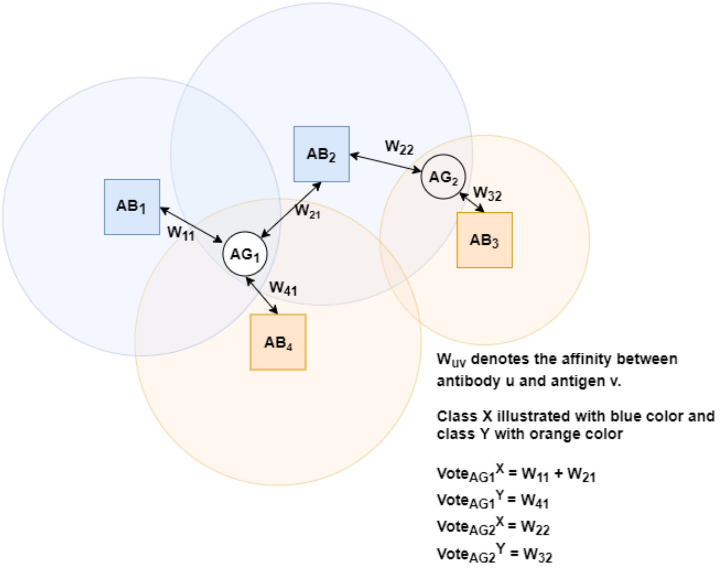
Next, the feature values and RR radius is mutated. Different mutation strategies were presented and discussed in Section 3.1.3. The mutation strategy used in the proposed model is inspired by that of MAIM [Baug et al., 2019], where the each value selected for mutation is multiplied by random number in the range 0.1-2.0, by some probability. The authors of MAIM tested several different coefficient ranges, with 0.1-2.0 scoring slightly better than the others. This mutation strategy was selected instead of VALIS' approach, which involves using a log-uniform density function, due to its simplicity. The RR radius is mutated the same way as the feature values. Additionally, the aforementioned strategy of continually repeating the mutation process until at least one mutation occurs is used for the feature values and RR radius (in union). This strategy was implemented to increase the diversity of the antibody population.

4.2.10 Class Prediction

After the training phase of the algorithm is finished, the model performs classification (predicting the class) of the previously unseen antigens in the testing set. The voting heuristic was decided based on the discussions in Section 3.1.8. AISLFS [Dudek, 2012] tested three different strategies – with two of them scoring roughly the same in terms of accuracy. Conversely, VALIS only tested one voting strategy, which included weighting an antibody's class vote according to its *Weighted Accuracy* in addition to its affinity to the antigen. The decision was made to use AISLFS' second class voting strategy, as the authors tested several alternatives and their selection therefore carry more weight than that of VALIS. Additionally, the strategy of VALIS was tested during preliminary testing – performing slightly worse than that of AISLFS. As such, each antigen in the testing set is classified by using a voting function for each class k in the dataset:

$$v_{ag,k} = \sum_{ab \in B_k} W_{ab,ag}$$

where k is the class, B_k is the set of antibodies with class k and $W_{ab,ag}$ is the affinity between antigen ag and antibody ab. In other words, all the antigens in the testing set are presented to the final antibody population and the antibody-antigen connections are calculated. Then, for every antigen, all the antibodies connecting to it votes on a class. The antibody class vote is simply the class of the antibody and the weighting of the vote is determined by the affinity between the antibody and the antigen. The class voting scheme is illustrated in Figure 4.4, where four antibodies vote on the classes of two antigens – based on affinities. $Vote_{AG1^X}$ refers to the voting tally for assigning class X to antigen $AG1$.



W_{uv} denotes the affinity between antibody u and antigen v.

Class X illustrated with blue color and class Y with orange color

$Vote_{AG1}^{X} = W_{11} + W_{21}$

$Vote_{AG1}^{Y} = W_{41}$

$Vote_{AG2}^{X} = W_{22}$

$Vote_{AG2}^{Y} = W_{32}$

Figure 4.4: *The class prediction (voting tally) strategy of the proposed model*

Finally, after all the class votes have been summed up for each class, the class with the highest cumulative vote tally is assigned to the antigen. If an antigen isn't connected to any antibodies, it is assigned the class of the antibody with the lowest $\frac{d(ab,ag)}{r_{ab}}$ ratio, i.e. the Euclidean distance to the antibody divided by the antibody RR radius.

4.2.11 Cross-Validation

The proposed model was run with *k-fold cross-validation*, as introduced in Subsection 2.1.6. To prevent knowledge leakage, all the manipulation of the dataset other than shuffling and splitting into k parts are carried out within the training loop. This can be seen in the pseudocode of the algorithm in Section 4.2.1. If, for instance, the normalization was moved outside the loop, the model would potentially get information about the testing set beforehand – which would result in overly optimistic results.

4.2.12 Feature Space Assessments

As introduced in Subsection 3.1.7, several machine learning techniques (including AISs) suffer from the "curse of dimensionality", meaning that the models tend to have a hard time when presented with high-dimensional data (large number of features) [Dudek, 2012][Baug et al., 2019]. In the proposed model, this dimensionality problem is reduced through multiple strategies. First, antibodies are initialized both randomly (like in MAIM [Baug et al., 2019]) and with antigen feature values (like in AISLFS [Dudek, 2012]). This scheme, paired with initialising the antibody RR radii large (set to include a randomly selected antigen of the same class as the antibody), serves as an initial "stepping stone" for helping the antibodies find antigens in the feature space.

Additionally, the local feature selection employed by each antibody helps reduce the amount of dimensions which are actually used by each antibody (when calculating affinity/Euclidean distance). Through inspection of the features used by each antibody during preliminary testing, the local feature selection scheme reduces the feature space dimensionality considerably. This feature reduction potential was also found in AISLFS, where the authors found that high accuracy results could be obtained with a fraction of the available features, for some datasets.

4.2.13 Pseudocode

The full pseudocode of the proposed model can be seen in Algorithm 1.

```
Data: a labeled dataset
Result: a set of antibodies capable of classifying previously data samples
parse dataset;
initialize antigens;
shuffle antigens;
split antigens into testing and training sets;
for training split in k do
    initialize antigens from training set k;
    extract features values;
    normalize feature values;
    initialize antibodies;
    for generation in G do
        calculate number of antibodies to be replaced ($n$);
        calculate connections between antibodies and antigens;
        calculate fitness of all antibodies;
        perform tournament selection to find $n$ antibodies which are
          allowed to reproduce;
        clone the $n$ antibodies $m$ times each;
        mutate the $n*m$ clones;
        remove the $n$ worst-performing antibodies from the antibody set;
        move the $n$ best-performing clones into the antibody set;
    end
    remove antibodies with fitness < antibody removal threshold;
    present the antibodies with the unseen testing set in k;
    calculate connections between antibodies and antigens;
    predict the class of all the antigens in the testing set;
    calculate accuracy;
end
calculate the average accuracy for all the testing sets;
```

Algorithm 1: Pseudocode of the proposed model

4.3 Pre-processing and Feature Extraction

This section introduces the pre-processing and feature extraction of the proposed model. As the input data to the model are news articles, strategies for extracting feature values which represent the class label of the articles were needed. The features of the proposed model are divided into four main categories. These categories are term frequencies, grammatical features, text embeddings and sentiment analysis. Overall, the features used include both features from other state of the art fake news classification models, as well as unique features introduced in this work. It should be noted that not all the proposed features were used in the final accuracy testing of the model. Rather, an assessment of the performance of the various features was carried out – in relation to the second research question from Section 1.2. This assessment was conducted in the feature testing phase (see Subsection 5.2.2) and the subsequent accuracy testing was based on the findings of this testing.

4.3.1 Pre-processing

For the word counting features, pre-processing was performed on the raw news article texts. The pre-processing flowchart can be seen in Figure 4.5, where text copies are colored green and processing operations are colored orange. Copies of the processed text were saved between each pre-processsing step, as different features required different degrees of pre-processing. Stanford CoreNLP [Manning et al., 2014] was used for tokenization and lemmatization, while removal of stop words and unwanted characters was carried out manually. Tokenization refers to the strategy of dividing the raw text into separate words, as explained in Section 2.1.8. Whitespaces were used as delimiters. Further, unwanted characters such as ”.”, ”,”, ”(”, ”)” and ” ´” was removed from each token, using a list of unwanted characters. Thereafter, the tokens (words) were lemmatized using Stanford CoreNLP. Lemmatization (sometimes referred to as ”word stemming”, although word stemming often denote a more simplistic and crude approach were word endings are simply cut off) refers to reducing words to their base (dictionary) form. This process maps different words carrying the same meaning to the

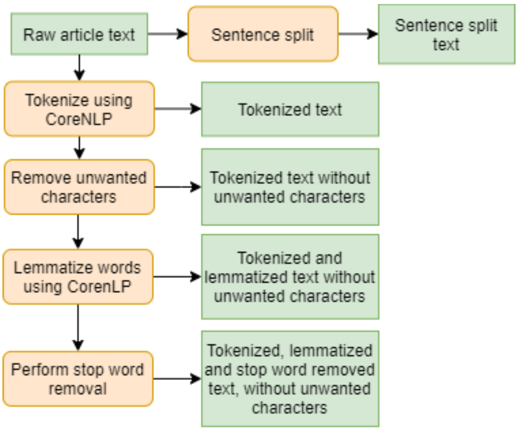

Raw article text
Sentence split
Sentence split text
Tokenize using CoreNLP
Tokenized text
Remove unwanted characters
Tokenized text without unwanted characters
Lemmatize words using CorenLP
Tokenized and lemmatized text without unwanted characters
Perform stop word removal
Tokenized, lemmatized and stop word removed text, without unwanted characters

Figure 4.5: *The pre-processing of news article text corpus*

same word – making subsequent calculations easier and more reliable. For instance the lemmatization of "am", "are" and "is" is "be", and "walked", "walks" and "walking" are all lemmatized to "walk". It should be noted that lemmatization was only used when certain term frequency features were used, as the computational load of lemmatization was considerable.

Lastly, stop words were removed from the tokenized text, using a lexicon. As introduced in Section 2.1.8, stop word removal refers to the removal of words carrying little symbolic meaning. Examples of such words are "by", "before", "it" and "the". As these words may obscure feature extraction, they are removed from the text samples. The stop word lexicon used was manually checked to ensure that no words that might be specifically useful for fake news classification (in relation to the research conducted, see Section 3.2) were removed.

4.3.2 Term Frequency Features

Most of the features extracted in the proposed model are based on calculating the frequencies of various terms (words). These terms are selected based on the findings from the literature study in Section 3.2.1. The features are computed on the tokenized headline and text corpus of the article samples. Lexicons containing the relevant terms are used as look-up tables, to identify the use of certain words in the article samples. The term frequency (TF) is then calculated by dividing the count of matches found (according to the lexicon in question) by the total number of words in the headline and article text:

$$TF_{term} = \frac{count(term)}{count(total\ number\ of\ words)}$$

Further, the term count is computed as:

$$count(term) = 2 * count_h(term) + count_a(term)$$

where $count_h(term)$ is the number of term matches in the article headline and $count_a(term)$ is the number of term matches in the full article text. It should be noted that terms appearing in the headline are weighted double. The reasoning behind this is the findings of 3HAN [Singhania et al., 2017] (as presented in Section 3.2.2), where the authors found that the headlines of news articles generally were more representative of the class of the article, than the remaining article text. As such, it was decided to assign the headline twice the weight of the subsequent article text – for the term frequency features.

Additionally, TF was chosen over Term Frequency-Inverse Document Frequency (TF-IDF) (as introduced in Subsection 2.1.8). The reason for this choice was twofold. Firstly, computing TF-IDF instead of TF would increase the computing time of feature extraction, as two iterations through the dataset samples would be needed (instead of just one). Additionally, TF-IDF is mostly used on single terms, while the proposed model uses lexicons of multiple terms for each feature. As most articles generally include at least one of these words, the extra computation of the TF-IDF would most likely not change the calculated term frequency values notably. Secondly, some information regarding the occurrences of certain terms across all the samples are already provided to each sample, during the Min-Max normalization. As the feature values are normalized across all the samples, this information is already used for computing the final feature values.

Half of the term frequency features used in the proposed model were inspired by the model of Rashkin et al. [Rashkin et al., 2017]. The reason for this was that the authors provided the lexicons used for their TF computations, as well as providing information regarding the effectiveness of each feature. This information was considered as highly valuable guidance for TF-based features. For these features (1-8), the lexicons used was those of Rashkin et al., unless otherwise stated. An overview of the features used in the proposed model, together with lexicon length and reference(s) can be seen in Table 4.2.

Feature	**Description**	**Lexicon length**	**Reference**
1	Swear words	1383	[Rashkin et al., 2017]
2	First person pronouns	5	[Rashkin et al., 2017]
3	Second person pronouns	5	[Rashkin et al., 2017]
4	Modal adverbs	94	[Rashkin et al., 2017]
5	Action adverbs	15	[Rashkin et al., 2017]
6	Manner adverbs	128	[Rashkin et al., 2017]
7	Superlative forms	2304	[Rashkin et al., 2017]
8	Comparative forms	2118	[Rashkin et al., 2017]
9	Strongly subjective words	8222	[Horne and Adali, 2017]
10	Numbers	n/a	[Meel and Vishwakarma, 2019]
11	Negations	24	[Gravanis et al., 2019]
12	Negative opinion words	4783	[Hu and Liu, 2004][Gravanis et al., 2019]
13	Positive opinion words	2006	[Hu and Liu, 2004][Gravanis et al., 2019]
14	Exclamation and question marks	8	[Meel and Vishwakarma, 2019]
15	Quotation marks	4	n/a
16	Divisive topics	43	[Meel and Vishwakarma, 2019][Gravanis et al., 2019]
17	Word count	n/a	[Gravanis et al., 2019]
18	Flesch-Kincaid Grade Level	n/a	[Gravanis et al., 2019][Burgoon et al., 2003]
19	Flesch-Kincaid Reading Ease	n/a	[Choudhary and Arora, 2021]
20	Headline text embedding	n/a	[Nakamura et al., 2020]
21	Head + tail text embedding	n/a	[Nakamura et al., 2020]
22	Head text embedding	n/a	[Nakamura et al., 2020]
23	Tail text embedding	n/a	[Nakamura et al., 2020]
24	Headline sentiment analysis	n/a	[Bhutani et al., 2019]
25	Head + tail sentiment analysis	n/a	[Bhutani et al., 2019]

Table 4.2: The features of the proposed model

Feature 1: Swear Words

Firstly, the term frequencies of swear words were calculated. Although seeming like a relatively naive feature, Raskin et al. found that the frequency of swear words was the most effective feature for separating fake and real news articles (with swear words appearing 7 times as often in fake news articles as in real ones). The lexicon used for swear words was the union of the list from Rashkin et al. (compiled from Wiktionary) and the "Offensive/Profane Word List" from Luis von Ahn's Research Group. The reason for using an union of both lists, was that the list of Rashkin et al. was relatively short by itself.

Feature 2: First Person Pronouns

Rashkin et al. also found that first person singular pronouns appeared twice as often in fake news articles as in real ones. A short lexicon containing the terms: "i", "me", "my", "mine" and "myself", was used to calculate the term frequency of first person pronouns.

Feature 3: 2nd Person Pronouns

Rashkin et al. also found that second person pronouns was the second-most distinctive feature of fake news articles, appearing almost 7 times more often in fake news articles as real news articles. A short lexicon containing the terms: "you", "your", "yours", "yourself" and "yourselves", was used to calculate the term frequency of second person pronouns. Similarly to the first person pronouns, the small size of the lexicon results in this feature having low computational demands.

Feature 4: Modal Adverbs

The third most distinctive feature found by Rashkin et al. was the usage of modal adverbs, meaning words that add additional value to verbs. Examples of adverbs from the lexicon are "needlessly" and "noticeably".

Feature 5: Action Adverbs

A term frequency feature based on the action adverbs was also implemented. Action adverbs are mostly similar to modal adverbs, but could be described as more extreme in their symbolism. Rashkin et al. found that action adverbs generally appear twice as often in fake news articles as in real ones. Examples of action adverbs from the lexicon are "freely" and "immaturely". It should be noted that the lexicon used for computing action adverb TFs was relatively short, only containing about 15 terms – making the computational demands of this feature low.

Feature 6: Manner Adverbs

Manner adverbs describe how someone does an action, for instance ”slowly” or ”foolishly”. Rashkin et al. theorize that fake news authors try to enliven news stories to attract readers, by including such words in their articles. Further, the authors found that manner adverbs appear almost twice as often in fake news articles as in real ones.

Feature 7: Superlative Forms

Superlative forms are adjectives which are used to compare on thing with another, taken to the maximum extent. This means words describing being the best or worst at some adjective, like ”worst”, ”finest” and ”slowest”. Rashkin et al. found that fake news generally tend to have higher frequencies of superlatives. They ground this in fake news authors wanting to imply a greater degree of dramatization than real news authors.

