

Nina Duong and Ola Johannessen Kruge

A Content-Based Artificial Immune System for Music Recommendation

Master's thesis in Master of Science in Informatics

Supervisor: Pauline Catriona Haddow

July 2022

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



Norwegian University of
Science and Technology

Nina Duong and Ola Johannessen Kruge

A Content-Based Artificial Immune System for Music Recommendation

Master's thesis in Master of Science in Informatics

Supervisor: Pauline Catriona Haddow

July 2022

Norwegian University of Science and Technology

Faculty of Information Technology and Electrical Engineering

Department of Computer Science



Norwegian University of
Science and Technology

Abstract

The automatic playlist continuation task involves suggesting appropriate songs to be added to a playlist. The qualities of a good playlist include diversity, common themes, and coherence, which require the playlist recommendations to also meet these qualities. Spotify made the challenge for RecSys 2018 regarding APCs, and additionally, it created a dataset with 1 million playlists that was related to the challenge. The majority of contestants developed hybrid or pure collaborative recommendation systems for the APC task, but content-based approaches seem to be less investigated. The reason for this may be that the features taken from the Spotify API are limited to nine features, making collaborative-filtering superior.

The thesis objective is to develop a content-based Artificial Immune Recognition System (AIRS) tailored to the APC task, named MAIRS - Music Artificial Immune Recognition System. The thesis presents its own similarity measure and classification method for the proposed model. There was an online evaluation of the proposed model, where 12 participants were asked to rate each song recommendation and compare them to Spotify's own recommendations. In the end, it appeared that the proposed model could not compete with Spotify on the basis of the online evaluation. In spite of this, MAIRS produced some positive results, and should be further investigated with the addition of collaborative filtering and more features.

Sammendrag

Automatisk spilleliste fortsettelse (ASF) innebærer å foreslå sanger som kunne passet inn i en original spilleliste. Diversitet, et gjennomgående tema og sammenheng mellom sangene er elementer som beskriver en god spilleliste. Rekommanderingssystemer innenfor musikk må derfor ta hensyn til disse elementene for å oppnå gode musikk forslag. I 2018 ble, RecSys, en konkurranse i regi av Spotify opprettet for å utfordre allmenheten i å lage egne modeller for å videre utforske forskjellige metoder å oppnå gode ASF'er på. I den sammenheng delte Spotify et datasett med 1 million bruker opprettede spillelister. Majoriteten av deltagerne i konkurransen bidro med enten hybride eller rene sammenhengsbaserte systemer, hvor rene innholdsbaserte systemer ble nedprioritert. Årsaken til dette kan ha vært det faktum at Spotify sitt API ikke inneholder mer enn ni lyd trekk til å beskrive hver sang, som gjør det enda mer gunstig å anvende sammenhengsbaserte systemer.

Avhandlingens mål er å utvikle et innholdsbasert kunstig immun system skreddersydd til ASF ved navn 'MAIRS - Music Artificial Immune Recognition System'. Avhandlingen kommer til å presentere egne likhets mål og klassifiserings metoder brukt i den foreslåtte modellen. Igjennom en undersøkelse med 12 stykker, blir modellen evaluert opp mot Spotify ved at hver bruker svarer på spørsmål i henhold til rekommenderinger fra både MAIRS og Spotify. Spørsmålene er rettet mot selv-lagde evalueringsmål med oppgave i å gi innsikt i hvor godt rekommending systemene gjør det. Resultatene fra brukerundersøkelsen viste til slutt at MAIRS ikke presterte like bra som Spotify. Til tross for dette produserte MAIRS positive resultater verdt å videre utforske, spesielt i kombinasjon med sammenhengsbaserte systemer og flere audio trekk til å beskrive sangene.

Preface

The following master thesis is the result of research conducted at the Norwegian University of Science and Technology in Trondheim, Norway, between 26.08.2021 and 11.07.2022. It would be our pleasure to thank our supervisor, Pauline Catriona Haddow, for her excellent guidance throughout the project. You and the rest of our team gave great feedback at the bio-inspired computing research group meetings, CRABLAB. Both the fun factor and quality of the project would have lowered without these meetings.

Nina Duong and Ola Johannessen Kruge
Trondheim, July 11, 2022

Contents

Abbreviations	1
1 Introduction	3
1.1 Motivation	3
1.2 Goals and Research Questions	4
1.3 Research Method	5
1.4 Structured Literature Review Protocol	6
1.4.1 Research questions	6
1.4.2 Research Strategy	6
1.5 Preliminary Process Overview	8
1.6 Thesis Structure	11
2 Background Theory	13
2.1 Evolutionary Algorithm	13
2.1.1 Representation	14
2.1.2 Population	14
2.1.3 Fitness function	15
2.1.4 Parent Selection	15
2.1.5 Recombination	15
2.1.6 Mutation	15
2.1.7 Survivor Selection (Replacement)	16
2.1.8 Niching	16
2.2 Immune system (IS)	17
2.3 Artificial Immune System (AIS)	18
2.4 Recommendation System	19
2.4.1 Collaborative Filtering	19
2.4.2 Content-based Filtering	20
2.5 Evaluation of recommendation systems	20
2.6 Levels of contents - Onion model	20

3	State of the Art	23
3.1	Music Recommendation	23
3.1.1	Cold start	23
3.1.2	Recommendation diversity, novelty and serendipity	24
3.1.3	Automatic Playlist Continuation (APC)	25
3.1.4	Data sets	26
3.1.5	Evaluation	30
3.2	Artificial Immune Systems	32
3.2.1	AIS algorithms	32
3.2.2	Clonal selection	33
3.2.3	Initialisation	35
3.2.4	Diversity	37
3.2.5	Classification	39
4	Model and Architecture	41
4.1	Dataset	41
4.2	Model Architecture	44
4.2.1	Model Structure	44
4.2.2	Chromosome representation	44
4.2.3	Affinity, Affinity Threshold, Stimulation Calculation	45
4.2.4	Proposed model flow chart	47
4.2.5	Model Parameters	49
4.2.6	Initialisation	49
4.2.7	Memory cell identification	50
4.2.8	ARB generation	51
4.2.9	Train ARB	52
4.2.10	Memory cell introduction	54
4.2.11	Generation of Recommended Song Set (RSS)	55
4.3	Online evaluation questionnaire	60
5	Experiments and Results	63
5.1	Preliminary tests	63
5.1.1	Effects of KNN as similarity measure	63
5.1.2	Accuracy measurement of the RSS	64
5.1.3	Audio features and playlist type	65
5.2	Visualisation tools	65
5.3	Experimental Plan	65
5.3.1	Overview of experiment plan	66
5.4	Experimental Setup	67
5.4.1	Exp.1	68
5.4.2	Exp.2	68

5.4.3	Exp.3	69
5.4.4	Exp.4	69
5.5	Experimental Results	70
5.5.1	Exp.1	70
5.5.2	Exp.2	76
5.5.3	Exp.3	81
5.5.4	Exp.4	85
6	Evaluation and Conclusion	91
6.1	Discussion and goal evaluation	91
6.2	Limitations	94
6.2.1	Spotify rate limiting	94
6.2.2	Evaluation metrics	94
6.3	Contributions	95
6.4	Future Work	96
6.4.1	Offline evaluation and optimisation	96
6.4.2	Audio feature expansion	96
6.4.3	Parameter tuning to music	96
6.4.4	Investigation of other design decisions	97
	Bibliography	99
	Appendices	105