Feature 8: Comparative Forms

The comparative form lexicon used contain similar words as the superlative form lexicon, but the adjectives listed don't identify that the subject is the best or worst at something – but rather *more* of something. Examples of words from the lexicon used for identifying comparative forms are ”sooner”, ”greater” and ”worse”.

Feature 9: Strongly Subjective Words

It was hypothesized that fake news articles use a greater amount of strongly subjective words than real news articles – as the authors of fake news articles would like to elicit an emotional response from their readers [Horne and Adali, 2017]. Examples of such words are ”abhorrent”, ”undoubtedly” and ”amazing”. The lexicon used for computing the term frequency was the MPQA subjectivity lexicon. It should be noted that only the words denoted as ”strongly subjective” in the subjectivity lexicon were used.

Feature 10: Numbers

Rashkin et al. also found that the inclusion of numbers was more than half as likely in fake news articles as in real ones [Rashkin et al., 2017]. This characteristic was also found in [Meel and Vishwakarma, 2019], where the authors ground this on fake news articles lacking concrete numerical data on which to base the article. As such, term frequency of numbers were calculated in the news article samples. A regular expression was used to identify numbers in the article texts.

Feature 11: Negations

Raskin et al. also found that negations appeared more often in fake news articles than real ones [Rashkin et al., 2017]. The authors used LIWC [Pennebaker et al., 1999] to identify the negation terms. Additionally, in their fake news benchmark study, Gravanis et al. also use term frequency of negations in their best-performing feature set [Gravanis et al., 2019]. To identify the use of negations in the article samples, a lexicon was manually constructed, containing words such as "not", "no", "none" and "never".

Feature 12: Negative Opinion Words

Gravanis et al. also used a "negative emotions" feature in their best-performing feature set [Gravanis et al., 2019]. Considering the aforementioned aim of fake news articles to elicit emotional responses from readers, such a feature is inherently intuitive. As Gravanis' lexicon isn't publicly available, the lexicon used to identify such negative opinion terms was taken from a customer review mining model [Hu and Liu, 2004] and includes words such as "absurd", "2-faced" and "disbelieve".

Feature 13: Positive Opinion Words

A feature identifying positive emotion terms was also used in the best-performing feature set of Gravanis et al. The lexicon used by Gravanis et al. is not publicly available. However, Hu and Liu also include a list of words being associated with positive opinions, in their model. This list was used to calculate the term frequency of positive opinion words in the news article samples of the proposed model. The list includes words such as "admire", "passionate" and "relish".

Feature 14: Exclamation and Question Marks

Inspired by the findings of Meel and Vishwakarma in their fake news survey paper [Meel and Vishwakarma, 2019], a feature computing the frequency of exclamation and question marks was implemented. Meel and Vishwakarma found that fake news generally include a larger number of exclamation and question marks and ground this on the fake news authors' wish to draw the reader's attention. To identify the use of such characters used in combination as well as by themselves, a regular expression was constructed. This regular expression match combinations like "?!", "??" and "!!", in addition to "!" and "?".

Feature 15: Quotation Marks

Upon manual inspection of the relevant dataset, it was theorized that fake news articles may also contain a higher frequency of quotation marks than real news articles. As such, a feature which computes the frequency of quotation marks (in various forms) was implemented.

Feature 16: Divisive Topics

Based on the discussions in Gravanis et al.'s benchmark study [Gravanis et al., 2019], as well as the discussions of Meel and Vishwakarma, it was hypothesized that fake news articles may generally have a different topic selection than real news articles. Continuing off of the previously mentioned theory of fake news authors wanting to elicit emotional response in their readers, the fake news articles may revolve around divisive topics – to a greater extent than real news articles. Divisive topics refer to topics which tend to cause disagreement between people. Based on this theory, a lexicon was manually constructed, containing terms such as ”vaccine”, ”gun control”, ”gay” and ”climate change”. This lexicon was subsequently used to compute the term frequency of divisive topics in the news articles.

4.3.3 Grammatical Features

Features focusing more on the grammatical structures of the news articles were also implemented. These features use statistical characteristics of the various news articles to compute feature values.

Feature 17: Word Count

Word count was also used as a feature in the best-performing feature set in Gravanis et al.'s benchmark study [Gravanis et al., 2019]. It was theorized that fake news articles might generally be shorter in length than real news articles, due to having less real-world events to base the article on. Although initially seeming overly simplistic, the feature was implemented due to the low computational cost. As the article texts are tokenized anyway, the extra computational effort of calculating word count is negligible. The implemented feature simply calculates the total amount of words in the headline and article text of each dataset sample.

Feature 18: Flesch-Kincaid Grade Level

Flesch-Kincaid Grade Level refers to a formula which calculates the equivalent US grade level of education required to be able to understand a given text. The formula outputs a number between 0-20, indicating the equivalent grade level. For instance, a grade level of 8 is roughly equivalent to the grammatical complexity of the Harry Potter books, while a grade level of 16 maps to the complexity of an average academic paper. The grade level is calculated as:

$$FK\ Grade\ Level = 0.39 * \frac{total\ words}{total\ sentences} + 11.8 * \frac{total\ syllables}{total\ words} - 15.59$$

Gravanis et al. used Flesch-Kincaid Grade Level in their benchmark study, inspired by the linguistic features proposed by Burgoon et al. [Burgoon et al., 2003]. Based on these results, it was decided to include Flesch-Kincaid Grade Level as a feature in the proposed model. The reasoning behind this feature is that the authors of fake news articles may not have as much of a formal journalistic education as real news article authors and thus the grammatical complexity may be lower in fake news articles than in their real counterparts. Additionally, the targeted audience of fake news articles may be different than for real news articles and as such the writing style may be simplified.

This feature calculation places an additional computational load on the model, as the number of sentences and syllables had to be calculated for each dataset sample. Sentence count was found by using Stanford CoreNLP's tokenization tools and syllable count was calculated using a manually constructed algorithm.

Feature 19: Flesch-Kincaid Reading Ease

Flesch-Kincaid Reading Ease works similarly to Flesch-Kincaid Grade Level in that the formula gives a score based on how difficult an English text is to read. Although using many of the same core components, the different weighting factors results in a different measure of the grammatical complexity. Flesch-Kincaid Reading Ease is calculated as:

$$FK\ Reading\ Ease = 206.835 - 1.015 * \frac{total\ words}{total\ sentences} - 84.6 * \frac{total\ syllables}{total\ words}$$

The result will be a score between 0-100, where a low score means that the text has low readability and vice-versa. In their linguistic-based model for fake news classification, Choudhary and Arora utilize Flesch-Kincaid Reading Ease as a linguistic feature – obtaining promising results [Choudhary and Arora, 2021]. As such, the Flesch-Kincaid Reading Ease is used as a feature in the proposed model, to provide an indication of the relative readability of the news articles. It is hypothesized that these two Flesch-Kincaid features may somewhat replace the LIWC-based grammatical features used by Rashkin et al. in their model [Rashkin et al., 2017]. As LIWC is a paid product, it was not used for extracting features in the proposed model. Further, as the parameters for calculating the reading ease are the same as for calculating the grade level, the additional computational complexity was also low.

4.3.4 Sentence Embeddings

Sentence embeddings were used as feature values, inspired by the promising results of Fakeddit [Nakamura et al., 2020] and FakeBERT [Kaliyar et al., 2021]. Google's BERT model [Devlin et al., 2019] was used to extract fixed-length sentence embeddings, through the use of the bert-as-service tool [Xiao, 2018]. Other text embedding models were considered as well, but were ultimately discarded. The reasoning behind this was twofold. Firstly, the authors of FakeBERT tested text embeddings with GloVe and word2vec in addition to BERT, but the results obtained were worse than those obtained with BERT. This is most likely due to the aforementioned bi-directional processing of BERT, where the input text is processed both from left-to-right and right-to-left. This allows for the extraction of contextual information, where words are weighted based on the context in which they appear in – leading to more representative output vectors. Additionally, BERT is able to take entire sentences as input – not just single words (like GloVe, ELMo and word2vec). If single word embeddings were to be used, it would raise the problem of word selection, for each article. Simply taking the most frequent word(s) from each article as input to the word embedding model would place a large dependency on the stop word removal, as the most frequent words of the raw article texts would likely be words like "a" and "the". The words carrying the most symbolic "impact" in each article would most likely be difficult to find without more comprehensive processing. Additionally, if several word embeddings were to be extracted from each news article, the feature space would quickly grow too large – as each word embedding typically would consist of 50-1024 feature values.

A pre-trained BERT model was used for extracting the sentence embeddings, more precisely the "BERT-Base, Uncased"-model. Uncased means that the text has been lowercased before tokenization, when the model was trained (on Wikipedia text corpus). The reasoning behind selecting the Uncased version was that the authors of BERT generally recommend this over the Cased version, unless one is aware that case information is important to the specific application. Additionally, the authors of Fakeddit [Nakamura et al., 2020] also used the Uncased version of the model. Further, the Base-edition of the model was chosen over the Large-edition. Due to limited processing power, the extra computational effort of the Large-model was deemed unnecessary. Additionally, the Large-model would result in a fixed output vector length of 1024 instead of 768, which would worsen the aforementioned dimensionality problem.

The aforementioned problem of excessively large feature spaces was still a relevant problem when using BERT for sentence embeddings. The resulting output vectors were of a fixed length of 768 floating values, regardless of the length of the input sentence. This is not an issue for Neural Network-based models like Fakeddit, but AIS models are generally less robust to large feature spaces due to the previously mentioned "curse of dimensionality" [Dudek, 2012][Baug et al., 2019]. Excessively large feature spaces may result in the feature space being too large for the antibodies to successfully locate the antigens. In the proposed model, this problem was somewhat mitigated through the strategies highlighted in Subsection 4.2.12, but nonetheless the choice was made to use a maximum of one of the sentence embedding features for the final accuracy testing – to keep the feature space as small as possible.

Feature 20: Headline Text Embedding

The first sentence embedding feature uses the headline of the news articles as input to BERT. As most of the news article samples used for training and testing had single-sentence headlines, it was theorized that using the headlines for text embedding extraction was a representative way to capture the symbolic meaning of each news article. Additionally, the authors of Fakeddit [Nakamura et al., 2020] used Reddit submission titles for their text embeddings, obtaining promising results. As these Reddit submission titles are similar to the news article headlines in terms of writing style and grammatical structure, it was theorized that high accuracies could be obtained using the headlines for text embedding.

Feature 21: Head + Tail Text Embedding

Additionally, to extract contextual information not present in the article headlines, a feature based on text embeddings extracted from the article texts was also implemented. As generating sentence embeddings for the entire article text corpus was far too computationally extensive and would result in too large feature spaces (768 feature values for each sentence/512 tokens), it was decided to only use the first and last sentence of each news article for extraction. This approach is often referred to as *head + tail*. Sun et al. found that the *head + tail* approach resulted in promising result for classifying news articles [Sun et al., 2019]. It should be noted that Sun et al. used *Chinese* news articles and performed classification in terms of theme category ("sports", "house", "business" etc.) and not fake/real. Nevertheless, these results support the theory that the sentiment of news articles could potentially be represented by only using the first and last sentence of the text corpus – reducing the computation time significantly.

The first and last sentences of each news article were concatenated and the resulting string sent as input to BERT. This concatenation was done in order to extract a single output vector of 768 feature values. If the two sentences were sent separately, the result would be two output vectors of 768 values each – worsening the aforementioned dimensionality problem. As the maximal text input length of BERT is 512, the concatenated input string was cut short (from the end) if it somehow exceeded 512 tokens.

Feature 22: Head Text Embedding

As previously mentioned, the *head + tail* approach to BERT text embeddings performed the best in the paper by Sun et al. However, the accuracy results obtained were only slightly better than using the first and last sentence separately [Sun et al., 2019]. Due to the lower processing time of only using the first (or last) sentence for text embedding extraction, it was decided to also test the *head* approach. This involves simply using the first sentence of the news article corpus as input to BERT.

Feature 23: Tail Text Embedding

The *tail* approach to text embedding was also implemented. This approach involves simply using the last sentence of the news article corpus as input to BERT. Similar to the *head* approach, this feature also ensures that the entire last sentence is passed to BERT (unless its length exceeds 512 tokens), unlike the *head + tail* approach (where the last sentence may be cut short).

4.3.5 Sentiment Analysis

As shown by Meel and Vishwakarma in their benchmark study for fake news classification, several researchers have found success in using sentiment analysis to aid fake news classification [Meel and Vishwakarma, 2019]. Bhutani et al. also found that including sentiment analysis as a feature in their model improved fake news classification accuracy [Bhutani et al., 2019]. Their model was also tested on the (notoriously difficult) Liar dataset, which provides additional legitimacy to their findings. As such, it was decided to use sentiment analysis to predict a sentiment class, which was then used as a feature value in the proposed model.

The sentiment analysis tool of Stanford CoreNLP was used to extract the news article sentiments. As CoreNLP was already used for tokenization and pre-processing, it was the natural choice for the task of sentiment analysis. Furthermore, the sentiment analysis tool of CoreNLP is built on top of a sentiment treebank (parsed text corpus annotated with semantic word classes, like ”verb” or ”noun”), which allows it to better capture the meaning of longer phrases, as well as the effects of negations, in a more precise way than traditional approaches [Socher et al., 2013]. The tool takes an unprocessed sentence/phrase as input and outputs a score between 0-4 based on the extracted opinion. The opinions range from very negative (0) to very positive (4). These sentiment-representative integers are further normalized and used directly as feature values for classification. Additionally, the *Shift-Reduce Constituency Parser* of CoreNLP was used to speed up the analysis significantly – making the feature feasible for larger datasets.

Feature 24: Headline Sentiment Analysis

Similar to the text embeddings, sentiment extraction is computationally heavy and as such it would be infeasible to use the entire news article corpus as input. Therefore, it was decided to split the sentiment analysis into *headline* and *head + tail* features, similar to the text embeddings. For the first of the two sentiment analysis features, the headline of each news article was used as input to the sentiment analysis. The resulting sentiment class predicted by CoreNLP was used as a feature value.

Feature 25: Head + Tail Sentiment Analysis

A *head + tail* approach was used to predict the sentiment of the news article corpora. The first and last sentence of each news article was concatenated and input to the sentiment extraction tool. As CoreNLP doesn't have a strict input length limit like BERT, the concatenated string was not cut short.

Chapter 5

Experiments and Results

This chapter aims to explain the experimental setup for testing of the proposed model, as well as discussing the obtained results.

5.1 Visualization Tools

In order to better understand the functioning of the proposed model, two visualization tools were implemented. These were used as part of the model refinement testing, to tweak and reason about model parameters – as well as in the main testing, to better understand the model's strengths and weaknesses.

5.1.1 2D Solution Visualization

The first visualization tool allows for a 2-dimensional visualization of the solution provided by the model. An example of a solution plot can be seen in in Figure 5.1, where the model has been run on the Spirals dataset. In the plot, antigens are illustrated as squares, with their color indicating their class label. Antibodies are illustrated as partly transparent circles, with radii corresponding to their RR radius. The antibody classes are illustrated with the same colors as the antigens. The antigens are plotted on top of the antibody circles, so that their color isn't altered even if the antigens are covered by several antibody RRs. Additionally, the values of relevant model parameters are provided at the side of the plot.

An example of a solution plot can be seen in Figure 5.1. This visualization tool allowed for debugging and an enhanced understanding of how the antibodies evolve. Generally, the antibodies form the expected spiral pattern, with the larger RR radii appearing toward the outskirts – where the distances between antigens

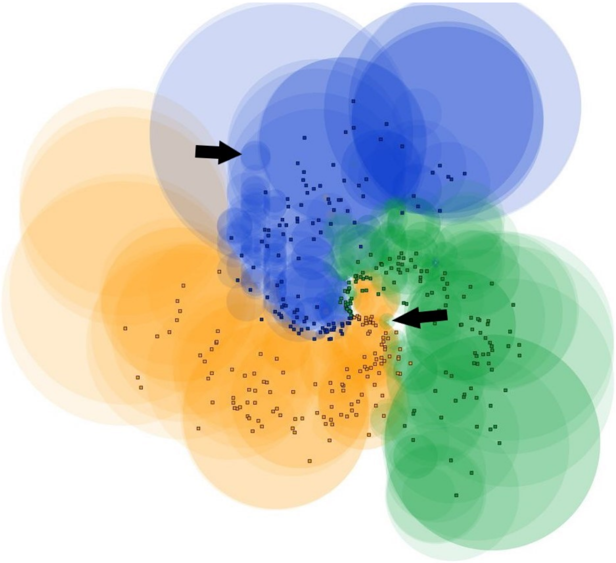


Figure 5.1: *Example solution plot for the Spirals dataset*

are large. Towards the middle, where the distances between different-class antigens are much smaller, the antibody RR radii are correspondingly smaller. Upon close inspection, antibody circles not containing any antigens can be found (two of these are identified with black arrows) This is to be expected, as the antibodies are evolved using the training set of antigens. The antigens in the plot are from the testing set, which the antibodies have never been exposed to. Although these antibodies have a fitness of zero (due to not connecting to any antigens) for the testing set, this was not the case for the training set, or the antibody would not have survived to the final generation. This is closely tied to the issue of *overfitting*, where the produced solution is too tailored to the training set – sacrificing general applicability. This issue doesn't, however, seem to be detrimental for the produced solutions – as most antibodies cover at least one same-class antigen. Additionally, having some "unused" antibodies may boost the robustness of the model, as these antibodies mark given areas in the feature space as most likely belonging to some class label – although not necessarily being used for all data samples.