List of Figures

1.1	Initial topic investigation	8
1.2	Overview of further topic investigation	10
2.1	The generation cycle of the Evolutionary Algorithm	14
2.2	Crossover and mutation operators	16
2.3	Onion model and level of contents	21
3.1	Variable-sized detectors and detection of non-self space	35
3.2	Recognition shape and classification accuracy	36
4.1	MPD data structure	42
4.2	MAIRS 1.0 - Model overview	47
4.3	MAIRS 2.0 - Model overview	47
4.4	MARIS 2.0 and feature space exploration	51
4.5	The flow chart of process Train ARB	52
4.6	The flow chart of generation of RSS	55
4.7	Closest affinity	56
4.8	Average affinity	57
4.9	Range method	58
4.10	Recognition region	59
5.1	Exp.4 overview	69
5.2	Exp.1 playlist independent results	71
5.3	Exp.1 playlist 1017 results	72
5.4	Exp.1 playlist 589 results	72
5.5	Exp.1 playlist 89 results	73
5.6	Exp.1 playlist dependent comparison of similarity measures	74
5.7	Exp.2 average score MAIRS 1.0 and Range method	76
5.8	Exp.2 average score of MAIRS 1.0 and RR 0.2	77
5.9	Exp.2 method comparison playlist 1017	78

5.10	MAIRS 1.0 Song Recommendations - Playlist 1017	79
5.11	Range Method Song Recommendations - Playlist 1017	79
5.12	Exp.2 method comparison playlist 589	80
5.13	Exp.3 method comparison MAIRS 1.0 and MAIRS 2.0	81
5.14	Exp.3 method comparison playlist 1017 MAIRS 1.0, Range method and MAIRS 2.0	82
5.15	Exp.3 method comparison playlist 589 MAIRS 1.0, Range method and MAIRS 2.0	82
5.16	Exp.3 method comparison playlist 89 MAIRS 1.0, Range method and MAIRS 2.0	83
5.17	Exp.4 playlist dependent model comparison MAIRS 2.0 and Spotify	85
5.18	Exp.4 liking to diversity ratio model comparison MAIRS 2.0 and Spotify	86
5.19	Exp.4 playlist independent model comparison MAIRS 2.0 and Spotify	88
5.20	Exp.4 user preference model comparison MAIRS 2.0 and Spotify .	89

List of Tables

1.1	Research guidelines	7
3.1	Comparison of the different datasets	28
3.2	Strength and weakness of proposed datasets	28
4.1	Spotify API Audio feature description	43
4.2	MAIRS parameter explanation	49
4.3	Online evaluation questionnaire	60
5.1	Experiment plan overview	66
5.2	Methods compared in the experiments	67
5.3	Default parameters in all experiments	67
5.4	Exp.2 parameters	68
5.5	Exp.3 parameters	69

Abbreviations

AA	= Average affinity
AIS	= Artificial immune system
APC	= Automatic playlist continuation
CA	= Closest affinity
CBF	= Content based filtering
CF	= Collaborative filtering
EA	= Evolutionary algorithm
KNN	= K-nearest Neighbour
MAIRS	= Music Artificial Immune Recognition System
MPD	= Million Playlist Dataset
MRS	= Music recommendation systems
RM	= Range method
RS	= Recommendation system
RSS	= Recommended song set
RR	= Recognition region

Chapter 1

Introduction

The motivation and research goal for a novel bio-inspired algorithm for automatic playlist continuation: *MAIRS - Music Artificial Immune Recognition System* is presented in section 1.1 and 1.2. Further, in section 1.3 a short explanation of the research method is presented, then the structured literature review protocol is shown in section 1.4. The chapter ends with a presentation of the preliminary process overview and thesis structure in section 1.5 and section 1.6, respectively.

1.1 Motivation

In a digital age, where users are faced with a significant amount of data, the recommendation system has become an essential part of everyday life to assist users in selecting products and services that are suitable for their needs. The music industry in particular has seen a rapid growth in the recent years, with users now able to access a greater variety of content. The need for user-tailored recommendations is, therefore, a necessity. Spotify, for instance, has changed the way music is recommended based on a user's preferences, moods, and emotions, presenting new opportunities for industry players and listeners alike. The streaming service has over 140 million active users and is considered one of the leading streaming services today.

In 2018, Spotify hosted a challenge in collaboration with MIT and Johannes Kepler University at the Recommender Systems Conference. The challenge focused on music recommendation, particularly automatic playlist continuation, which involved suggesting appropriate songs to be added to a playlist. This allows the Recommender System to increase user engagement by expanding listening beyond the end of existing playlists and making playlist creation easier. Likewise,

the recommendation system must maintain a balance between offering the user familiar items whilst expanding their taste by recommending items that differ from what the user is familiar with. The combination of familiarity and novelty makes the task of providing recommendations tailored to the needs of individual users more complex than traditional classification.

Most of the approaches exhibited in this challenge involved traditional neural networks with collaborative filtering. The music recommendation system is, however, not much explored in the field of bio-inspired AI. A paper by Dionisios N. Sotiropoulos showed, however, promising results when using an artificial immune system with negative selection to create an accurate model of a user's music taste's negative space. The reason for this is that users have difficulty articulating the music they do not like while their taste in music is readily evident in their playlist. Therefore, determining users' music preferences is almost solely based on positive training samples. Finally, by the use of clonal selection, this paper proposes an AIS-based model for predicting the positive space of a user's music preferences.

1.2 Goals and Research Questions

This section outlines the goal of this thesis and the research questions that will be investigated. The goal serves as a statement of the thesis's objective. To assess the overall goal's achievement, the research questions divide the overall goal into smaller, more manageable sub-goals.

Goal *To investigate the applicability of an Artificial Immune System (AIS) with content-based filtering (CBF) for Automatic Playlist Continuation (APC) task with limited features*

A novel approach to Automatic Playlist Continuation is to use AIS in combination with content-based filtering to solve the problem. In terms of APC, content-based filtering is also less investigated than collaborative filtering. Therefore, the thesis aims to develop a model that is on par with other state-of-the-art models using AIS and content-based filtering as a base. In addition, learn and investigate different results from experiments and propose, implement, and revise model components.

Hypothesis *Similar songs appears in the same playlist*

Songs in the same playlist tend to share common characteristics, such as musical style or musical genre. Thus, it is reasonable to assume that songs that are close to one another in the search space are also likely to share similar traits

which could reflect these common music genres or styles. The songs must exhibit coherence and share a common theme when listened to, in order to be considered similar.

Research question 1 *What similarity measure should be applied to ensure that songs in a playlist can be classified as 'similar' despite a limited number of features in the representation?*

The first research question explores the different similarity measures between songs from an original playlist and the dataset. An original playlist, in the broadest definition, is a collection of songs created by a user, intended to listen to together. While a dataset is a collection of x amount of songs, minus the songs in the original playlist.

Research question 2 *How should AIRS be refined for content-based filtering of Automatic Playlist Continuation while taking music diversity into account?*

Research question 3 *How can similarity be encouraged while maintaining diversity in the proposed model?*

Research question 3 is built upon RQ2. While RQ2 focuses on the development of AIS and CBF with diversity, this question seeks to restore the similarity that may have been lost.

Research question 4 *How can the AIS model ensure similarity whilst achieving music diversity in the recommendations?*

1.3 Research Method

In the initial phase of the research, a literature review was conducted on music recommendation systems and artificial immune systems, both separately and in combination. The structured literature review provided the basis of the goal and research questions. The strengths of both the fields and similar models were further analysed as well as music evaluation metrics and datasets. The knowledge from the literature research laid the foundations for designing the proposed model. Key findings of the structured literature review is presented in section 3.

After completing the design of the proposed model, the implementation process began. A decision was made on which programming language(s) to use and the implementation of a framework that made it easy to both evaluate the results and test the effects of different parameters. The final phases of the project revolved around structurally conducting the tests and evaluating the results according to

the experiment plan. The results from the experiment plan were evaluated in terms of how they answered the research questions, which contributions were made to the field and how the model could be further developed.

1.4 Structured Literature Review Protocol

1.4.1 Research questions

The literature search revolved around gathering information in order to get the best foundation for fulfilling the research goal. In the course of the literature review, new insights made it necessary to adjust the research goal. Every adjustment narrowed down the literature search and made the searches more specific. However, in the initial phase of the project, the literature review revolved around exploring and trying to answer the following questions to their best:

- Which methods exist within the world of AIS and what are their strengths and weaknesses?
- Are there any established ways of representing music when building a recommendation system?
- Do any of the representations fit AIS models?
- What characterizes a successful music recommendation system? And are there any established music recommendation methods?
- Which datasets are freely available, easy to use, and consist of enough music features?

1.4.2 Research Strategy

To ensure that the literature search was comprehensive, it was essential to examine an extensive number of relevant articles. In that regard, Google Scholar was a logical starting point since it includes results from all of the different publishing platforms. Additionally, guidelines were followed to ensure the quality of the literature used to answer the research questions. These are summarized in the following table 1.1.

Publishing platforms:
<ul style="list-style-type: none"> • IEEE Explore, ScienceDirect, ResearchGate, SpringerLink
Important keywords:
<p>AIS related:</p> <ul style="list-style-type: none"> • Artificial Immune System, Clonal Selection, Negative Selection, Recommendation System AIS. <p>Music recommendation related:</p> <ul style="list-style-type: none"> • Recommender systems, Recommender systems survey, Evaluation metrics music, Automatic music playlist, Music recommendation benchmark. <p>Dataset related:</p> <ul style="list-style-type: none"> • Dataset music recommendation, Dataset music survey, Dataset music.
Qualifying (worth taking a look at):
<ul style="list-style-type: none"> • The content of the abstract seems relevant to the search query. • The article is written by a recognized person within the field. • The conclusion section is clear and the results seem relevant to the research questions.
Evaluating (at least one of the five):
<ul style="list-style-type: none"> • The article is presented in a recognized journal. • The article has a substantial amount of references to other papers. • The research is compared to recognized works within the field. • The citation count is high in relation to the date of publication. • The article is highly relevant to the research goal.

Table 1.1: Research guidelines

As well as carefully selecting keywords to obtain the most relevant papers, the title of an article played a significant role in deciding whether the article is clicked on or not. Upon identifying a title that appeared relevant, a short time was spent reviewing the abstract and conclusion in order to decide the article's relevance and whether it was worth reading more closely. As a final consideration when evaluating whether an article could be included in the report, it was necessary to satisfy one of five criteria. It is important to note that the inclusion criteria differed from the areas examined in the study. Due to the size of the AIS and music fields, it was possible to be somewhat more picky. However, the combined size of the two was significantly smaller and the inclusion criteria had to be broadened.

1.5 Preliminary Process Overview

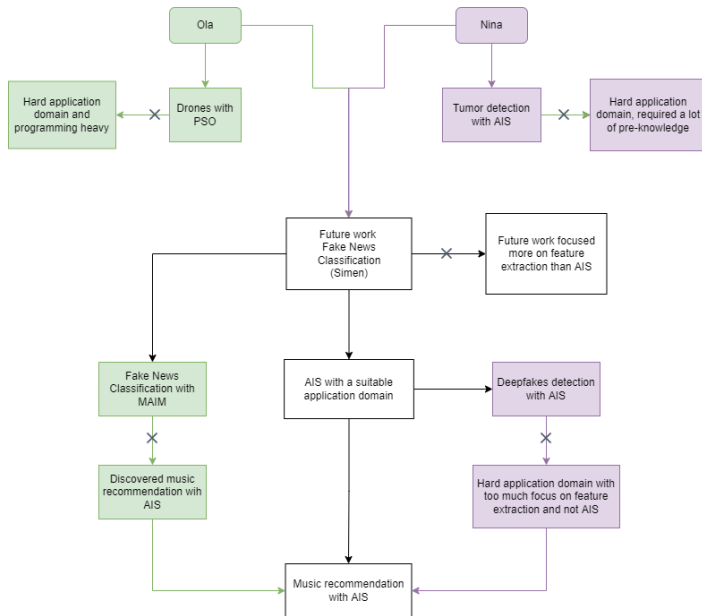


Figure 1.1: Initial topic investigation

Figure 1.1 illustrates an overview of the exploration of different projects in the initial phases of the project. The topic could be selected freely provided that it included evolutionary algorithm and contributed to the field. Due to this

freedom, several different application domains and research areas were explored before selecting the research topic. As the authors of this project have worked independently during the pre-project phase, their differing backgrounds have contributed to the selection of a common solution. The authors were both interested in working on a bio-inspired algorithm, however, it was unclear what type of application domain and what specific bio-inspired method would serve as the foundation of the model. As seen in figure 1.1 investigation of drones combined with particle swarm optimization and tumour prediction was conducted separately in the initial phases of the pre-project. Yet they were eventually discarded due to the extensive programming work required to integrate bio-inspired methods with the domains. Eventually, both authors discovered an article about fake news classification, [53] which in the end served as an inspiration for the combination of AIS and classification tasks. This discovery, led to more concrete literature research on AIS and classification tasks. The research was still done separately at this stage, and to the frustration of both, it was hard to find a novel yet interesting application domain that could suit AIS with classification. But as a result of gaining better knowledge of AIS and its applicability in different domains, the idea of music classification with AIS came to mind. Considering only a few articles were found on the combination of AIS and music classification, with promising results, there was a clear incentive to investigate the idea further. It was eventually decided that AIS and music recommendation would be the topic for the thesis. Due to both authors' interest in the topic and their familiarity with the AIS domain, it was evident that collaboration would be beneficial.

Following the selection of a topic, there was still a great amount of work to be done, and the following months were spent developing knowledge of recommendation systems and the different established methods for integrating music into recommendation systems. In terms of results, a combination of collaborative- and content based-filtering was considered the best [14]. First, it was necessary to determine whether these methods were compatible with AIS. Both filtering methods have been implemented previously with AIS, but only content-based filtering have been combined with AIS in the domain of music.