It is important to note that the main utility of the plotting tool is for 2-dimensional datasets, i.e. datasets with two features. For datasets with more than 2 features, one could still plot a select two of the features – but these would not be expected to be as easily divided into areas of same-class antigens as the Spirals dataset. However, the plotting provides a useful tool for testing and comparing extracted features, as well as reasoning about how different parameters change the solutions produced.

5.1.2 Accuracy Plotting

Additionally, a tool for visualising the accuracy over time was implemented. This tool calculates the accuracy of the current antibody set (at some generation), on either the testing or the training set. The accuracy at each generation is the *average accuracy at that generation*, across all k cross validation splits. To prevent knowledge leakage, a copy of the current antibody set is used to calculate the testing set accuracy. This way, the antibodies stay ”uncontaminated” with testing set antigens. The accuracy plot provides insight into convergence/divergence of the model, as well as giving a measure of how well the initialization scheme (see Section 5.4.1) works.

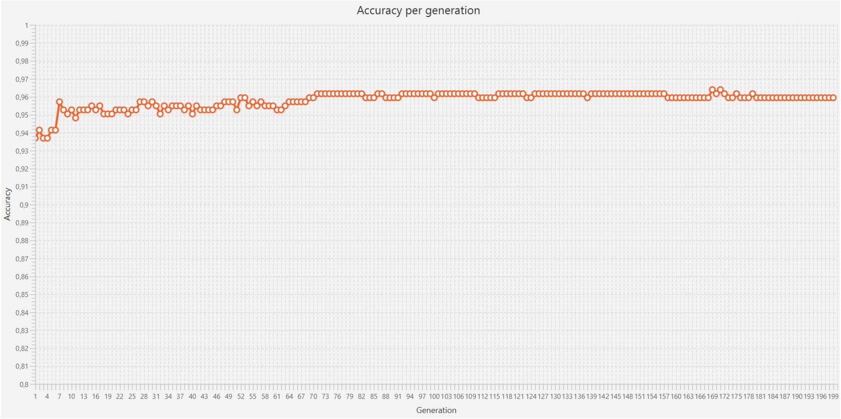


Figure 5.2: *Example accuracy plot for the training sets, using the Iris dataset*

In Figure 5.2, the accuracy plot for the *training sets* can be seen (using the Iris dataset). As expected, the model converges towards a very high accuracy. This is expected, as the antibodies are evolved using these data samples (antigens) for training. Additionally, the plot illustrates the effectiveness of the initialization scheme – combining two antibody initialization schemes and starting off with large RR radii. Even when the amount of antibodies are significantly less than the total amount of antigens in the training set (the model was run with an AB_{ratio} of 0.7 in this example), the model still has a high accuracy initially – before any antibody cloning and mutation has been performed. The convergence happens relatively quickly, which traditionally has been an issue for AIS models. However, the random initialization scheme contributed to a slower convergence, indicating that more local search was performed by the antibodies. Plots and discussions related to the initialization scheme are presented in Subsection 5.2.1.

In Figure 5.3, the accuracy plot for the *testing sets* can be seen (also using the Iris dataset). It is important to note that the accuracies plotted are the accuracies obtained when performing classification on the current testing set, which the antibodies have never seen before. As such, the accuracies are both lower and more volatile (compared to the accuracies obtained from the training set) – due to not evolving specifically towards classifying these antigens. However, there is a clear trend to be identified, where the accuracy is increasing (and converging) over time.

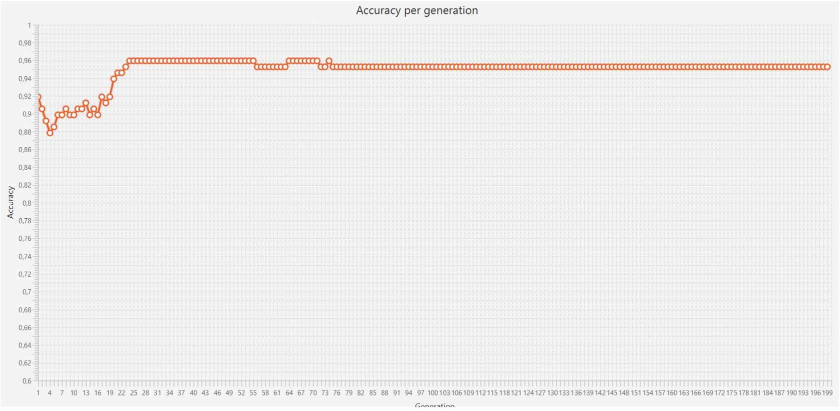


Figure 5.3: *Example accuracy plot for the testing sets, using the Iris dataset*

5.2 Experimental Plan

The primary goal of the experimental plan was to answer the research questions put forth in Section 1.2. As such, the testing plan was split into three distinct phases. The first testing phase was related to the first research question and sought to answer what design choices may be suited for the underlying AIS. As the actual testing on fake news datasets aren't conducted until phase 2 and 3 of the testing, the first phase used traditional classification benchmark datasets for evaluating design decisions.

The second phase was concerned with answering the second research question, i.e. identify which feature extraction strategies are suitable for an AIS model adapted for classifying news articles. As a wide variety of features have been proposed, with varying degrees of computational complexity, it was important to pick out which features were the most beneficial to the proposed model.

The information obtained from the second testing phase was used to guide the accuracy testing in phase 3. The third phase sought to answer the third research question, i.e. how the proposed model performs (in term of classification accuracy) compared to other state of the art fake news classification models. As the main goal of the thesis was to investigate the application of an AIS model to the task of fake news classification, this performance comparison served as an important evaluation criteria.

5.2.1 Model Refinement Testing

As mentioned, the proposed model is a product of a continuous and iterative process of implementing/altering parts of the model and evaluating changes on the produced results. The aim of the model refinement testing was to identify which design decisions should be made regarding the underlying AIS – so that it wouldn't be a performance bottleneck for the final fake news classification. The full model refinement testing plan is given in Table 5.1.

RR Radius Initialization (MT-1)

In Section 4.2.5, two different RR radius initialization approaches were presented; maximizing RR radius without including any different-class antigens in the RR (AISLFS [Dudek, 2012]) and setting the RR radius to the distance to a random antigen of the same class as the antibody (VALIS [Karpov et al., 2018]). Testing was conducted to evaluate the impact of these RR radius initialization schemes on the results. The solution and accuracy plotting tools were used to inspect the solutions produced by the model.

Antibody Replacement Ratio Testing (MT-2)

As mentioned in Section 4.2.8, the proposed model uses a dynamic antibody replacement ratio, AB_{rr}. This allows for experimentation with how many anti-

ID	**Test Description**	**Goal**
MT-1	Compare the AISLFS and VALIS RR radius initialization schemes by utilising the solution plotting tool	Gain an understanding of the impact of selected RR radius initialization scheme on the produced solutions
MT-2	Run accuracy tests on the underlying AIS, to assess the impact of the antibody replacement parameter	Gain an understanding of the impact of different antibody replacement ratios
MT-3	Run accuracy tests on the underlying AIS, on the **Iris**, **Wine** and **Diabetes** datasets	Evaluate and compare the obtained results with those obtained by state of the art AIS models

Table 5.1: The testing plan for model refinement testing

bodies should be replaced each generation. Testing was conducted to evaluate the effects of running the AIS model with different AB_{rr} values.

Preliminary Accuracy Testing (MT-3)

The aim of the preliminary accuracy testing was to test the performance of the base of the proposed model. The results of this testing provided an evaluation of how well the AIS performs on regular (not fake news) benchmark classification datasets. If only testing on fake news classification datasets was conducted, it would be difficult to tell if the main bottleneck (in terms of accuracy) was the feature extraction or the underlying AIS. As such, the aim of the preliminary testing was not to produce results competitive with state of the art models – by comprehensive parameter tuning towards each dataset. Rather, the aim was to test if the underlying AIS performs reasonably well on various benchmark datasets – such that the subsequent testing may give a precise estimate of the performance of the feature extraction strategies adapted.

Hypothesis of MT-1

It was theorized that the VALIS approach would generally produce significantly larger RR radii than the AISLFS, at least initially. This hypothesis was based on the fact that the AISLFS initialization strategy prevents antibodies from including any different-class antigens in their RR (contrarily to the VALIS approach). Further, the AISLFS approach would likely have higher accuracies initially, as antibodies don't include any different-class antigens in their RR. However, it was hypothesized that after some generations, the VALIS approach would catch up to the AISLFS approach – in terms of accuracy. As the VALIS approach allows for much larger RR radii, the model is given more room to explore – therein achieving more accurate results.

Hypothesis of MT-2

It was hypothesized that the antibody replacement ratio would have a large impact on the classification accuracy throughout the generations. When large parts of the antibody population are replaced every generation, the accuracy (on both the testing and training sets) were expected to oscillate considerably. This is due to the chosen approach of replacing the poorest antibodies with new clones, regardless of whether the fitnesses of the clones exceed the fitnesses of the antibodies. Further, too low values for the replacement ratio was expected to result in accuracies which don't improve throughout the generations – due to too little exploration being conducted by the antibody population.

Hypothesis of MT-3

The model was expected to produce accuracies slightly lower than those obtained by state of the art AIS models, on benchmark (non-fake news) datasets. As these models are developed specifically with these benchmark datasets in mind, it would be unrealistic to expect the proposed model to perform on par with these models. Additionally, these models generally perform specific parameter tuning towards each dataset. Comparatively, the preliminary accuracy testing of the proposed model was conducted with the same parameters for all the datasets. This lead to the assumption that the proposed model would generally achieve accuracies of 1-5% lower than than those obtained by the state of the art AIS models, across the datasets.

5.2.2 Feature Testing

For the feature testing, the usefulness of the various features was found by performing feature scoring, using the Java-ML library [Abeel et al., 2009]. The feature scores were found by computing the information gain (mutual information) ratio between each feature and the class labels, therein evaluating the dependency between features and class labels. The information gain is calculated as:

$$IG(f_i, C) = H(f_i) - H(f_i|C)$$

Where $H(f_i)$ is the entropy of feature f_i and $H(f_i|C)$ is the entropy of feature f_i, when considering (observing) class C_k. The entropies are calculated as:

$$H(f_i) = -\sum_{j \in X} p(x_j) log_2(p(x_j))$$

$$H(f_i|C) = -\sum_{k \in C} p(C_k) \sum_{j \in X} p(x_j|C_k) log_2(p(x_j|C_k))$$

Where X is the data samples set and $p(x_j)$ and $p(C_k)$ are the probability distributions of x_j and C_k, respectively. $p(x_j|C_k)$ is the joint probability distribution of x_j, given C_k. This results in a score for each feature, based on how useful the feature is for classification (higher scores being better). The feature scores were calculated after normalization of the feature values. By calculating these feature scores on all three datasets, an accurate estimate of the various features' usefulness for classification was obtained. The information found through this testing scheme was then used for the subsequent accuracy testing.

As the feature scoring strategy could not be applied to the text embedding features (as these have multiple feature values each), another feature evaluation scheme was used. This scheme includes running the algorithm (with specifications as presented in Section 5.3) using one text embedding feature at a time (without any other features). The classification accuracies were obtained and used to rank the text embedding features, internally. These classification accuracies were calculated for all the datasets. The obtained feature rankings were then used to reason about which text embedding feature to use in the subsequent accuracy testing phase.

Additionally, mutual information between the features was calculated. The JavaMI toolbox [Pocock, 2012] was used to calculate the mutual information of each feature pair. By calculating the mutual information between all the features, on all three datasets, general dependence between the features was identified – which aided the decision of which features should be used in the subsequent accuracy testing phase. The mutual information is calculated as:

$$I(X,Y) = \sum_{y \in Y} \sum_{x \in X} p_{(X,Y)}(x,y) * log_2 \left(\frac{p_{(X,Y)}(x,y)}{p_X(x) * p_Y(y)} \right)$$

Where X and Y are the feature value vectors, $p_{(X,Y)}$ is the joint probability distribution and p_X and p_Y are the marginal probability distributions. The probability distributions were estimated using histograms (by JavaMI) and the resulting mutual information is a number in the range:

$$[0, min(H(X), H(Y))]$$

Where $H(X)$ and $H(X)$ are the entropies of X and Y, calculated as previously introduced. Additionally, it should be noted that the mutual information was calculated before the feature values were normalized. The reason for this was to gain a more accurate estimate of the independence between the features. Moreover, as the text embedding features consist of 768 values each, the mutual information cannot be calculated the same way as for the other features. As such, the mutual information between the text embedding features was calculated by finding the mutual information between each pair of text embedding vectors, for each dataset sample, and averaging across all the samples.

The mutual information was deemed most important for the text embedding- and sentiment analysis-based features. As these features are computationally intensive (and the text embedding features each add a large number of dimensions to the feature space), it was preferred to only use one feature from each of these categories for the final accuracy testing. This is similar to the Fakeddit model, which only use a single text embedding feature – although not explicitly attributing this to feature space minimization [Nakamura et al., 2020]. Further, the two sentiment analysis features and the four text embedding-based features were expected to have a high degrees of dependence between them – leading to the assumption that using more than one of them was not worth the computational effort. The full feature testing plan is given in Table 5.2.

Hypotheses of FT-1, FT-2 and FT-3

The sentiment analysis-based features were expected to have high feature scores for all datasets, due to their sophistication compared to the other non-text embedding features. Feature 1 (swear word TF) and feature 17 (word count) were expected to have low features scores across all the datasets, as these are relatively simplistic and as such may not be able to accurately capture the news article characteristics. Further, there may be considerable variation of feature scores (for the same features), on the different datasets. The Kaggle dataset is generally easier to classify than the two other datasets and as such, some features may work well on the Kaggle dataset but not on the Liar and FNN datasets. Features 9 (strongly subjective words), 12 (TF of negative opinion words), 13 (TF of positive words) and 14 (exclamation + question marks TF) were expected to belong to this category of features.

Hypotheses of FT-4, FT-5 and FT-6

Feature 20 (headline text embedding) was expected to perform slightly better than the other text embedding features, across all three datasets. This expectation was based on the theory that the article headlines serve as summaries

ID	**Test Description**	**Goal**
FT-1	Perform feature scoring of all (non-text embedding) features, on the **Kaggle** dataset	Score features according to what value they bring to the task of fake news classification
FT-2	Perform feature scoring of all (non-text embedding) features, on the **Liar** dataset	
FT-3	Perform feature scoring of all (non-text embedding) features, on the **FakeNewsNet** dataset	
FT-4	Perform feature ranking of all the text embedding features, on the **Kaggle** dataset	Rank text embedding features according to what value they bring to the task of fake news classification
FT-5	Perform feature ranking of all the text embedding features, on the **Liar** dataset	
FT-6	Perform feature ranking of all the text embedding features, on the **FakeNewsNet** dataset	
FT-7	Calculate mutual information between all (non-text embedding) features, on the **Kaggle** dataset	Identify features which have high levels of dependence between them
FT-8	Calculate mutual information between all (non-text embedding) features, on the **Liar** dataset	
FT-9	Calculate mutual information between all (non-text embedding) features, on the **FakeNewsNet** dataset	
FT-10	Calculate mutual information between the text embedding features, on the **Kaggle** dataset	Identify text embedding features which have high levels of dependence between them
FT-11	Calculate mutual information between the text embedding features, on the **Liar** dataset	
FT-12	Calculate mutual information between the text embedding features, on the **FakeNewsNet** dataset	

Table 5.2: The testing plan for feature testing

capturing the writing style and opinions/sentiments of the articles. It was theorized that the first and last sentence of the article texts were less representative of the article class, both used together (head + tail, feature 21) and separately (feature 22 and 23).