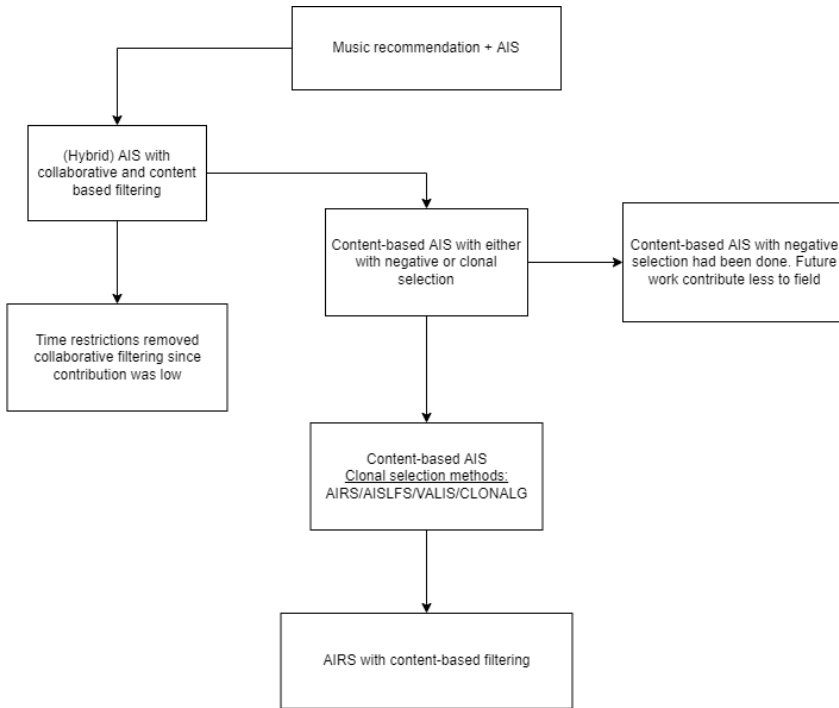


Figure 1.2: Overview of further topic investigation

After deciding to not include collaborative filtering in the model, the following design decision was AIS related. Either negative selection or clonal selection had to serve as the foundation for the AIS part of the model. In the initial literature search of music recommendation systems, it was found that diversity among song recommendations was an essential component of a successful recommendation algorithm. It was found that using clonal selection meant a greater amount of control over the level of diversity in the algorithm, which could be beneficial if the music diversity was too low or too high. Considering it also already existed research on negative selection and music recommendation, employing clonal selection with music would make this a novel approach. Having obtained this insight, the research goal of "To investigate the applicability of an Artificial Immune System (AIS) with content-based filtering (CBF) for Automatic Playlist Continuation (APC) task with limited features" was finally established.

1.6 Thesis Structure

The subsequent chapters will be presented as follows. Chapter 2 is devoted to introducing the reader to basic concepts and theories required to understand before reading the state of the art. Further, chapter 3 seeks to present the most relevant and established techniques within the field of music recommendation and AIS. The topic of chapter 4 concerns the model design decisions in light of the research presented in chapter 3. Chapter 5 contains the experiment plan, its results and an evaluation of them. Chapter 6 presents a discussion of the results, evaluates them against the research objective, and finally presents the future works of the project. On a final note, it should be mentioned that some of the content in chapter 2 and 3 is adapted from the research project conducted during the 2021 Fall semester.

Chapter 2

Background Theory

This chapter presents the background theory needed to understand before examining the proposed model. Starting with evolutionary algorithms in section 2.1, followed by immune system and artificial immune systems in section 2.2 and 2.3. Then theory related to recommendation systems and music is presented in section 2.4, 2.5 and 2.6.

2.1 Evolutionary Algorithm

An evolutionary algorithm (EA) is a type of search algorithm that uses strategies from nature to overcome different optimisation problems. The algorithms borrow key concepts from evolution in order to create possible solutions to the problem. Normally, this process is done over several generations (iterations), with each generation bringing the algorithm closer to a viable solution. The flowchart in figure 2.1 illustrates the general scheme of an evolutionary algorithm. Evolutionary algorithms always include a few essential components, which are described in the following paragraphs.

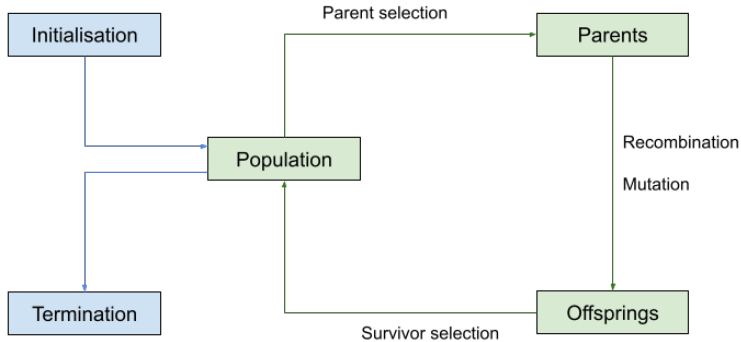


Figure 2.1: The generation cycle of the Evolutionary Algorithm, adapted from [17]

2.1.1 Representation

The first step in defining an EA is establishment of the individuals in a population. The objects that form possible solutions within the original problem context are called phenotypes, while their encoding, namely the individuals within the EA, are called genotypes. This first design step is also called representation, since it specifies how phenotypes are mapped onto a set of genotypes that are said to represent them. The representation differ between algorithms and are often tailored to the application domain. A common approach is to represent each individual as a bit string or as an vector of features.

2.1.2 Population

The role of the population is to hold (the representation of) possible solutions, which form the unit of evolution. The initial population is usually seeded by randomly generated individuals. However, other problem-specific heuristics could also be applied to the initialisation phase to produce an initial population with a higher fitness. It is not the individuals of a population that change or adapt; it is the population as a whole that does so. In nearly all evolutionary applications, the increase of population size creates competition among the limited resources.

Population diversity is determined by the number of different solutions in a population. There is, however, no single measure of diversity. Typically measured by the number of different fitness values, phenotypes or genotypes in the population. Entropy and other statistical measures are also used.

2.1.3 Fitness function

Defining the fitness function is often seen as the hardest part of creating an evolutionary algorithm. A poorly designed fitness function may either converge on an inappropriate solution or have difficulty converging at all. The goal of the fitness function is to represent the requirements the population should adapt to meet, and determine how "good" the solution is with respect to the problem in consideration. It forms the basis for selection, and so it facilitates improvements i.e. it defines what improvement means. As evolutionary algorithms can be applied to a wide range of different problems, the fitness functions can vary depending on the application domain.

2.1.4 Parent Selection

The purpose of parental selection is to distinguish among individuals based on their quality to potentially create offsprings of higher quality through crossover or mutation. Thus, in conjunction with survivor selection, parent selection is responsible for pushing quality improvements. However, choosing high-quality individuals over those of lower quality might make the EA too greedy, leading to a local optimum that forces the population convergence prematurely. To prevent this, tournament selections and fitness-based selections are often used as parental selection strategies to include a wider range of parents of different quality.

2.1.5 Recombination

Recombination (also known as crossover) is an variation operator that combines information from two parent genotypes into one or more offspring genotypes. This is decided by what parts of each parent are combined and how probabilistic the operator should be. By pairing two individuals with different yet desirable characteristics, the EA is able to produce an offspring that complements those traits. While some will have undesirable combinations of traits, and most be no better or worse than their parents, some will have improved characteristics. The crossover process can take many forms, such as one-point crossover, k-point crossover, and uniform crossover.