Hypotheses of FT-7, FT-8 and FT-9

Feature 9 (strongly subjective words) and 12 (negative opinion words) were expected to have some dependence, as both features reflect a very subjective writing style. Further, feature 17, 18 and 19 (word count, FK grade level and FK reading ease) were expected to share some mutual information, as these features are based on similar parameters. Finally, features 24 and 25 (the two sentiment analysis features) were expected to have considerable dependence, as the sentiment of the headline and article text corpus most likely are written in a similar tone.

Hypotheses of FT-10, FT-11 and FT-12

All four text embedding features were expected to have some dependency, as they are all calculated using BERT. Further, feature 22 (first sentence) and 23 (last sentence) were expected to have considerable levels of mutual information with feature 21 (concatenated first and last sentence), as these features share parts of the input data. Feature 20, which use the article headlines as input, was expected to have the least amount of mutual information with the other text embedding features.

5.2.3 Accuracy Testing

For the accuracy testing, accuracy and standard deviation were calculated on all the datasets. The accuracy testing was performed with subsets of the total amount of proposed features. The reasoning behind only using a subset of features was that several of the features were computationally heavy to extract. If all the features were to be used in the accuracy testing phase, the amount of data samples used would have had to be very low – to not exceed the available computational power.

In addition, no preliminary testing was conducted on the features themselves, so some of them may be unfit for usage and therein detrimental to the accuracy of the model. By evaluating the features separately, the worst-performing ones may be excluded from the subsequent testing. Moreover, using several text embedding features would result in a large feature space, which is sub-optimal for the AIS model and may thus induce poor results. As such, two feature subsets were used for accuracy testing, on each of the three datasets. These feature subsets were determined after reviewing the results of the feature testing phase.

The first feature subset (for a given dataset) simply includes the best-performing features found on that dataset (in the feature testing phase) – disregarding the text embedding features. The second feature subset is equal to the first, but also includes the best-performing text embedding feature (on the particular dataset). The results obtained with the two feature subsets could then be compared, to reason about the performance of the text embedding feature and evaluate the proposed model's performance on large feature spaces.

Additionally, it was decided to test the model's performance using two universal feature subsets, which were kept the same for all the datasets. Similarly to the dataset-specific feature sets, one of these feature subsets included the best-performing features (excluding the text embedding features) and one including a text embedding feature. The reason for testing with universal feature sets was to test the model's general applicability, as the overall goal of the thesis is to design a robust model capable of classifying news articles found ”in the wild”.

The feature subsets were determined upon the completion of the feature testing phase, so that a balance between minimizing the feature count and at the same time not discarding valuable features could be found. When deciding the universal feature subsets, features scoring well across all the datasets were generally favoured, although very high performance on single datasets was also considered. The results obtained were then compared to the results of other state of the art models, in Section 5.4.3. The full accuracy testing plan is given in Table 5.3 and the respective feature sets are given in Subsection 5.4.2.

Hypotheses of AT-1, AT-2 and AT-3

The proposed model was expected to produce stable (low standard deviation) accuracy results which may compete with current state of the art results, on all three datasets. As the feature sets for these tests didn't include any text embedding features, the resulting feature space was much smaller than for AT-4, -5 and -6. It was theorized that this may result in lower accuracy upon initialization (as valuable text embedding information was not used), but that the model would have an easier time improving accuracy throughout the generations. As such, the results were expected to be more stable than the results from AT-4, AT-5 and AT-6.

ID	**Description**	**Goal**
AT-1	Calculate binary classification accuracy and standard deviation on the **Kaggle** dataset, using feature set 1	Obtain an estimate of accuracy, which may be compared to other state of the art results
AT-2	Calculate 6-class classification accuracy and standard deviation on the **Liar** dataset, using feature set 3	
AT-3	Calculate binary classification accuracy and standard deviation on the **FakeNewsNet** dataset, using feature set 5	
AT-4	Calculate binary classification accuracy and standard deviation on the **Kaggle** dataset, using feature set 2	Obtain an estimate of accuracy, which may be compared to other state of the art results + investigate the effects of large feature spaces on the accuracy results
AT-5	Calculate 6-class classification accuracy and standard deviation on the **Liar** dataset, using feature set 4	
AT-6	Calculate binary classification accuracy and standard deviation on the **FakeNewsNet** dataset, using feature set 6	
AT-7	Calculate classification accuracy and standard deviation on the **Kaggle**, **Liar** and **FakeNewsNet** datasets, using (the universal) feature set 7	Obtain an estimate of the model's robustness to feature subset selection
AT-8	Calculate classification accuracy and standard deviation on the **Kaggle**, **Liar** and **FakeNewsNet** datasets, using (the universal) feature set 8	Obtain an estimate of the model's robustness to feature subset selection + investigate the effects of large feature spaces on the accuracy results

Table 5.3: The testing plan for accuracy testing

Hypotheses of AT-4, AT-5 and AT-6

The proposed model was expected to produce good accuracy results, on all three datasets. Further, some uncertainty was prevalent when it came to the inclusion of the text embedding feature. It was theorized that the feature would be highly valuable for classification, but that the large amount of feature values would be problematic. The model refinement testing did not include testing on datasets with more than 13 features, so the model's ability to handle large feature spaces, which the text embedding features produce, was not known. The local feature selection scheme was expected to mitigate the dimensionality problem, but the effectiveness of the scheme was not clear. As large feature spaces generally have been a problem for AIS models, it was theorized that the model might have problems improving the accuracy after initialization – leading to static accuracies throughout the generations.

Hypotheses of AT-7 and AT-8

Due to the proposed model's local feature selection scheme, it was theorized that the accuracy results obtained from the universal feature sets would only be slightly worse than the ones obtained with the dataset-specific feature sets. This theory was based on the assumption that the local feature selection scheme would discard the features which weren't valuable for classifying the given dataset – therein reducing the problem of feature set selection.

5.3 Experimental Setup

5.3.1 Model Refinement Testing

MT-1 and MT-2 were run with varying parameter values, as the goal of these tests was to investigate the impact of AIS design choices. The parameters used for obtaining the results shown in the solution plots are presented in the respective figures. Further, the benchmark datasets used for preliminary accuracy testing are presented in Table 5.4. The selection of datasets was based on their regular use as benchmark datasets for classification models and as such, the accuracy results could easily be compared to those of state of the art AIS classification models. Additionally, the datasets contain a varying amount of samples, features and classes – testing the robustness of the proposed model.

Dataset	**Samples**	**Number of classes**	**Number of features**	**Reference**
Iris	150	3	4	[Dua and Graff, 2017]
Wine	178	3	13	[Dua and Graff, 2017]
Diabetes	768	2	8	[Dua and Graff, 2017]

Table 5.4: The benchmark datasets used for preliminary accuracy testing

The parameter values used for the preliminary accuracy testing are presented in Table 5.5. These values were kept the same for all the datasets. n_f is the length of the feature vector, i.e. the amount of features for the respective dataset.

Parameter	**Value**
Cross validation split (k)	5
Generations (G)	200
Antibody ratio (AB$_{ratio}$)	1.0
Antibody initialization split (AB$_{init-split}$)	0.5/0.5
Antibody replacement ratio (AB$_{rr}$)	0.1
Feature vector mutation probability (p$_{fv}$)	1/n$_f$
RR radius mutation probability (p$_r$)	1/n$_f$
Antibody clones (AB$_c$)	10
Antibody removal threshold (AB$_{rt}$)	0.01
Number of runs	5

Table 5.5: The parameter values used for preliminary testing

5.3.2 Feature and Accuracy Testing

The fake news classification datasets used for accuracy and feature testing are given in Table 5.6, together with their respective characteristics.

Dataset	**Total training samples**	**Number of classes**	**Reference**
Liar	15052	6	[Wang, 2017]
Kaggle	20387	2	[Kaliyar et al., 2021]
FakeNewsNet	15212	2	[Shu et al., 2018]

Table 5.6: The datasets used for feature and accuracy testing

The model parameters used for the feature and accuracy testing are given in Table 5.7. Preliminary testing showed that the model had trouble evolving the antibody population in a constructive way, on the FakeNewsNet (FNN) dataset. As such, it was chosen to adapt slighly different parameters for the tests which used the FNN dataset (AT-3, AT-6 and the FNN runs of AT-7 and AT-8). Finally, the feature sets used for the accuracy testing were determined upon the completion of the feature testing. These feature sets can be found in Subsection .

Parameter	**Tests using FNN dataset**	**Remaining tests**
Cross validation split (k)	4	4
Generations (G)	100	100
Antibody ratio (AB$_{ratio}$)	1.0	1.0
Antibody initialization split (AB$_{init-split}$)	0.5/0.5	0.9/0.1
Antibody replacement ratio (AB$_{rr}$)	0.1	0.05
Feature vector mutation probability (p$_{fv}$)	1/n$_f$	1/n$_f$
RR radius mutation probability (p$_r$)	1/n$_f$	1/n$_f$
Antibody clones (AB$_c$)	10	10
Antibody removal threshold (AB$_{rt}$)	0.01	0.01

Table 5.7: Parameter values used for feature and accuracy testing

5.3.3 Samples Used and Number of Runs

Due to limited available computational power during testing, all the dataset samples could not be used for testing. As such, a subset of samples was used for the different tests. These samples were randomly selected out of the total of sample amount (randomized for each model run). Due to difference in computational complexity, some tests could be run with more samples than others. The number of samples used for each test can be seen in Table 5.8.

Test ID	**Number of samples**
MT-1, -2, -3	All available samples
FT-1, -2, -3	1000
FT-4, -5, -6	500
FT-7, -8, -9	1000
FT-10, -11, -12	300
AT-1, -2, -3	1200
AT-4, -5, -6	800
AT-7	1200
AT-8	800

Table 5.8: Samples sizes used for testing

A varying number of runs were used for the various tests conducted in the feature and accuracy testing phases. The reason for this was two-fold. Firstly, most of the tests in the feature testing phase did not need to be run multiple times – as they are relatively deterministic. For feature test 1-9, the only difference between runs would be the samples used (as these are selected at random). However, it was decided that the large sample sizes used would make this variety of samples insignificant. The number of runs used for each test in the feature and accuracy testing are given in Table 5.9.

Test ID	**Number of runs**
FT-1, -2, -3, -4, -5, -6, -7, -8, -9	1
FT-10, -11, -12	3
AT-1, -2, -3, -7	5
AT-4, -5, -6, -8	3

Table 5.9: The number of runs for feature and accuracy testing

5.4 Experimental Results

5.4.1 Model Refinement Testing Results

RR Radius Initialization (MT-1)

The difference in overall RR radii sizes can be seen clearly when studying Figure 5.4. Every parameter except for the RR radius initialization approach was kept the same for both runs. As expected, the VALIS approach results in significantly larger RR radii than the AISLFS approach, as shown by the intense colors in the VALIS plot. Interestingly, the RR radii stay larger even after the antibodies are given significant time to mutate (the plots are from generation 200). As the RR radii are larger with the VALIS approach, the coverage of the feature space is likely also greater with this approach [Baug et al., 2019]. This increased coverage might lead to improved performance on high-dimensional data, which is significant for the proposed model.

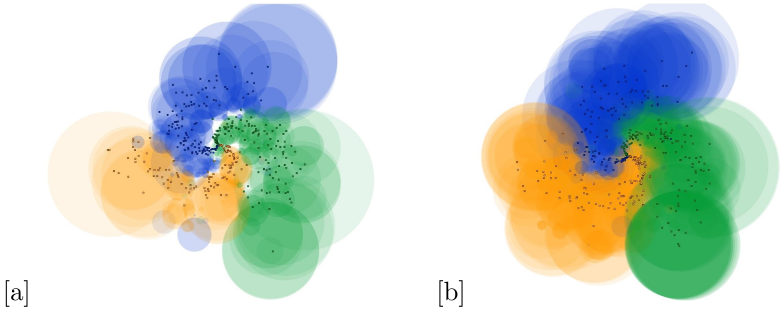


Figure 5.4: *RR radius initialisation using AISFLS [a] and VALIS [b] (MT-1)*

The training set accuracy plots for the two RR radius initialization schemes are shown in Figure 5.5 and Figure 5.6. These are the average of 5 runs with each initialization scheme. The plots highlight the impact of the scheme selection. As expected, the AISLFS initialization approach (Figure 5.5) resulted in very high accuracies from the very first generation. Half of the antibodies are initialized with the training set antigens' feature values and the RR radii are initialized to not contain any different class antigens. Thus, it was expected that the accuracy would be high from the start. The initial decrease in accuracy is due to the replacement of the poorest performing antibodies, regardless of whether their replacements have higher fitness values. The RR radii are being initialized as large as possible, without including any different class antigens. As such, a small mutation to the RR radius (making it larger) or to the feature values (moving the antibody towards its closest different-class antigen) will lead to the inclusion of a different-class antigen into the antibody's RR – therein reducing the weighted accuracy and thus the fitness score. Thus, antibodies which had high fitness scores just one generation before might be replaced the next generation. It takes some time for the antibodies to assess their local feature space and therefore the upward trend is slower than than the initial drop.

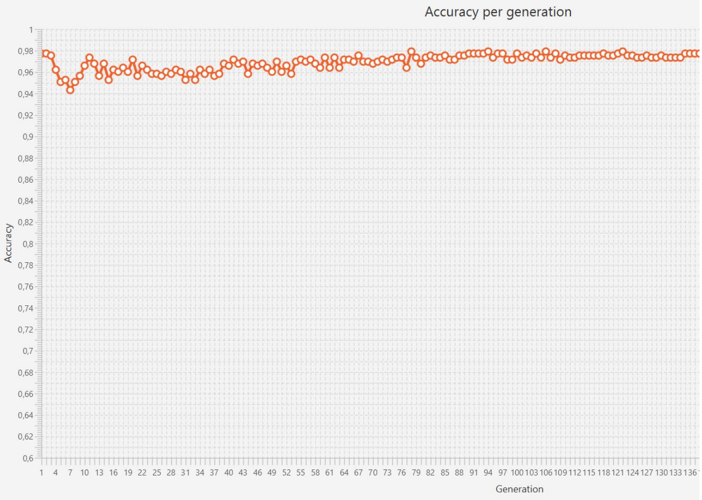


Figure 5.5: *Using AISLFS' RR radius initialization approach (MT-1)*

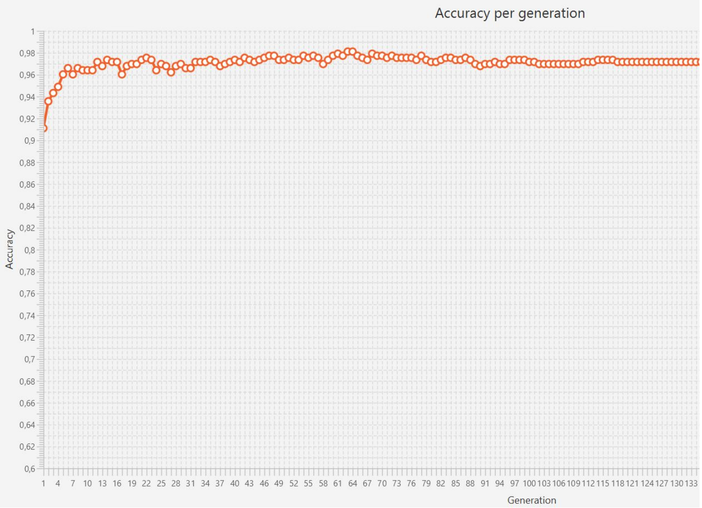


Figure 5.6: *Using VALIS' RR radius initialization approach (MT-1)*

Comparatively, the VALIS initialization approach (Figure 5.6) produced lower accuracies initially – but efficiently increasing through the first generations, before converging. Although the AISLFS approach generally achieved higher accuracies on the training set than the VALIS approach, the accuracies on the testing sets were generally worse, as can be seen from Table 5.10. These values were calculated from 5 runs with each initialization scheme, on the Wine dataset. The fact that the AISLFS approach scores higher on the training sets, but not the testing sets, is probably due to some degree of overfitting to the training set samples, as the RR radii are initialized to not include any antigens of a different class – therein being much smaller. This allows for less exploration and less robustness to unseen antigen samples. It should also be noted that the testing set standard deviation is larger for AISLFS than for VALIS. This is attributed to the overfitting problem.

RR initialization approach	**Training set**	**Testing set**
AISLFS	0.965 (0.004)	0.922 (0.022)
VALIS	0.958 (0.004)	0.936 (0.006)

Table 5.10: Accuracies (STD) on the training and testing sets (using Wine dataset)

Overall, the VALIS approach was deemed the most suited for the proposed model. This was both due to the generally higher and more stable results on the testing set and the larger RR radii. Having large RR radii was deemed desirable, as the proposed model had to handle high-dimensional data during the fake news testing.