2.1.6 Mutation

Mutation is another variation operator, where different values in the genotype of an individual are changed based on a random probability. The mutations are intended to introduce diversity into the population. Often used in order to prevent chromosomes from becoming too similar to each other, thus slowing or even stopping convergence before reaching the global optimum. Unless the

mutation probability is too high, there may be some similarities between the offspring and its parent.

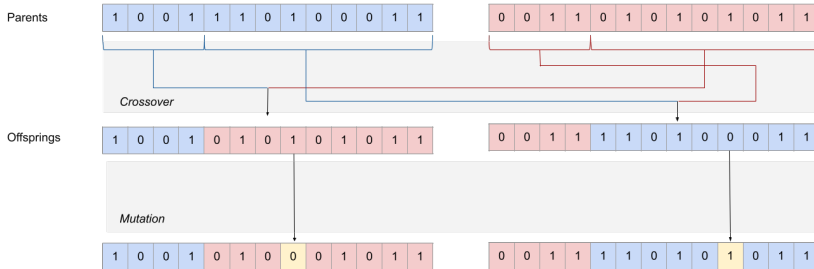


Figure 2.2: Crossover and mutation operators

2.1.7 Survivor Selection (Replacement)

The survivor selection policy of an evolutionary algorithm determines which individuals will be allowed into the next generation. In contrast to parent selection, which is typically stochastic, survivor selection is often deterministic. The decision is often based on their fitness values, favouring those with higher quality. In some cases, the age of individuals is taken into consideration. The importance of this step is that it needs to strike a balance between selecting the fittest individuals (elitism), while maintaining diversity in the population. Thus, the two standard methods include the fitness-based method of selecting the top segment from the whole population and the age-biased method of selecting only from the offspring. Tournament selection is also a possible strategy to maintain this balance.

2.1.8 Niching

Niching method is a set of methods that attempts to find more than one solution during a single search. It allows individuals to survive in separate pockets of the search space. The concept of niching is driven by multi-modal problems, where either individuals must be kept near several local optimums or multiple good solutions must be found. As a consequence, it promotes more diversity in the population preventing the population being stuck in a local optimum causing premature convergence. The three most common niching strategies are fitness sharing, clearing, and crowding. *Fitness sharing* decreases the fitnesses of individuals that share the same niche (close to each other in the search space) [44],

and is especially useful for multimodal problems. *Clearing* is similar to fitness sharing, but instead of sharing fitness values between individuals that share the same niche, the fitness of some of those individuals are decreased [39] [44]. *Crowding* is when similar individuals in the population are replaced by those recently created through recombination [12]. There are three types of crowding: standard crowding, deterministic crowding, and restricted tournament crowding.

2.2 Immune system (IS)

The biological immune system consists of immune cells that circulate around the body through the blood and lymph, forming a network to protect against infectious agents (*pathogens*) [23]. The immune system is classified into two components: innate and adaptive. The first ones are the body's first line of defence and provide a general guard against infection, virus, and bacteria. If the pathogens manage to bypass the innate immune system, the adaptive immunity is activated. The immune system begins to recognise a pathogen as "non-self" and distinguishes these from "self", which are cells that belongs to the body. To fight the pathogens, the immune system use cellular and chemical defences to attack. The adaptive system has a *memory mechanism* that learns to recognise the shape of the unseen pathogen, allowing for a faster response if the same shape re-appear [41].

Specific immune responses are triggered by *antigens*. Antigens are found on the surface of pathogens and are unique to that specific pathogen. The immune system responds to antigens by producing *lymphocytes* known as B- and T- cells. The B cell releases Y-shaped proteins called *antibodies* that bind to the antigens. The binding depends on a chemical structure and charge, and the likelihood of a bond occurring is called *affinity*. This binding *activates* the cell, and triggers the *clonal selection* process. In the clonal selection, the cells are cloned proportional to the affinity and mutated inversely proportional to the affinity. Selection pressure is achieved, which implies the cells with higher affinity survives [10].

Antibodies alone are often not sufficient to protect the body against pathogens. In these instances, the immune system uses T cells to destroy infected body cells. The T cells are divided into killer and helper T cells. The killer T cells assist with the elimination of infected body cells by releasing toxins into them and promoting cell death. Helper T cells act to activate other immune cells like the B cell. T and B cells are produced in the bone marrow, before maturing in the thymus. This maturing process is known as *negative selection*, a process where T cells that bind to self-antigens are eliminated [31]. This process protects the body against T cells that encourage attacks on the self-cells, also known as autoimmune diseases.

2.3 Artificial Immune System (AIS)

Artificial Immune Systems (AIS) is a branch of biologically inspired computation which incorporates characteristics from the natural immune systems, including diversity, distributed computation, error tolerance, dynamic learning and adaptation, and self-monitoring. Despite AIS possessing common concepts and processes with evolutionary methods, it exhibits peculiarities that separates them as it's own type. The application areas of artificial immune system are various and include, but not restricted to, learning, anomaly detection and optimisation. The most important types of AIS, in terms of classification, are based on the concepts of negative selection and clonal selection.

AIS generally follows the notion of shape space, where the antibodies and antigens exist as points in a shape space S . Antigens are instances from the dataset, and is represented as a vector of parameters $x = \langle x_1, x_2, \dots, x_n \rangle$, referred as the generalised shape of the antigen that belongs to space S . The antibody, which is the detector, is formed after the antigen's representation, attached with the class (to predict the antigen) and a recognition radius. The parameters that define the generalised shape vary according to the kind of problem adopted. It also determines the complexity of the antibody representation and, subsequently, its recognition shape, which in turn is highly dependent on the AIS model employed. The affinity is the measure of the distance between the antibody and antigen in the shape space and dictate how similar they are. The antibody binds to (classify) all the antigens that have an affinity above the defined threshold. This region of space is called the recognition space. The recognition radius determines the specificity of the antibody.

The union of the recognition region of all the antibodies are termed immune repertoire. Ideally, the immune repertoire should cover all the regions of space that do not correspond to autoantigens. To avoid holes in the shape space, each antibody can be given a different radius and affinity measures.

Curse of dimensionality

The number of features that represent antigens and antibodies has a tendency to create a high dimensionality in the shape space. This phenomenon is also known as the curse of dimensionality and in short, the curse makes spherical RR's decreasingly effective when the model uses datasets containing a high number of features [34]. With increasing dimensions, the volume of the recognition region is progressively reduced.

2.4 Recommendation System

A recommendation system (RS) aims to suggest the most relevant items to users, predicting their interest in an item based on related information about the items, the users, and their interactions [2]. An item can range from specific products from Amazon to personalised services such as music and movies. RS is primarily directed towards individuals who lack sufficient personal experience or competence to evaluate the potentially overwhelming number of alternative items a system offers [24]. Even though RS enhances the users' experience, the main goal is to increase sales of products and profit. That is why the common technical goals of recommendation systems are as follows [2]:

1. **Relevance (Similarity):** The most obvious goal of an RS is to recommend relevant items to the user at hand because users are more likely to consume items they find interesting.
2. **Novelty:** RS should recommend items the user has not seen before. Recommending popular items repeatedly can lead to a reduction in sales diversity.
3. **Serendipity:** Serendipity differs from novelty in that the user discovers unexpected and surprising recommendations. This way, the user would not lose interest by getting similar items and helps to expand the range of interest. On the other hand, providing serendipitous recommendations often recommends irrelevant items.
4. **Diversity:** When all the recommended items are similar, the chance of the user not liking any of them increases. With a more diverse recommendation, there is a higher chance that the user does not get bored by the items and might like at least one of these.

The majority of recommendation systems use a hybrid approach that combines collaborative filtering and content-based filtering, see 2.4.1, and possibly other approaches. These are either implemented separately and then combined, unified into one model or the capabilities of one of the approaches is added onto the other.

2.4.1 Collaborative Filtering

A well-known approach to recommendation system design is collaborative filtering (CF). Collaborative filtering is based on the assumption that users who share similar interests will like the same items. CF is divided into user-based and item-based approaches. In the user-based CF approach, a user will receive recommendations of items liked by similar users. In the item-based CF approach, a user will receive recommendations of items that are similar to those they have liked in the past.

2.4.2 Content-based Filtering

Another common recommendation technique is content-based filtering (CBF), which base the recommendations on the description of an item and the user's preferences. The CBF starts by analysing the features of items preferred by a particular user to determine preferences that can distinguish these items. These preferences are then used to find other items with a high degree of similarity. The similarity can be calculated by using traditional measures such as cosine similarity, or use statistical learning and machine learning methods to learn users' interests from the historical data. In this way, CBF can be seen as a user-specific classification where the classifier learns the user's likes and dislikes based on an item's features.

2.5 Evaluation of recommendation systems

The evaluation of a recommendation system determines the algorithm's quality and makes it possible to compare different systems. Evaluating a RS is a complex task and includes more criteria than simply measuring the rate of predicting items originally in the set of items (accuracy). A variety of criteria like coverage, novelty, serendipity, stability, diversity, and scalability goes into the equation that eventually evaluates the performance of the algorithm. Excluding several of these parameters might lead to either underestimating or overestimating the algorithm.

Evaluation of a RS is mainly divided into two primary types of evaluation: online and offline evaluations. Online evaluations differ from offline evaluations as they include users, making it possible to monitor the user's behavior to evaluate the system. The conversion rate of the user's clicks is hence an fundamental metric in online evaluation. Either way, offline evaluations of a system are the most common way to evaluate RS. By using standardised frameworks and identical evaluating measures, the foundation for comparing different algorithms becomes consistent. The main disadvantage with offline evaluations is the lack of ability to validate the system's performance over time in an evolving environment where the data and the user's behavior might change. It is important to include other metrics besides accuracy to avoid this pitfall. Measuring the degree of novelty and diversity has been shown to be crucial for user satisfaction and are secondary metrics that should be considered besides accuracy when evaluating RS.

2.6 Levels of contents - Onion model

Features in music recommendation can contain a variety of information, ranging from metadata (genre) from user-generated content (reviews) to audio features or

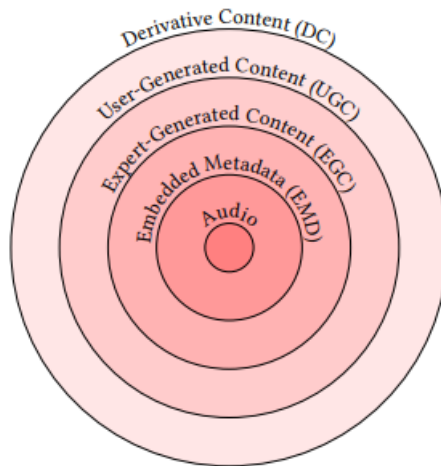


Figure 2.3: The "onion model" visualises the levels of content in the music domain (adapted from Deldjoo et al. [14])

semantic knowledge. Which features to include is a crucial factor for the music recommendation system. A survey by Deldjoo et al. [14] proposed the "onion model" which is a hierarchical model that describes the different levels of content in music recommendation. The onion model consists of several layers of content categories, starting with the audio signal at its core and gradually adding layers of content that exhibit higher subjectivity and more semantics. Following is a description of the layers:

1. **Audio:** At the core of content is a lossy-encoded digital representation of the recorded (or digitally produced) acoustic signal, where features are extracted from the core audio without any semantics. Some typical examples on this level include time-domain, spectral, tonal, and rhythm descriptors.
2. **Embedded Metadata (EMD):** This layer of content encompasses the collection of information that pertains to the audio signal, such as artist name, producer, track title, release date or other multimedia data such as album cover artwork. Embedded metadata often serves as a bridge to the higher layers, linking audio content with expert, user-generated, and derivative content.
3. **Expert-Generated Content (EGC):** Expert-Generated Content contains richer and more semantic music metadata than EMD. This includes

attributes such as genre, style or mood, contextualises music in terms of era, origin and trend and provides a more detailed description of content or performing artist. The content is usually high quality since it is subjected to musicological perceptiveness but can exhibit bias.

4. **User-Generated Content (UGC):** User-generated content is any form of media, such as text, posts, images, videos, reviews, created by individual people and published online or to a social network. It is often referred to as community metadata and examples include tags, reviews, song explanations, and playlist-tagged tweets. In addition, the content is available in a variety of languages and modalities, and includes location-based information.
5. **Derivative Content (DC):** The Derivative Content is new content derived from original content in the inner layers of the model, offering information relevant for content recommendation. It comprises re-used original content, such as remixes, mash-ups, covers or parodies as well as repurposing it into movies, videos, and advertisements, producing a different perception of the original content.

Chapter 3

State of the Art

This chapter presents the state of the art in the domains related to the proposed model. The chapter is split into two sections, the first section being related to music recommendation systems, section 3.1 and the second section 3.2 investigating the state of the art of AIS.

3.1 Music Recommendation

3.1.1 Cold start

The cold start problem is a central challenge in recommendation systems and can be divided into three subproblems, data sparsity, new item, and new user.

- **Data sparsity:** the entire user-item matrix has a low amount of interactions
- **New user:** new user in system with few to none interactions
- **New item:** new items that lacks sufficient user interactions

While these three subproblems differ in various ways, the absence of sufficient user-item interactions is the root of the problem in all cases. Among these are items with few interactions due to low popularity, also known as items in the long tail of popularity. Cold start affects particularly collaborative filtering models since it bases the recommendations on user-item interactions, making it less likely to recommend items in the long tail.

An example of using content based filtering to recommend items can be seen in

the work by Soleymani et al.. The audio features used with CBF were assigned to five psychologically validated music attributes named MUSIC. The music recommendation was determined by the MUSIC attributes, which were compared with the users' listening history records. The listening records were based on users' explicit feedback on whether or not they enjoyed the song. The five MUSIC attributes proved to represent the users' preference for music recommendations effectively. As the model incorporated only five audio features, it was able to alleviate the curse of dimensionality and the cold start problem.