Antibody Replacement Ratio (MT-2)

As can be seen from comparing Figure 5.7 and Figure 5.8, the antibody replacement ratio (AB_{rr}) parameter has a significant impact on the model accuracy. In Figure 5.7, where an antibody replacement ratio of 0.3 have been used, the testing set accuracy fluctuates wildly the first 50 generations. This is due to almost half the antibodies being replaced by mutated clones during the first generations. Although still ending up at a decent accuracy, the graph has fallen to this accuracy and the trend is strictly downwards. Furthermore, the fluctuation is not optimal as the model is inherently less stable and more dependent on randomly discovering favourable mutations (leading to more variable runs). The reason for testing such large values for the antibody replacement ratio was that VALIS [Karpov et al., 2018] uses a static value of 0.5 for the ratio, as mentioned in Subsection 4.2.8.

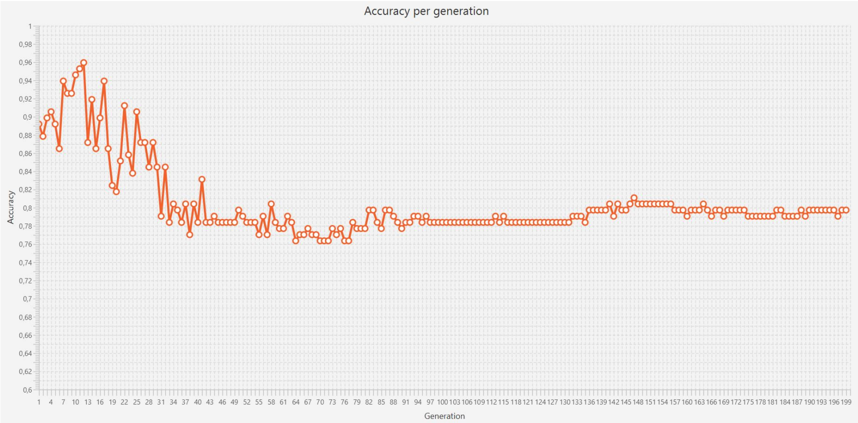


Figure 5.7: *Using a antibody replacement ratio of 0.3 (MT-2)*

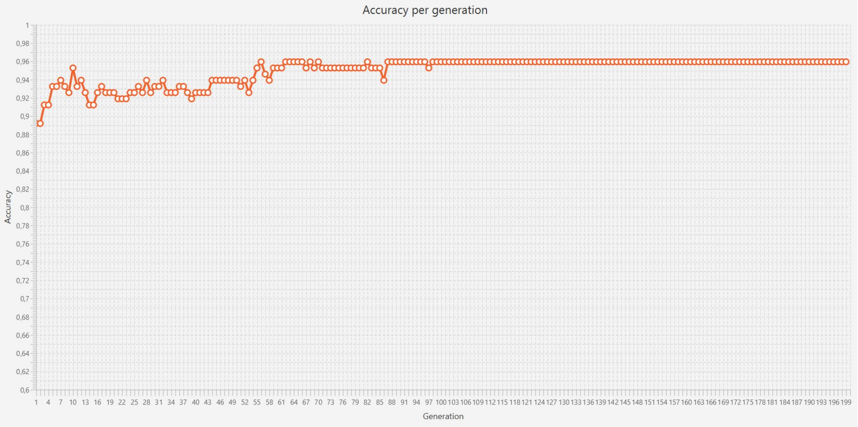


Figure 5.8: *Using an antibody replacement ratio of 0.1 (MT-2)*

An antibody replacement ratio of 0.1 produces more stable results, as can be seen in Figure 5.8. The accuracy increases quickly the first generations and converges after around 50 generations. The trend is also generally upwards, toward a higher accuracy. It could be argued that such a low replacement ratio doesn't provide enough exploration, as only a maximum of 10% of the antibodies are replaced – declining throughout the generations. However, each antibody selected for reproduction each produces multiple clones with random mutations to feature values, RR radius and feature subset selection. This heuristic introduces considerable diversity into the population, even though relatively few antibodies are replaced.

Additionally, it should be noted that the plots are based on the accuracies obtained on the testing sets, which the model doesn't use for training. A copy of the antibody population is used at each generation to compute the testing set accuracy, in order to prevent knowledge leakage. As such, the antibody population evolves without ”seeing” the antigens which the accuracy plots are calculated from. The occasional dips in accuracy are therefore to be expected.

Preliminary Accuracy Testing (MT-3)

The results of the preliminary accuracy testing are presented in Table 5.11. The accuracy results are averaged over the 5 runs, for each dataset, with the standard deviations enclosed in parentheses. For the AISLFS accuracy scores, the results obtained using the second affinity measure (similar to the one used by the proposed model) and the Euclidean distance metric was used [Dudek, 2012]. As two of the affinity measures in AISLFS scored roughly the same, the one most similar to the proposed model was chosen.

Dataset	**Proposed model**	**VALIS**	**MAIM**	**AISLFS**	**AIRS**
Iris	0.959 (0.005)	0.956 (0.006)	0.965 (0.006)	0.954 (0.005)	0.960 (0.019)
Wine	0.944 (0.005)	0.972 (0.05)	0.967 (0.03)	0.974 (0.05)	n/a
Diabetes	0.702 (0.006)	n/a	0.757 (0.006)	0.740 (0.009)	0.742 (0.044)

Table 5.11: Accuracy results (STD) for preliminary accuracy testing

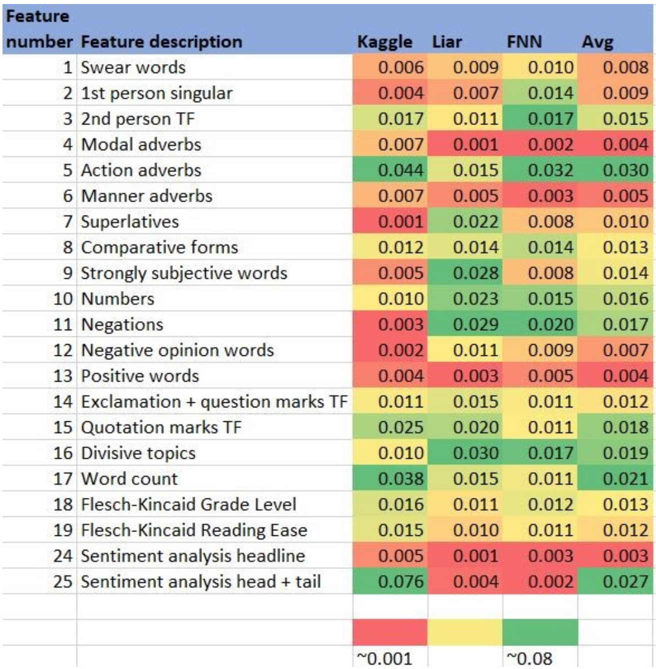
As expected, the obtained results generally aren't quite on par with the other state of the art AIS models in pure classification accuracy. This was to be expected as the model has neither been designed nor parameter tuned towards these datasets specifically. Notably, the model performs best on the Iris dataset, scoring slightly better than both VALIS and AISLFS – which served as considerable inspiration for the implementation of the model. The high performance on the Iris dataset might be due to the low amount of features for this dataset, compared to the two other datasets. The accuracy results on the Iris dataset also indicate that the relatively small amount of samples is not a problem for the model. Conversely, the model's poorest results (compared to the other models) are on the Diabetes dataset, which have about 5 times as many data samples as the Iris and Wine datasets. However, the Diabetes dataset is notorious for being difficult to perform classification on – so there's no clear correlation between increasing amounts of data samples and decreasing classification accuracy for the proposed model. Additionally, the low standard deviations for all the datasets provide additional weight for the obtained results – indicating stable runs and a robust model.

Although not performing quite on par with the other models on the Wine and Diabetes datasets, the accuracy results of the proposed model are well within the acceptable range. The model performs reasonably well on all datasets, which shows robustness to varying numbers of samples, classes and features. As the general applicability was the most important aspect to evaluate in the preliminary accuracy testing, the overall performance of the model was acceptable. Poor accuracy results on fake news classification would likely be caused by the feature extraction/selection and not the underlying model itself.

5.4.2 Feature Testing Results

FT-1, FT-2 and FT-3

The results of the feature scoring of the non-text embedding features are given in Figure 5.9. The respective feature scores are presented column-wise, with one column per dataset. Additionally, the average score for each feature, across all three datasets, are presented in the rightmost column. To improve readability, color grading has been applied to the values, ranging from green (highest 10% in column), through yellow, to red (lowest 10% in column). It should be noted that this coloring is relative to each column, not across all the columns. The averaged feature scores in Figure 5.9 provide insight into the performance of the various (non-text embedding) features. The scores are generally quite low, which was expected due to the difficulty of fake news classification compared to using benchmark classification datasets (where feature values aren't extracted). For comparison, the scores of the features from the Iris dataset generally span the 0.15-0.5 range, i.e. being roughly 10 times more valuable for classification than the best extracted features. Notably, the performance of the features vary sig-



Feature number	Feature description	Kaggle	Liar	FNN	Avg
1	Swear words	0.006	0.009	0.010	0.008
2	1st person singular	0.004	0.007	0.014	0.009
3	2nd person TF	0.017	0.011	0.017	0.015
4	Modal adverbs	0.007	0.001	0.002	0.004
5	Action adverbs	0.044	0.015	0.032	0.030
6	Manner adverbs	0.007	0.005	0.003	0.005
7	Superlatives	0.001	0.022	0.008	0.010
8	Comparative forms	0.012	0.014	0.014	0.013
9	Strongly subjective words	0.005	0.028	0.008	0.014
10	Numbers	0.010	0.023	0.015	0.016
11	Negations	0.003	0.029	0.020	0.017
12	Negative opinion words	0.002	0.011	0.009	0.007
13	Positive words	0.004	0.003	0.005	0.004
14	Exclamation + question marks TF	0.011	0.015	0.011	0.012
15	Quotation marks TF	0.025	0.020	0.011	0.018
16	Divisive topics	0.010	0.030	0.017	0.019
17	Word count	0.038	0.015	0.011	0.021
18	Flesch-Kincaid Grade Level	0.016	0.011	0.012	0.013
19	Flesch-Kincaid Reading Ease	0.015	0.010	0.011	0.012
24	Sentiment analysis headline	0.005	0.001	0.003	0.003
25	Sentiment analysis head + tail	0.076	0.004	0.002	0.027
		🟥	🟨	🟩	
		~0.001		~0.08	

Figure 5.9: *The feature scores of the non-text embedding features (FT-1, -2, -3)*

nificantly between the three datasets – illustrating the importance of testing on several datasets. The feature scores vary less between the Liar and FNN datasets, which was expected as these are harder to classify than the Kaggle dataset. Thus, features like sentiment analysis, which only outputs 4 different values, are inefficient on these datasets. Surprisingly, none of the features that were expected to perform better on the Kaggle dataset (feature 9, 12, 13 and 14) actually met their expectation. In fact, feature 9 and 12 actually scored better on the Liar and FNN datasets than on the Kaggle dataset. It is unclear why this is the case, though it may be due to the Kaggle fake news articles having limited vocabularies – thus not matching with the relatively advanced terms of these lexicons.

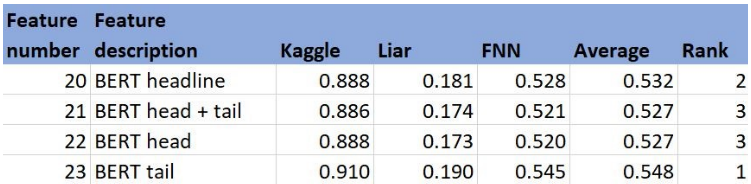
The sentiment analysis feature using *head + tail* scored better than the one using the article headlines. In fact, the headline sentiment analysis feature scored the worst of all the features tested. This hints that the article headlines are less representative of the article content than theorized. Further, the high averaged score of the *head + tail* sentiment analysis feature is mostly attributed to the high performance on the Kaggle dataset (the highest achieved score of all the features), which pulls the average score up significantly. This difference in performance across the datasets is most likely due to the varying difficulties of the datasets. As the sentiment analysis features produce relatively coarse-grained feature values (integers in the [0,4] range, before normalization), articles without extreme sentiments (i.e. relatively neutral) result in very similar values. It was theorized that the news articles in the Kaggle datasets are generally more extreme in their expressed opinion and as such sentiment analysis works better on this dataset.

Feature 1 (swear word TF) scored relatively poorly (as expected), but notably several other features scored worse. Among the features scoring worse were two of the TF features which Rashkin et al. found to be very representative of article class – further highlighting the importance of evaluating feature performance across multiple datasets. Surprisingly, feature 17 (word count) achieved the third best average score. Upon further investigation, it was discovered that the word count ratio was the inverse of the theorized ratio – with the average fake news article being almost twice as long as the average real news article (in the Kaggle dataset).

Additionally, the two novel features 15 (quotation mark TF) and 16 (divisive topics) perform well across all the dataset. These results highlight the need for creativity when researching applicable features for fake news classification.

FT-4, FT-5 and FT-6

The obtained classification accuracies using the four text embedding features (individually) are presented in Figure 5.10. The accuracies obtained from the various datasets are given column-wise, including the average accuracy for each feature. Additionally, the ranking of each text embedding feature is given in the rightmost column, based on the average classification accuracy. Contrarily to the hypotheses of FT-4, FT-5 and FT-6, the headline text embedding feature did not outperform the other text embedding features. This supports the findings from the testing of the sentiment analysis features (in FT-1, FT-2 and FT-3) – where it was found that the news article headlines generally were less representative of article class labels than expected. Surprisingly, the *tail* approach (where only the last sentence is used) scored better than all the other text embedding approaches, on all three datasets. This is notable and may be used in future research into fake news classification. It is hypothesized that this might be due to the nature of fake news articles specifically, where the authors wish to end the articles by pushing a narrative or stating a drastic opinion. Unfortunately, the sentiment analysis features did not employ the separate *head* and *tail* approaches, only the fused approach where the head and tail are concatenated. As such, it couldn't be determined if the separate *tail* approach would perform better for sentiment analysis as well.

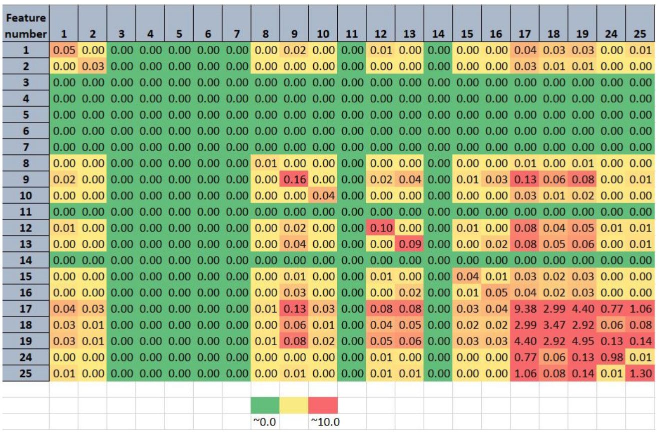


Feature number	Feature description	Kaggle	Liar	FNN	Average	Rank
20	BERT headline	0.888	0.181	0.528	0.532	2
21	BERT head + tail	0.886	0.174	0.521	0.527	3
22	BERT head	0.888	0.173	0.520	0.527	3
23	BERT tail	0.910	0.190	0.545	0.548	1

Figure 5.10: *The feature evaluations of the text embedding features (FT-4, -5, -6)*

FT-7, FT-8 and FT-9

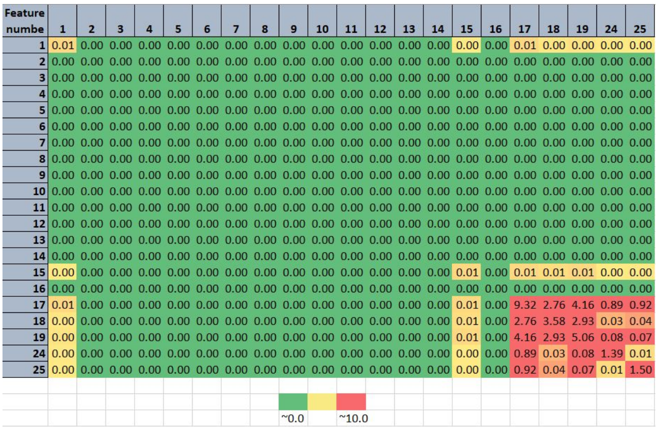
The mutual information matrices for the Kaggle, Liar and FNN datasets are given in Figure 5.11, 5.12 and 5.13, respectively. It should be noted that the matrices are symmetrical, i.e. $MI(X,Y) = MI(Y,X)$. Further, the varying values found on the diagonals are attributed to the varying representativeness of the features extracted (MI of two similar, constant values are 0) and the fact that the probability distributions are estimated using only parts of the total amount of dataset samples. The values are color graded inversely of the feature scores, with the lowest values colored green and the largest colored red. This is due to high levels of mutual information being inherently negative in this setting.