In spite of exploiting the new item problem, the constituent CBF fails to handle new users and sparse data. McFee et al. resolve these issues by improving the content-based audio similarity based on learning from collaborative data. Similarity learning is treated as an information retrieval problem, where the similarity is learned to optimise the ranked list of results in response to a query example. The improved similarity measure is then applied to previously unseen items for which collaborative filter data is unavailable. As a result, the proposed methods are said to outperform competing methods for content-based music recommendation. In addition to solving the cold start problem, this also deals with other challenges in music recommendation, such as diverse recommendations which is discussed in the next section.

3.1.2 Recommendation diversity, novelty and serendipity

Most of the research on music recommendation systems (MRS), measures their success based on how accurate they are at predicting highly relevant items to a target user, and the field has for long solely focused on promoting this perspective [29]. However, various studies such as [29] have pointed out the quality of recommendations in the other properties of RS (diversity, novelty and serendipity). Note here that the terms diversity, novelty and serendipity will be used interchangeably. Although this goal is often achieved at the expense of accuracy, it enhances the user experience by making the recommendation list more diverse and including more unknown items. As with the cold start issue, collaborative filtering impedes novel recommendations since it usually relies on historical data, so using content-based filtering would address this issue by taking advantage of audio similarity, seen in the previously mentioned paper [51]. In early studies, Yoshii et al. proposed a hybrid model with rating-based CF and acoustic CBF which outperformed both the constituting CF and CBF models in terms of accuracy and diversity. The diversity measure was based on how many different artists who were represented among the recommended items. Furthermore, the proposed system addressed the new-item challenge. Unfortunately, it came at the cost of computational complexity.

It may also be useful to incorporate diversity into the similarity measure, for example by using metric learning. Metric learning is a way to calculate the distance between objects. In [48], a hybrid system combined collaborative- and content based- filtering to improve the diversity of music recommendation. The similarity distance in the proposed model was computed with help of metric learning, where a dynamic weight, based on user interaction, was designated to different acoustic features. The goal was to minimise the distance based on audio content and user interaction patterns. The improved similarity estimation, resulted in better accuracy beside recommending a wider range of artist, indicating a rise to diversity as well. In the paper by [35], metric learning to rank items was suggested by combining artist-based similarity measures with data on audio usage derived from interaction data. The proposed model compact the audio by representing each item as a histogram over codewords by using vector quantisation. This optimises the feature space such that the system provides more novel recommendations based on audio content.

3.1.3 Automatic Playlist Continuation (APC)

A task that has been widely recognised in recent years is automatic playlist continuation (APC), which emerged from the ACM Recommender Systems Challenge 2018 (RecSys 2018) [9]. The task of APC consists of adding the most appropriate tracks that fit the same target characteristics of the original playlist. It is important to note that APC is a variant of automatic playlist generation (APG), which creates a sequence of tracks in accordance with some characteristics.

The most important factors that lead to positive user perception of a playlist include variety (e.g. in genres, style and artist), coherence (e.g. of songs, lyrical content, tempo and mood) and common theme (e.g. in location, story and era) [33]. Personal preference also plays a major role, meaning that a highly liked or disliked song has a strong influence on how the entire playlist is perceived. Furthermore, a good playlist should be familiar in theme, genre, or include a good mix of familiar and unknown songs. The APC typically misses the user's intent behind playlists, which is why the metadata associated with user-generated playlists such as titles and descriptions is possibly a good starting point to create intent-based models.

Neural network is a common approach for APC used to extract knowledge from manually curated incomplete playlists to learn the characteristics. In RecSys

2018 [9], this approach was the most popular among the top teams. Several teams took advantage of the multi-stage architecture, where most only had two stages. The first stage consists of retrieving a small set of relevant tracks. In this stage, the majority chose matrix factorisation as their primary CF approach to learn a low-dimensional dense representation for each playlist and track. Tracks that occur together frequently in user-created playlists are assigned similar representations. Therefore, tracks from a particular artist, album, or music genre may be assigned a close representation. There were also multiple teams that calculated the playlist-track similarity with neighbourhood-based collaborative filtering models. The second stage attempts to increase accuracy by re-ranking the small set given a set of features and is based mostly on MLR.

The teams that implemented hybrid approaches with content-based filtering performed marginally worse than those without [9]. It may be due to the fact that the additional information made the problem more complex and the solutions were unable to successfully generalise the information obtained from the external sources. Moreover, the majority relied on the descriptors from Spotify's APIs. This could potentially be more effectively addressed by extracting their own characteristics from the audio, for example the MUSIC attributes from Soleymani et al.. The evaluation in RecSys 2018 may not be representative of real-life as it did not consider diversity of recommendations and new items.

The content-based filtering RS of Dionisios N. Sotiropoulos has shown good results in similar circumstances to APC. Basing their approach of music recommendation on the fact that the recommendations rely on the feedback the user supply in the form of positive music instances (favourable songs). Their proposed method is an AIS-based one-class classifier, which filter out the negative instances by recognising the region space of positive instances. This approach to filtering out negative instances is known as negative selection, further explained in section 3.2.1. The main source of inspiration for the proposed approach relates to the fact that users' interests tend occupy a constrained volume of a given music collection, which can resemble the objective of APC. When compared with traditional one-class classification approaches, the results favour the the proposed approach.

3.1.4 Data sets

While the development of new algorithms and approaches within a field of research is of most importance, defining a benchmark dataset is also critical. Having

a reference benchmark makes it easier to compare results from the different systems and creates a robust foundation for evaluating challenges and breakthroughs within the research field. Within the world of MRS/APC, there is as of now, no established benchmark dataset such as MNIST, CIFAR, or ImageNet in computer vision. Music recommendation has, however, some additional challenges attached to its application area that makes it harder to create an established dataset that does not apply to the field of computer vision.

Appealing features (features to consider)

According to an article by FMA, [13] a benchmark dataset should contain some qualities. The following list presents what features a dataset within music information retrieval (MIR) should strive to contain in order to be a reference benchmark within the field.

- **Large scale:** In order to fully mimic the current diversity and millions of songs found in existing popular music platforms, there is important that the dataset contains the large amount and same type of songs in order to have an equal foundation for retrieving and recommending music.
- **Permissive licensing and available audio:** As most music is protected by copyright restrictions, creating a dataset, especially one containing raw audio, is a great challenge to the MIR research. A dataset in MIR is therefore often a smaller dataset distributed with audio, or larger dataset without audio. This creates two challenges, the first being that the dataset without audio restricts the researchers to only use the audio-features given by the datasets creators. This limits the possibility to try out new audio related features and find new exiting ways to describe music. The second challenge is that the datasets that provide links to sites containing the audio file, have no assurance that the files will disappear without notice.
- **Quality audio:** To be able to accurately extract audio features of a song, the audio should be of high quality. Many of the datasets that contains downloadable audio is of limited length (often 30 seconds). This can be a problem as some parts of a song might not be representative for the rest of the song and might lead to non-representative audio features.
- **Metadata rich.** A dataset rich with metadata should be included as this feature is shown to be useful in music recommendation [14].
- **Easily accessible:** If a dataset is easily accessible, the greater foundation for comparison.

- **Future proof and reproducible:** Research within an area takes time and progress and can extend over decades. It is important to have a reliable and reproducible dataset which is always available to the public. This guarantees that new research always can compare on the same premise as other research within the field.

Beside these qualities in the field of MIR, a benchmark dataset within MRS/APC should contain some additional qualities.