Feature number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	24	25
1	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.04	0.03	0.03	0.00	0.01
2	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.01	0.01	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00
9	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.16	0.00	0.00	0.02	0.04	0.00	0.01	0.03	0.13	0.06	0.08	0.00	0.01
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.01	0.02	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.10	0.00	0.00	0.01	0.00	0.08	0.04	0.05	0.01	0.01
13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.00	0.09	0.00	0.00	0.02	0.08	0.05	0.06	0.00	0.01
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.04	0.01	0.03	0.02	0.03	0.00	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.02	0.00	0.01	0.05	0.04	0.02	0.03	0.00	0.00
17	0.04	0.03	0.00	0.00	0.00	0.00	0.00	0.01	0.13	0.03	0.00	0.08	0.08	0.00	0.03	0.04	9.38	2.99	4.40	0.77	1.06
18	0.03	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.01	0.00	0.04	0.05	0.00	0.02	0.02	2.99	3.47	2.92	0.06	0.08
19	0.03	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.08	0.02	0.00	0.05	0.06	0.00	0.03	0.03	4.40	2.92	4.95	0.13	0.14
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.77	0.06	0.13	0.98	0.01
25	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.01	0.00	0.00	0.00	1.06	0.08	0.14	0.01	1.30

🟩 🟨 🟥
~0.0 ~10.0

Figure 5.11: *The MI matrix for the Kaggle dataset (FT-7)*

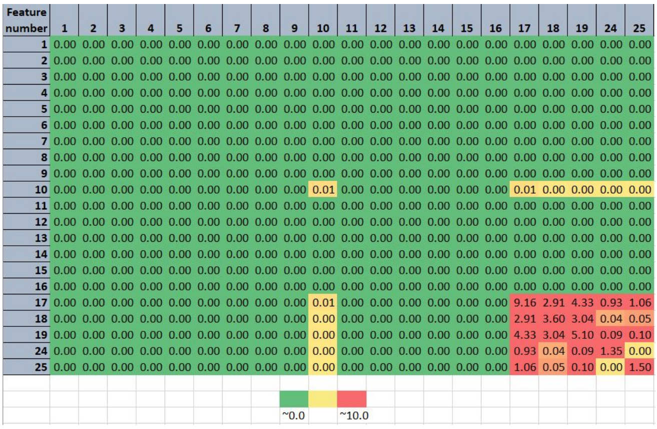


Feature numbe	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	24	25
1	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.01	0.01	0.00	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	9.32	2.76	4.16	0.89	0.92
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	2.76	3.58	2.93	0.03	0.04
19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	4.16	2.93	5.06	0.08	0.07
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.89	0.03	0.08	1.39	0.01
25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.92	0.04	0.07	0.01	1.50

🟩 🟨 🟥
~0.0 ~10.0

Figure 5.12: *The MI matrix for the Liar dataset (FT-8)*

The differences of mutual information between the features are clear when comparing the three tables. For the Kaggle dataset (Figure 5.11), there are slight dependencies found for most of the features. Comparatively, there is very little dependency to be found for the Liar (Figure 5.12) and FNN (Figure 5.13) datasets. The differences in mutual information are attributed to the varying degrees of classification difficulties of the three datasets – which result in differences in feature representativeness.



Feature number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	24	25
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	9.16	2.91	4.33	0.93	1.06
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.91	3.60	3.04	0.04	0.05
19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.33	3.04	5.10	0.09	0.10
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.93	0.04	0.09	1.35	0.00
25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.06	0.05	0.10	0.00	1.50

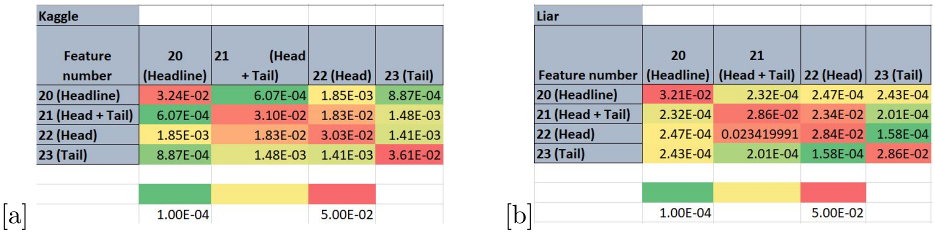
Figure 5.13: *The MI matrix for the FakeNewsNet dataset (FT-9)*

As hypothesized, feature 9 (strongly subjective words) and 12 (negative opinion words) have some dependency, for the Kaggle dataset. However, both features have higher levels of mutual information with feature 16 (divisive topics), 17 (word count), 18 (FK grade level) and 19 (FK reading ease) than they have between them. This reflects both the difficulty of predicting feature usefulness before testing and the potential of the Flesch-Kincaid features for identifying subjective and simplistic writing styles.

Interestingly, for all the datasets, most of the dependencies were found between the last five features. It was expected that feature 17, 18 and 19 would have some mutual information – as the word count is used for all of these features. However, the expected dependence between the two sentiment analysis features was not found. The mutual information between these features are generally low across all the datasets. Coupled with the results from the feature scoring, this finding further supports the theory that the headlines of the news articles are generally not very representative of the class labels.

FT-10, FT-11 and FT-12

The mutual information (MI) matrices for the text embedding features, on all three datasets, can be seen in Figure 5.14 and 5.15. As mentioned in the feature testing plan, these values were calculated for each pair of features at a time, averaging the MI between each pair of the 768 feature values. In Figure 5.15 [b], the MI matrix has been averaged across all the datasets, in order to get a sense of dependency between the text embedding features. The color grading is set up



[a]

Kaggle				
Feature number	20 (Headline)	21 (Head + Tail)	22 (Head)	23 (Tail)
20 (Headline)	3.24E-02	6.07E-04	1.85E-03	8.87E-04
21 (Head + Tail)	6.07E-04	3.10E-02	1.83E-02	1.48E-03
22 (Head)	1.85E-03	1.83E-02	3.03E-02	1.41E-03
23 (Tail)	8.87E-04	1.48E-03	1.41E-03	3.61E-02

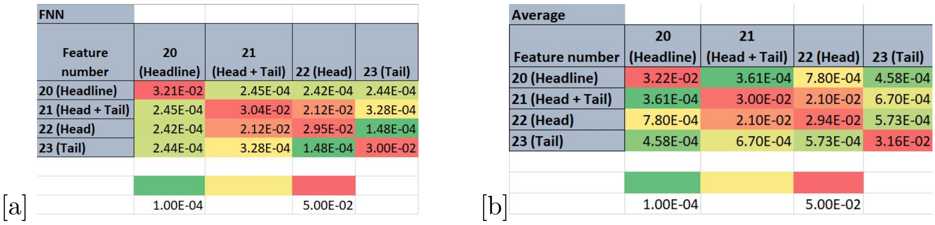
🟩	🟨	🟥	
1.00E-04		5.00E-02	

[b]

Liar				
Feature number	20 (Headline)	21 (Head + Tail)	22 (Head)	23 (Tail)
20 (Headline)	3.21E-02	2.32E-04	2.47E-04	2.43E-04
21 (Head + Tail)	2.32E-04	2.86E-02	2.34E-02	2.01E-04
22 (Head)	2.47E-04	0.023419991	2.84E-02	1.58E-04
23 (Tail)	2.43E-04	2.01E-04	1.58E-04	2.86E-02

🟩	🟨	🟥	
1.00E-04		5.00E-02	

Figure 5.14: *The MI matrix for the text embedding features, for the Kaggle [a] and Liar [b] dataset (FT-10, -11)*



[a]

FNN				
Feature number	20 (Headline)	21 (Head + Tail)	22 (Head)	23 (Tail)
20 (Headline)	3.21E-02	2.45E-04	2.42E-04	2.44E-04
21 (Head + Tail)	2.45E-04	3.04E-02	2.12E-02	3.28E-04
22 (Head)	2.42E-04	2.12E-02	2.95E-02	1.48E-04
23 (Tail)	2.44E-04	3.28E-04	1.48E-04	3.00E-02

🟩	🟨	🟥	
1.00E-04		5.00E-02	

[b]

Average				
Feature number	20 (Headline)	21 (Head + Tail)	22 (Head)	23 (Tail)
20 (Headline)	3.22E-02	3.61E-04	7.80E-04	4.58E-04
21 (Head + Tail)	3.61E-04	3.00E-02	2.10E-02	6.70E-04
22 (Head)	7.80E-04	2.10E-02	2.94E-02	5.73E-04
23 (Tail)	4.58E-04	6.70E-04	5.73E-04	3.16E-02

🟩	🟨	🟥	
1.00E-04		5.00E-02	

Figure 5.15: *The MI matrix for the text embedding features, for the FNN dataset [a] and the averaged MI matrix [b] (FT-12)*

similar to the non-text embedding matrices, with larger values colored red and smaller values colored green. However, the color grading is based on the entire table, not each individual column.

The results mostly match the expectations, with the *headline* approach generally sharing the least mutual information with the other text embedding features. However, the *tail* approach is also generally independent of the other features. This might explain why the *tail* feature performed better than all the other text embedding features during the feature ranking. Also, the fact that the *headline* feature scored second-best on all the datasets during the feature ranking suggests that there is some correlation between independence and classification usefulness for the text embedding features.

Feature Testing Assessment

The features used for the accuracy testing were selected based on their feature scores/ranks on the individual datasets. Generally, the top five features for each datasets was used, although this depended on the relative feature scores. Mutual information was also considered when deciding the feature sets. Using two or more features with high levels of mutual information would unnecessarily increase the run time of the model, as these don't provide any new information. For the universal feature sets (sets 7 and 8), features scoring well across all three datasets were preferred. The feature sets used for the accuracy testing are given in Table 5.12.

Feature number	**Feature description**	**Set 1**	**Set 2**	**Set 3**	**Set 4**	**Set 5**	**Set 6**	**Set 7**	**Set 8**
3	2nd person pronouns	x	x			x	x	x	x
5	Action adverbs	x	x			x	x	x	x
7	Superlatives			x	x				
9	Strongly subjective words			x	x				
10	Numbers			x	x	x	x	x	x
11	Negations			x	x	x	x	x	x
15	Quotation marks	x	x	x	x			x	x
16	Divisive topics			x	x	x	x	x	x
17	Word count	x	x					x	x
23	BERT text embedding (tail)		x		x		x		x
25	Sentiment analysis (head + tail)	x	x						

Table 5.12: *The feature sets used for accuracy testing (x indicates the feature being used)*

5.4.3 Accuracy Testing Results

AT-1, AT-2 and AT-3

The results obtained in AT-1, -2 and -3 can be seen in Table 5.13. The standard deviations are added in parentheses. Generally, the proposed model achieved good accuracy results across all the datasets. Additionally, the standard deviations are low for all three datasets – indicating stable runs. The accuracy plots for AT-1, AT-2 and AT-3 runs are given in Figure 5.16 and 5.17. It should be noted that these plots don't include information from all the runs on each test, rather they serve as an example of a typical run for each test. Although producing competitive accuracy results, it is clear from the accuracy plots that the model struggles with increasing the accuracy. This is likely due to the model having trouble with evolving the antibody population in a favourable way, as the antigens are difficult to differentiate – rendering the fitness evaluation (and thus antibody selection) ineffective. For AT-1 (Figure 5.16 [a]), the accuracy is almost completely static, although antibodies are continuously replaced. For the plots of AT-2 (Figure 5.16 [b]) and AT-3 (Figure 5.17), the accuracy is evolving, but not in a functional way. In the test hypotheses section, it was hypothesized that the model would have difficulties with improving accuracy when the text embed-

Test	Feature set	Dataset	Accuracy
AT-1	1	Kaggle	0.872 (0.002)
AT-2	3	Liar	0.227 (0.004)
AT-3	5	FNN	0.580 (0.005)

Table 5.13: *The results of AT-1, AT-2 and AT-3, standard deviation in parentheses*

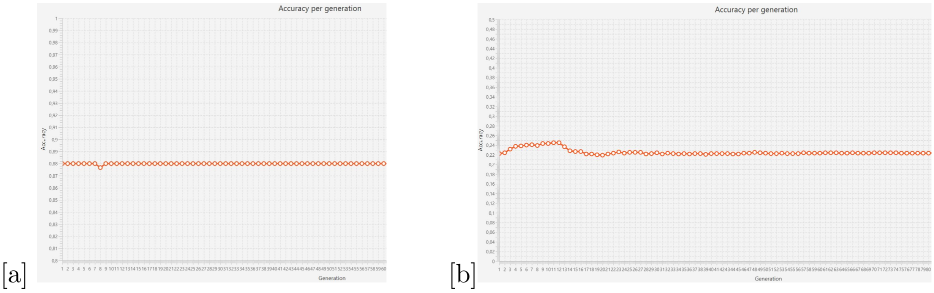


Figure 5.16: *Accuracy plot of AT-1 [a] and AT-2 [b]*

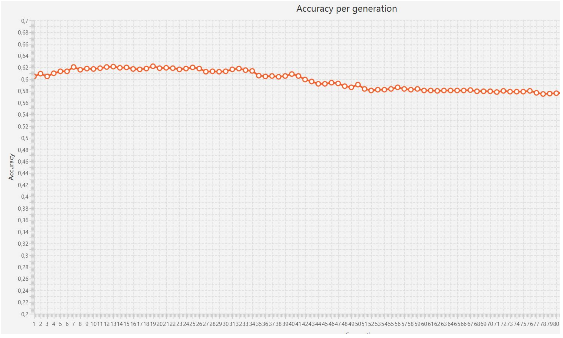


Figure 5.17: *Accuracy plot of AT-3*

ding features were used (due to the large number of features) but not without them. Given that the model didn't struggle with this during the preliminary accuracy testing, it was theorized that the extracted features were responsible for this lack of accuracy improvement. As the features used are extracted from raw news articles, they are not unequivocally representative of class labels. This was indicated through the low feature scores found in Subsection 5.4.2, compared to the features of a benchmark classification dataset like Iris. Thus, it was theorized that the lack of accuracy improvement was due to the antibodies struggling with separating different-class antigens in the feature space.

Although struggling with evolving the antibody population, the proposed model still produced results comparable with state of the art models. This is likely due to the initialization scheme which introduces parts of the antibodies randomly – therein covering the feature space to a sufficient degree. The voting scheme should likely also be credited, as antigen class labels are determined according to affinities, as well as the antibodies' feature selection vectors. It could therefore be argued that AISs might not be a bad choice for fake news classification, especially if more efficient feature extraction strategies are found. Further, it should be considered that the model was run with only a fraction of the total amount of samples for each dataset (see Table 5.8). With increased available computational power, the number of samples could also be increased. This would likely result in better accuracies, as the model would get a better overview of the data.

AT-4, AT-5 and AT-6

The accuracies obtained in AT-4, -5 and -6 are presented in Table 5.14.

Test	Feature set	Dataset	Accuracy
AT-4	2	Kaggle	0.917 (0.019)
AT-5	4	Liar	0.198 (0.004)
AT-6	6	FNN	0.513 (0.018)

Table 5.14: *The results of AT-4, AT-5 and AT-6, standard deviation in parentheses*

In the hypotheses of Subsection 5.2.3, there was considerable uncertainty tied to how the proposed model would handle large feature spaces. Further, it was theorized that the model might struggle with evolving the antibody population and that this would result in less stable runs than those of AT-1, -2 and -3. Interestingly, the model responds differently to the addition of text embedding features for the different datasets. The accuracy on the Kaggle dataset was improved, while those obtained on the Liar and FNN datasets were worsened. As the Kaggle dataset is objectively easier to classify than the other two datasets, there might be a correlation between difficulty of classification and effectiveness of text embeddings. This might also be compatible with the finding from the feature testing phase, where the sentiment analysis feature scored far better on the Kaggle dataset than on the other two datasets. As such, the sentiment analysis and text embedding features might be inherently too general for a specific task like fake news classification – leading to less effectiveness on difficult datasets which don't have clear negative/positive content. The standard deviations were also notably higher when the text embedding features were used. This indicates more volatile runs, as theorized in the hypotheses of Subsection 5.2.3.

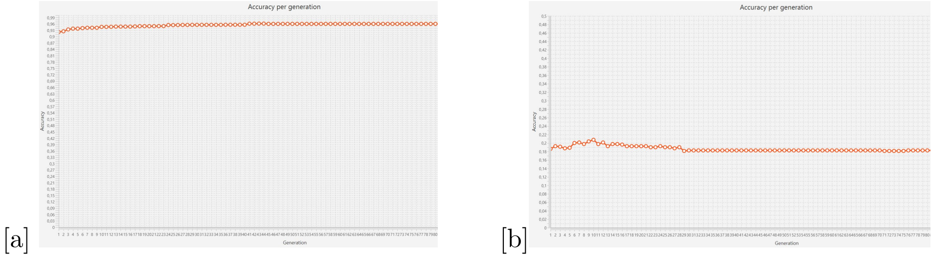


Figure 5.18: *Accuracy plot of AT-4 [a] and AT-5 [b]*

The accuracy plots of AT-4, AT-5 and AT-6 are presented in Figure 5.18 and 5.19. As for AT-1, -2 and -3, these plots don't use information from all the tests run, but rather they serve as examples of typical test runs. The plotted accuracy is the average accuracy at each generation, averaged for the k testing sets, for a single run. From the accuracy plots, it is clear that the model struggles similarly

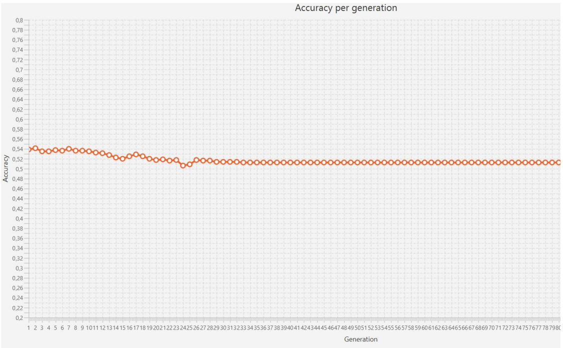


Figure 5.19: *Accuracy plot of AT-6*

to the non-text embedding runs (although improving slightly in AT-4). It was theorized that this was both due to the large feature space and the feature values themselves. Upon further investigation, it was found that many of the (training set) antigens only were covered by a single antibody – in the final generation. Additionally, considerable amounts of antibodies were deleted during the apoptosis process. This suggests that the antibodies struggle with locating the antigens in the high-dimensional feature space (leading to very low fitnesses). Although the antibodies are utilizing local feature selection, the chance of disregarding a large portion of the features is relatively low. If antibodies were to disregard all the text embedding-based feature values, they would have to randomly mutate so that only 7 out of 775 features are used. Even with a significant number of clones per antibody and many generations in which to mutate in, it is unlikely that the antibodies would evolve this way for the average run of the model. This means that the antibodies still have to handle large amounts of dimensions.

Finally, it should be noted that the run times were significantly increased when text embeddings were used. This extra processing time was due to both the feature extraction itself (handling requests through bert-as-service [Xiao, 2018]) and the extra processing time of evolving the antibody population with large amounts of features. As each antibody selected for reproduction produces a number of clones and each of these will mutate their feature values and feature selection vector – the computational effort increases drastically with increased number of features.

AT-7 and AT-8

The accuracies obtained using the universal feature sets are given in Table 5.15. As hypothesized, the accuracies are only slightly worse than those obtained with the dataset-specific feature sets (see Table 5.13 and 5.14). This suggests that the local feature selection scheme works as intended and antibodies disregard features which aren't useful for classifying a given dataset. This finding indicates that the model is robust to less effective features, when these these don't make up the vast majority of the features, i.e. when text embeddings are used. The accuracy plots of AT-7 and AT-8 are provided in the Appendix (Section 6.3.4). Considering the findings from AT-1 - AT-6, these were mostly as expected – although the model was able to improve the accuracy slightly on the Liar dataset in AT-8.

Test	**Feature set**	**Dataset**	**Accuracy**
AT-7	7	Kaggle	0.863 (0.004)
AT-7	7	Liar	0.226 (0.002)
AT-7	7	FNN	0.570 (0.021)
AT-8	8	Kaggle	0.882 (0.021)
AT-8	8	Liar	0.187 (0.011)
AT-8	8	FNN	0.511 (0.023)

Table 5.15: *The results of AT-7 and AT-8, standard deviation in parentheses*

Accuracy Testing Assessment

In Table 5.16, the proposed model's highest achieved accuracy on the Kaggle dataset is compared with accuracies obtained by other state of the art models. The results obtained by the proposed model are overall competitive with most of the models, notably beating the scores of both Ruchansky et al. and Mandical et al. Conversely, the achieved accuracy was notably worse than that of FakeBERT [Kaliyar et al., 2021]. However, it should be noted that the authors of FakeBERT conducted testing and training solely on the Kaggle dataset, while the proposed model has been an attempt at generalization across several datasets. It was thus expected that the results obtained by the proposed model would be lower on each individual dataset.

Kaggle dataset

Model	**Accuracy**
Proposed model	0.917 (0.019)
Kaliyar et al. [2021]	0.989
O'Brien et al. [2018]	0.935 (0.002)
Liu and Wu [2018]	0.921
Ahmed et al. [2017]	0.920
Ruchansky et al. [2017]	0.892
Mandical et al. [2020]	0.870

Table 5.16: *Accuracy (STD) comparisons for the Kaggle dataset, standard deviation in parentheses (where available)*

In Table 5.17, the model's achieved accuracy on the Liar dataset is compared to the results obtained by several state of the art models. Several results obtained by Bhutani et al. are included in the table, as these represent state of the art results using various model approaches [Bhutani et al., 2019]. It should be noted that only results obtained using the base version of the dataset (without metadata like party affiliation, history and context) were considered, as this was the dataset used for testing and training the proposed model. Further, the results achieved by Ozbay and Alatas were not considered [Altunbey Ozbay and Alatas, 2019]. This is due to the authors not specifying how they treated the dataset. As this dataset consists of six different classes, differentiation between them is notoriously difficult [Wang, 2017]. Ozbay and Alatas state accuracy results of up to 0.965. This drastic improvement (when comparing to the accuracy scores in Table 5.17) suggests that Ozbay and Alatas altered the dataset, for instance converting the six classes into two and performing binary classification. These results are not comparable to those in Table 5.17 and are thus excluded.

Liar dataset

Model	**Accuracy**
Proposed model	0.227 (0.004)
Wang [2017]	0.270
Bhutani et al. [2019], with Random Forest	0.351
Bhutani et al. [2019], with CNN	0.248
Bhutani et al. [2019], with M-Naive Bayes	0.236
Rashkin et al. [2017]	0.220

Table 5.17: *Accuracy (STD) comparisons for the Liar dataset, standard deviation in parentheses (where available)*

The achieved accuracy on the Liar dataset is competitive with most state of the art results, although Bhutani et al. obtained better results with their Random Forest classifier. Notably, the proposed model beats the model of Rashkin et al., which many of the features in the proposed model were based on. This accomplishment is attributed to both the other proposed features and the effectiveness of using an AIS for the fake news classification.

When comparing the obtained results on the FakeNewsNet dataset to those of state of the art models, it was discovered that most researchers use another version of the dataset. As mentioned, the dataset is sampled from PolitiFact and GossipCop and in the dataset used for training and testing the proposed model, these samples were mixed together and not source-labelled. However, most papers use a split version of the dataset, where classification is conducted on the PolitiFact and GossipCop parts separately. This made the obtained accuracy results unfit for comparison. Nevertheless, the FakeNewsNet dataset served as a valuable evaluation tool for the feature testing phase.

Overall, the proposed model produced competitive results on both the Kaggle and the Liar datasets, beating multiple state of the art models. It should also be considered that only a fraction of the total amount of samples were used, for all the datasets. A larger number of samples might also help mitigate the problem of separating different-class antigens which lie close together in the feature space. This is due to the feature values being normalized. More dataset samples would thus help ”calibrate” the feature values through normalization, so that the values would be more representative of class labels. The fact that the model was trained and tested on three distinct datasets should also further strengthen the weighting of the obtained results, considering that most researchers only test their models on single datasets. Consequently, the obtained results indicate that there is considerable potential in using Artificial Immune Systems for fake news classification.

Chapter 6

Evaluation and Conclusion

This chapter aims to present an overall assessment and discussion of the proposed model and the obtained results. Further, the contributions to the research community are presented and possibilities for future work within the research field are identified.

6.1 Conclusion and Goal Evaluation

The overall goal of this thesis was to *investigate the applicability of an Artificial Immune System for the classification of fake news articles.* This goal was subsequently divided into three separate research questions, aimed at addressing three different sides of the overall goal. The findings obtained from testing the proposed model are subsequently discussed, in light of these research questions.

Research question 1 *How should the traditional design of an Artificial Immune system be adapted to enable fake news classification?*

The underlying AIS base of the proposed model has been designed from scratch, taking inspiration from several state of the art AIS models. Architectural design choices were continuously made to tailor the AIS to the task of fake news classification. One of these choices was the local feature selection, which allows the antibodies to discard features which aren't useful for classification. The feature selection approach was based on that of AISLFS[Dudek, 2012], but was altered in that mutation was conducted on the feature values and RR radii, in addition to the feature selection vector. This feature selection scheme makes the model robust to features of varying effectiveness and also helps mitigate the "curse of dimensionality". The result of these choices was an AIS that produces high accuracies on both benchmark classification datasets and fake news datasets.

The testing results indicate that the model has trouble evolving the antibody population in a way that increased classification accuracy over time, on the fake news datasets. As this was only a problem on the fake news datasets, this was likely related to the feature extraction and not the AIS architecture. In fact, the model achieved accuracies competitive with state of the art models, even with the problem of antibody evolution. This suggests that the decisions made for antibody initialization and class prediction indeed were suitable for the task of fake news classification.

It should also be noted that the parameter values were mostly selected on the basis of regular (non-fake news) classification. As such, more experimental parameter tuning specifically towards fake news and the extracted features might help mitigate the problem of antibody evolution. The sample sizes used for fake news testing is also a notable side to this. Contrarily to the models which the accuracy results were compared to, the proposed model only used a fraction of the available samples for all three fake news datasets (see Table 5.8). Increased sample counts might result in greater coverage of the feature space, as well as enhanced fine-grained distinction between antigens, due to more data being used for normalization of feature values. Further, it is unknown how the state of the art models would perform with a similar fraction of the available samples. Additional strategies which might mitigate the problem of accuracy improvement are presented in Section 6.3.

Research question 2 *Which feature extraction strategies are suitable for an AIS adopted for the classification of fake and real news articles?*

A significant part of this thesis has been the investigation into which feature extraction strategies may be suitable for an AIS. As such, several noteworthy findings has been identified in the feature and accuracy testing phases. Firstly, it was shown that the effectiveness of features vary greatly for different datasets. Features like the sentiment analysis using the first and last sentence of each news article proved very valuable for the Kaggle dataset, but not on the other datasets. Further, it was shown that the two novel features of calculating term frequencies of divisive topics and quotation marks proved highly useful, both being amongst the top 5 features (amongst 25 features in total), when averaged across all the datasets. This highlights the need for more research into sophisticated feature extraction strategies for fake news classification.

It was found that using only headlines for feature extraction was generally less effective than hypothesized. This was the case both for the sentiment analysis and the text embedding features, where the approaches which used the article corpora scored better than those using headlines. The fact that the features were tested on three distinct datasets provide additional weight to these findings. Most state of the art models test their models and features on single datasets, which provide little information related to generality. As the main goal of most fake news classification models is to formulate a general model capable of classifying news articles found "in the wild", such generality is important.

Research question 3 *How does the proposed model perform, in terms of accuracy, compared to other fake news classification methods?*

The accuracy results obtained by the proposed model were generally competitive with those obtained by most relevant models, although falling short of one model for each of the two datasets used for comparison ([Kaliyar et al., 2021] and [Bhutani et al., 2019]). However, the obtained accuracies on the Kaggle and Liar datasets exceed those of multiple state of the art models, including the model by Rashkin et al., which served as the main inspiration for the term frequency-based features [Rashkin et al., 2017]. These findings highlight the potential of using AISs for fake news classification.

6.2 Contributions

Throughout the implementation and testing of the proposed model, several valuable contributions to the field were made. The main contribution was the investigation into adapting an AIS for the task of fake news classification. As this was a completely novel application area for AISs and also a novel approach to fake news classification, there was significant uncertainty tied to both model design and classification effectiveness. The achieved results indicate that there is potential in this application, although more research should be conducted into feature extraction and architectural decisions for the AIS model.

Two novel features were proposed, including divisive topic and quotation mark term frequencies. These were shown to perform well across all three datasets tested, contrarily to the vast majority of features tested. These features may be used in future research into fake news classification, regardless of the base model being an AIS or not.

The feature testing phase considered a significant amount of features and evaluated these in an objective manner, across three distinct datasets. These evaluations may serve as an assessment of feature effectiveness, valuable for future research into feature extraction. Most existing papers only test their features on single datasets, which they often produce themselves. Thus, the findings in these papers regarding feature effectiveness are generally less reliable than those found in this thesis.

Finally, it was found that news article headlines are generally less representative of the nature of the article, than theorized. For the sentiment analysis, the *head + tail* approach worked better than that using the article headlines as input. Similarly, for the text embedding features, the *tail* approach worked significantly better than the headline approach, across all the datasets. It was also notable that the text embedding feature which used the *tail* approach was found to outperform all the other text embedding features, for all the datasets. These findings serve as contributions to the research area of fake news classification using sentiment mining and text embeddings.

6.3 Future Work

Due to the broadness of the topic of this thesis, several potential extensions had to be discarded due to time and resource limitations. These potential extensions may serve as inspiration for further research.

6.3.1 Increased Sample Sizes

A simple extension of the work conducted in this thesis would be to investigate the model's performance using larger sample sizes. The datasets used for testing contain thousands of article samples and the full potential of this data was not used in the proposed model – due to computational restrictions. Using larger sample sizes would likely enhance the classification accuracy of the model, but it's uncertain to which degree. However, larger sample sizes would also reduce the efficiency of the model. As such, a trade-off between performance and efficiency should likely be the goal of future research.

6.3.2 Alternative AIS Design Decisions

Throughout the design of the underlying AIS, several design decisions were made to accommodate an AIS which would work well for news article classification. However, several assumptions had to be made for this design process, due to the novelty of using an AIS for this purpose. As such, future work could investigate alternative architectural decisions for the AIS. Such alternative decisions could include investigating the impact of using crossover, using different antibody selection schemes, testing other fitness calculation methods or employing negative selection.

6.3.3 Alternative Features

The field of feature extraction for fake news classification is rapidly evolving and the results found in this thesis could be used to formulate more efficient features, in future works. Firstly, it would be interesting to investigate if separate *tail* and *head* approaches for sentiment analysis would yield similar results to those of the text embedding features, i.e. if the *tail* approach would outperform the other approaches for sentiment analysis as well. Further, the effects of using TF-IDF instead of TF could be investigated. This investigation could be combined with using TF-IDF to identify words which appear more often in fake news articles than real ones, instead of using lexicons. More extensive pre-processing of samples could also be implemented, for instance the removal of outlier samples before normalization.

The lexicons used did not exclusively contain words in their lemmatized form. Using a tool like Stanford CoreNLP to lemmatize the lexicon terms and then comparing them to the tokenenized and lemmatized news articles might boost the effectiveness of the features which use lexicons. However, it should be noted that the computational load of lemmatization is considerable.

Another feature idea which wasn't implemented was word frequency analysis of the news articles, to gain an estimate of the vocabulary of the author. This feature would not use a lexicon, but rather count the appearances of words in a given article corpus. If a relatively small subset of words are used very frequently, this might suggest that the vocabulary of the author is somewhat limited – further indicating that the news article is labelled as fake. Such a feature should be calculated on text corpus where stop words have been removed, as words like "a" and "the" would appear frequently for all news articles.

Finally, the semantic-based approach could be extended to include other approaches like source- and propagation-based strategies. Several datasets include meta-data which thus could be utilized by the model. During the feature formulation phase of the model, it was experimented with using the Google Fact Check API to aid classification, through sending GET requests. Although this feature was discarded due to being deemed as "cheating" and not being semantic based, it may serve as inspiration for further research into feature extraction.

6.3.4 Alternative BERT Models

The BERT model used to generate text embeddings was the pre-trained BERT-Base Cased model. Future work could include investigating the effects of using alternative models, like BERT-Large (although this would result in 1024-length output vectors, instead of 768). The effects of using uncased vs cased versions could also be investigated. Furthermore, a BERT model could be trained from scratch, instead of using a pre-trained version trained on Wikipedia corpus. By training the model on explicit real and fake news articles, the performance of text embedding features might be improved.

Bibliography

Abeel, T., de Peer, Y., and Saeys, Y. (2009). Java-ml: A machine learning library. *Journal of Machine Learning Research*, 10:931–934.

Ahmed, H., Traore, I., and Saad, S. (2017). Detection of online fake news using n-gram analysis and machine learning techniques. In Traore, I., Woungang, I., and Awad, A., editors, *Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*, pages 127–138, Cham. Springer International Publishing.

Alpaydin, E. (2010). *Introduction to Machine Learning.* The MIT Press, 2nd edition.

Altunbey Ozbay, F. and Alatas, B. (2019). A novel approach for detection of fake news on social media using metaheuristic optimization algorithms. *Elektronika ir Elektrotechnika*, 25(4):62–67.

Baug, E., Haddow, P., and Norstein, A. (2019). Maim: A novel hybrid bio-inspired algorithm for classification. In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1802–1809.

Berghel, H. (2017). *Lies, Damn Lies, and Fake News*, volume 50. IEEE.

Bernhard and Vygen (2008). *The Traveling Salesman Problem*, pages 527–562. Springer Berlin Heidelberg, Berlin, Heidelberg.

Bhutani, B., Rastogi, N., Sehgal, P., and Purwar, A. (2019). Fake news detection using sentiment analysis. In *2019 Twelfth International Conference on Contemporary Computing (IC3)*, pages 1–5.

Bozarth, L. and Budak, C. (2020). Toward a better performance evaluation framework for fake news classification. *Proceedings of the International AAAI Conference on Web and Social Media*, 14(1):60–71.

Burgoon, J., Blair, J. P., Qin, T., and Nunamaker, J. (2003). Detecting deception through linguistic analysis. In *Detecting Deception through Linguistic Analysis*, pages 91–101.

Choudhary, A. and Arora, A. (2021). Linguistic feature based learning model for fake news detection and classification. *Expert Systems with Applications*, 169:114171.

Collins, F. S. (2020). Mutation. *National Human Genome Research Institute.*

CTCA (2017). Cancer treatment centers of america. *Cancer Center.*

Darwin, C. (1859). *On the Origin of Species by Means of Natural Selection.* Murray, London. or the Preservation of Favored Races in the Struggle for Life.

De Castro, L. and Von Zuben, F. (2000). The clonal selection algorithm with engineering applications 1. *GECCO 2000.*

Del Ser, J., Osaba, E., Molina, D., Yang, X.-S., Salcedo-Sanz, S., Camacho, D., Das, S., Suganthan, P. N., Coello Coello, C. A., and Herrera, F. (2019). Bio-inspired computation: Where we stand and what's next. *Swarm and Evolutionary Computation*, 48:220 – 250.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Dudek, G. (2012). An artificial immune system for classification with local feature selection. *Evolutionary Computation, IEEE Transactions on*, 16:847–860.

Fang, Y. and Li, J. (2010). A review of tournament selection in genetic programming. In *International Symposium on Intelligence Computation*, pages 181–192.

Gravanis, G., Vakali, A., Diamantaras, K., and Karadais, P. (2019). Behind the cues: A benchmarking study for fake news detection. *Expert Systems with Applications*, 128:201 – 213.

Hart, E. (2005). Not all balls are round: An investigation of alternative recognition-region shapes. In Jacob, C., Pilat, M. L., Bentley, P. J., and Timmis, J. I., editors, *Artificial Immune Systems*, pages 29–42, Berlin, Heidelberg. Springer Berlin Heidelberg.

Hart, E. and Timmis, J. (2008). Application areas of ais: The past, the present and the future. *Applied Soft Computing*, 8(1):191 – 201.

Homayounfar, H. (2003). An advanced island based ga for optimization problems. *Proc. Int. DCDIS Conf. on Engineering Applications and Computational Algorithms, Guelph, ON, 2003.*

Horne, B. D. and Adali, S. (2017). This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, page 168â177, New York, NY, USA. Association for Computing Machinery.

Idris, I. and Selamat, A. (2011). Negative selection algorithm in artificial immune system for spam detection. In *2011 Malaysian Conference in Software Engineering*, pages 379–382.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *Statistical Learning*, pages 15–57. Springer New York, New York, NY.

Jamieson, K. and Cappella, J. (2008). *Echo Chamber: Rush Limbaugh and the Conservative Media Establishment.* JSTOR.

Ji, Z. and Dasgupta, D. (2004). Real-valued negative selection algorithm with variable-sized detectors. In Deb, K., editor, *Genetic and Evolutionary Computation – GECCO 2004*, pages 287–298, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ji, Z. and Dasgupta, D. (2007). Revisiting negative selection algorithms. *Evolutionary Computation*, 15(2):223–251. PMID: 17535140.

Kaliyar, R. K., Goswami, A., and Narang, P. (2021). Fakebert: Fake news detection in social media with a bert-based deep learning approach. *Multimedia tools and applications*, page 1â24.

Kar, A. K. (2016). Bio inspired computing â a review of algorithms and scope of applications. *Expert Systems with Applications*, 59:20 – 32.

Karpov, P., Squillero, G., and Tonda, A. (2018). Valis: an evolutionary classification algorithm. *Genetic Programming and Evolvable Machines*, pages 453–471.

Klyuev, V. (2018). Fake news filtering: Semantic approaches. In *2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, pages 9–15.

Laurentys, C., Ronacher, G., Palhares, R., and Caminhas, W. (2010). Design of an artificial immune system for fault detection: A negative selection approach. *Expert Systems with Applications*, 37(7):5507–5513.

Levine, M. D. (1969). Feature extraction: A survey. *Proceedings of the IEEE*, 57(8):1391–1407.

Liu, Y. and Wu, Y.-F. (2018). Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Mahmoud, T. and Mahfouz, A. (2012). Sms spam filtering technique based on artificial immune system. *International Journal of Computer Science Issues*, 9.

Mandical, R. R., Mamatha, N., Shivakumar, N., Monica, R., and Krishna, A. N. (2020). Identification of fake news using machine learning. In *2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, pages 1–6.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., and McClosky, D. (2014). The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60. The Association for Computer Linguistics.

Markowska-Kaczmar, U. and Kordas, B. (2008). Multi-class iteratively refined negative selection classifier. *Applied Soft Computing*, 8(2):972–984.

Meel, P. and Vishwakarma, D. (2019). Fake news, rumor, information pollution in social media and web: A contemporary survey of state-of-the-arts, challenges and opportunities. *Expert Systems with Applications*, page 26.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space.

Nakamura, K., Levy, S., and Wang, W. Y. (2020). r/fakeddit: A new multimodal benchmark dataset for fine-grained fake news detection.

O'Brien, N., Latessa, S., Evangelopoulos, G., and Boix, X. (2018). The language of fake news: Opening the black-box of deep learning based detectors. *Center for Brains, Minds and Machines (CBMM).*

Oda, T. and White, T. (2005). Immunity from spam: An analysis of an artificial immune system for junk email detection. In Jacob, C., Pilat, M. L., Bentley, P. J., and Timmis, J. I., editors, *Artificial Immune Systems*, pages 276–289, Berlin, Heidelberg. Springer Berlin Heidelberg.

Pennebaker, J., Francis, M., and Booth, R. (1999). Linguistic inquiry and word count (liwc). *Journal of Language and Social Psychology.*

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations.

Pocock, A. (2012). Javami toolbox. *Journal of Machine Learning Research (JMLR).*

Rashkin, H. J., Choi, E., Jang, J. Y., Volkova, S., and Choi, Y. (2017). Truth of varying shades: Analyzing language in fake news and political fact-checking. *Conference: Proceedings of the 2017 Conference on Empirical Methods for Natural Language Processing.*

Read, M., Andrews, P. S., and Timmis, J. (2012). *An Introduction to Artificial Immune Systems*, pages 1575–1597. Springer Berlin Heidelberg, Berlin, Heidelberg.

Ruchansky, N., Seo, S., and Liu, Y. (2017). *CSI: A Hybrid Deep Model for Fake News Detection*, page 797â806. Association for Computing Machinery, New York, NY, USA.

Saleh, A., Karim, A., Shanmugam, B., Azam, S., Kannoorpatti, K., Jonkman, M., and De Boer, F. (2019). An intelligent spam detection model based on artificial immune system. *Information (Basel)*, 10(6):1–17.

Secker, A., Freitas, A. A., and Timmis, J. (2003). Aisec: an artificial immune system for e-mail classification. In *The 2003 Congress on Evolutionary Computation, 2003. CEC '03.*, volume 1, pages 131–138 Vol.1.

Sharma, A. and Sharma, D. (2011). Clonal selection algorithm for classification. In Liò, P., Nicosia, G., and Stibor, T., editors, *Artificial Immune Systems*, pages 361–370, Berlin, Heidelberg. Springer Berlin Heidelberg.

Shu, K., Mahudeswaran, D., Wang, S., Lee, D., and Liu, H. (2018). Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media. *arXiv preprint arXiv:1809.01286.*

Singhania, S., Fernandez, N., and Rao, S. (2017). 3han: A deep neural network for fake news detection. In Liu, D., Xie, S., Li, Y., Zhao, D., and El-Alfy, E.-S. M., editors, *Neural Information Processing*, pages 572–581, Cham. Springer International Publishing.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, pages 1631–1642.

Sun, C., Qiu, X., Xu, Y., and Huang, X. (2019). How to fine-tune BERT for text classification? *CoRR*, abs/1905.05583.

Volkova, S., Shaffer, K., Jang, J. Y., and Hodas, N. (2017). Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on Twitter. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 647–653, Vancouver, Canada. Association for Computational Linguistics.

Vosoughi, S., Roy, D., and Aral, S. (2018). The spread of true and false news online. *Science*, 359(6380):1146–1151.

Wang, W. Y. (2017). "liar, liar pants on fire": A new benchmark dataset for fake news detection.

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

Xiao, H. (2018). bert-as-service. https://github.com/hanxiao/bert-as-service.

Zhou, X. and Zafarani, R. (2018). A survey of fake news: Fundamental theories, detection methods, and opportunities. *Fake News Analysis, Detection and Intervention on Social Media.*

Zhou, X. and Zafarani, R. (2019). Network-based fake news detection: A pattern-driven approach. *ACM SIGKDD Explorations Newsletter*, 21:48–60.

Appendices

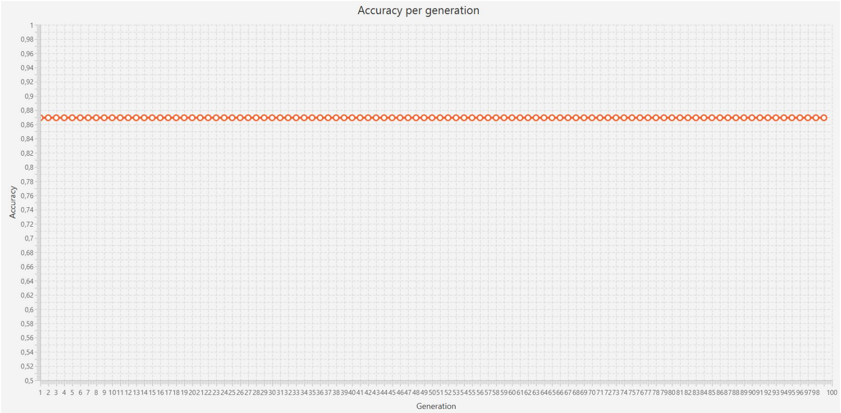


Figure 6.1: *Accuracy plot of AT-7 (Kaggle)*

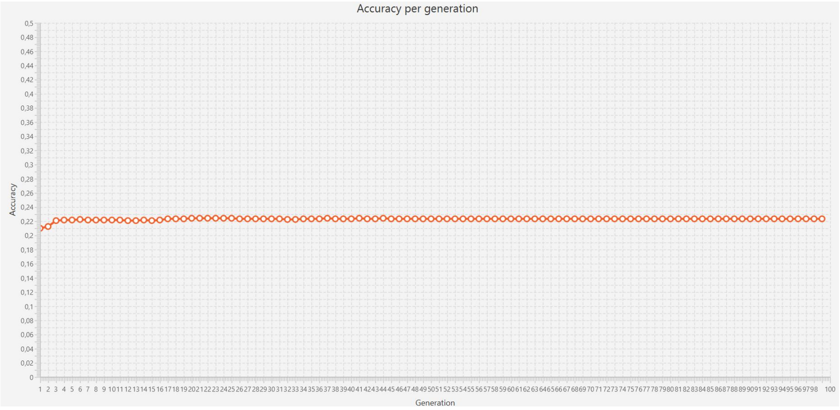


Figure 6.2: *Accuracy plot of AT-7 (Liar)*

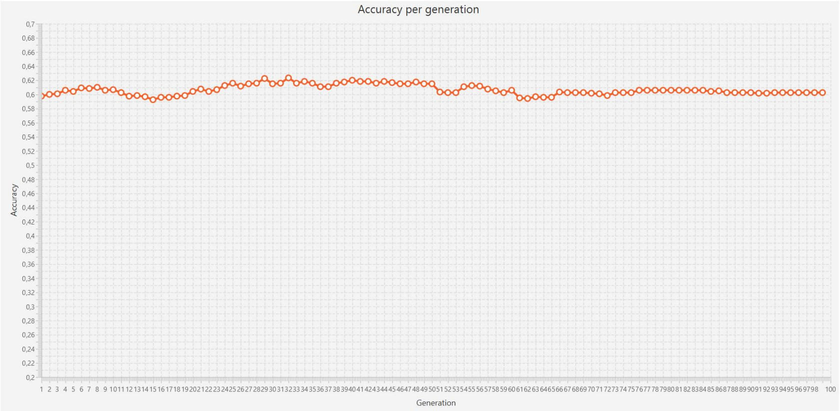


Figure 6.3: *Accuracy plot of AT-7 (FNN)*

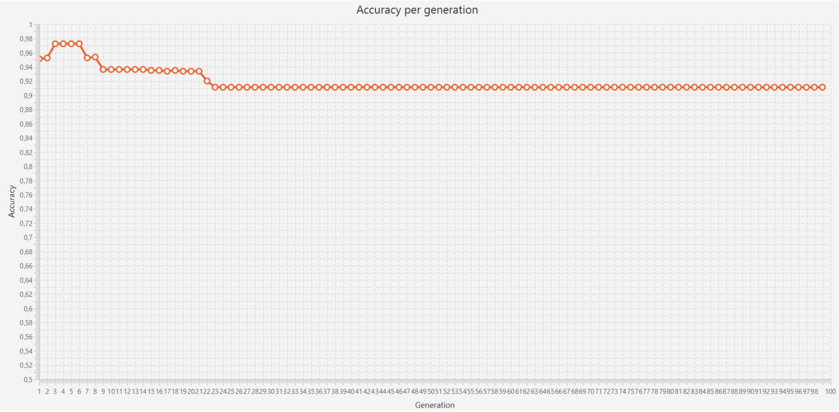


Figure 6.4: *Accuracy plot of AT-8 (Kaggle)*

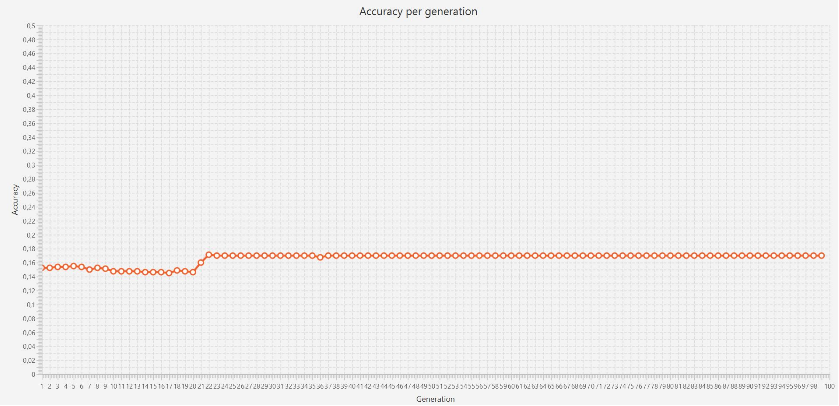


Figure 6.5: *Accuracy plot of AT-8 (Liar)*

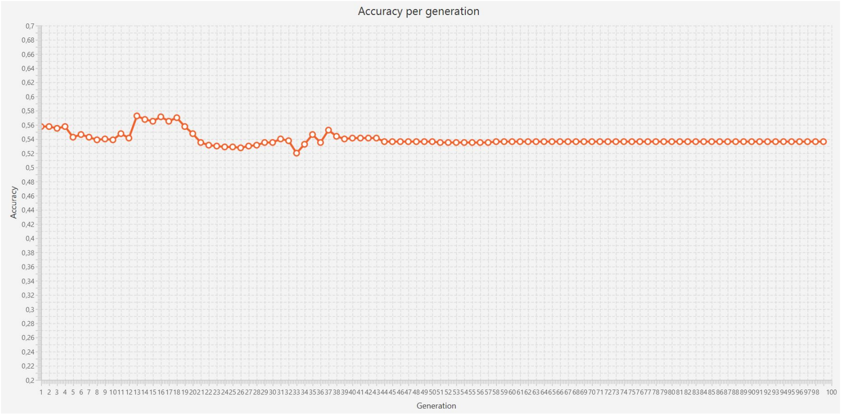


Figure 6.6: *Accuracy plot of AT-8 (FNN)*

Simen Sverdrup-Thygeson

An Artificial Immune System for Fake

NTNU

Norwegian University of
Science and Technology