- **User feedback:** User feedback consist often of explicit feedback related to song preferences of a user i.e. the ratings of songs. This is a valuable metric to RS to more accurately evaluate the user’s song preferences.
- **User playlists:** As user’s often categorise playlists into songs with similar characteristics, user generated playlist can be used as a metric to evaluate how well a system recommends relevant songs. By sampling some of the songs from the playlist as test data, the models can measure accuracy against it. A common approach for evaluating classification systems.

Relevant datasets

Dataset	Audio features	Individual tracks	Release date	Listenable	User playlists
MPD	Yes	2.2 M	2018	Yes	Yes
MSD	Yes	1 M	2011	Yes	No
30Music	No	5.6 M	2015	No	Yes
FMA	No	106 K	2017	Yes	Yes

Table 3.1: Comparison of the different datasets

Dataset	Main strength	Main drawback
MPD	Easy to download, test and listen to	Few audio features
MSD	Popular within the field	No user created playlists
30Music	Explicit user feedback	No audio or audio features
FMA	Downloadable audio	Lacks mainstream music

Table 3.2: Strength and weakness with the proposed datasets

Million Playlist Dataset

The Million Playlist Dataset (MPD) [9] was released based on a competition in the Recommender System Challenge 2018, arranged by Spotify. The task consisted of APC, and the dataset was hence tailored to this task. The dataset consists of 1 million user-created playlists, 66 million tracks and about 2.2 million unique tracks. Each playlist in the dataset includes a title and a tracklist where each track is provided with metadata (track id, track uri, duration, and more). The track uri is useful, as it makes it possible to extract audio-features freely and not rely on pre-computed audio features. A separate challenge dataset (test-set) was also included to be able to objectively validate the performance of the competitors. This dataset consisted of 10,000 incomplete playlists where an unknown amount of songs were withheld from the original playlist, and 500 recommended songs were provided as candidate tracks to the playlist. The challenge was rereleased in 2020, and has at the time of writing a total of 115 participating teams [59]. The huge number of participating teams on the same dataset, gives a great foundation for comparing results.

Million Song Dataset

The Million Song Dataset (MSD) [6], is a public dataset containing audio features and metadata for millions of contemporary popular music tracks. The dataset was released in 2011 and has been a popular dataset for researchers in MIR. The dataset consists of 1 million songs, with 44,745 unique artists and each song has 55 fields of attached metadata. The fields contain both song-related metadata (year, artist name and bars start) and acoustic features (pitches, loudness and timbre). The dataset does not contain audio, but by the use of Echo Nest API alongside with 7digital [3], it is possible to fetch 30 seconds of audio samples for each song. Serving as a way to extract audio features by the use of other software. The dataset does also not contain any user-related data, which can be a challenge to be able to evaluate the performance of different MRS.

30Music

The 30Music dataset [Turrin et al.] was released in 2015 and is a collection of listening and playlist data retrieved through the Last.fm public API. The dataset was designed to overcome common challenges related to user modelling and music recommendation, and contains some attractive features. These include, user listening sessions with contextual time information, the user playlists, and positive explicit user ratings of songs. The dataset has 5,6 million tracks, 50,000 user created playlists and 600,000 artists formed by 45,000 different users. In total creating 31 million play events organised into 2,7 million sessions, with 4,1 million explicit ratings. It is also enriched with additional metadata of the tracks (title, artist, playcount) and the users (age, gender, country, playcount, number

of playlists, and more).

Free Music Archive

Free music archive (FMA) [13] is an online free music collection, and released three equal datasets of different sizes in 2017. The datasets is well known and has been used extensively within MGR approaches, but also in MRS [1]. The largest dataset contains 106,574 tracks distributed over 161 genres and comes with a rich amount of metadata related to tracks, albums and artists. It additionally contains user-related data of listening counts, favourites and user-mixes, which is important when evaluating different MRS. FMA's biggest strength, however, is that the dataset comes with full length high quality audio for each track. At the time, it was rare that such a huge dataset contained full audio, and this comes from the fact that all the music in the dataset is permitted for redistribution. While this is great, the fact that all the songs can be distributed freely, the dataset lacks mainstream music and commercially successful artists. This could be a challenge as the data does not resemble real world data and one might argue that free music is of "lower" quality.

3.1.5 Evaluation

To compare recommendation systems and assess their performance, a methodology for evaluation is required [50]. The evaluations can be performed in online and offline experiments. Online evaluation involve providing recommendations to the users and then asking them about how they rate the items [50]. Offline evaluation, on the other hand, does not require the participation of actual users. Online evaluations are preferred because they can yield more accurate results with real users [42]. However, users' evaluations have its limitations, as recruiting large cohorts of users for evaluation purposes is challenging and time-demanding. Often the recruited users are not representative of the general population as the recruiting process itself is biased-centric and therefore cannot be completely controlled.

There does, however, seem to be a common trend to conduct user evaluation through a questionnaire. There seems to be no established methodology for online evaluation for MRS. An example of this can be found in the paper by Bogdanov et al. [7], where a preliminary questionnaire was conducted with 12 participants on a content-based music recommendation model. The questionnaire consisted of asking the users to rate each song recommendation based on five variables: familiarity, liking, listening intentions, and "give me more". The three first scales ranged from 0 to 4 and contained two positive and two negative steps, along with a neutral step, and the last scale consisted only of 0 and 1.

Scale numbers were also accompanied by descriptions. For the familiarity variable, for example, 2 represented knowledge of the artist, whereas 3 represented knowledge of the title. Additionally, to measure individual metric such as liking and familiarity, Bogdanov et al. divided the ratings into categories evaluating if a song fulfils the type of recommendation: a hit, a trust and a failure. Songs with a low familiarity rating and a high rating were considered hits. Failures had low liking and listening intentions, while trusted songs had high familiarity, liking, and listening intention ratings.

An additional study that used an online questionnaire as its online evaluation was Kamehkhosh and Jannach. In a similar matter of [7], Kamehkhosh and Jannach conducted a questionnaire in which participants had to evaluate each song based on five questions. The evaluation, in this case, examined the similarity of the songs on a variety of dimensions in addition to personal preference. As opposed to [7], the similarity dimensions were rated on a seven-point Likert scale, a standard rating scale. The Likert scale range from -3 (fully agree with the negative term) to +3 (fully agree with the positive term) [45] [43]. The order of the questions was also randomised among all participants. By randomising the sequence of questions, the questionnaire can detect non-serious respondents [45] and force users to examine the alternatives carefully.

Evaluation metrics

When conducting an online evaluation of an MRS, it is crucial to carefully consider the kind of metrics the questions are designed to investigate. Similarity (see section 2.4) is the first important metric to consider. It has been demonstrated that the degree of similarity in recommendations has a strong correlation with users' perceptions of the system, especially in their acceptance and their trust [?]. Several different methods exist for measuring the similarity of recommendations. For instance, [28] asked participants to rate the extent to which they agree that the song fits the given playlist. The [7], on the other hand, measured similarity by the binary scale "give me more," which entails giving more items that are similar to the given recommendation.

Familiarity is another important evaluation metric, as Swearingen and Sinha emphasise in their user studies. Multiple online evaluations have also observed the familiarity metric [7][28]. The researchers found that familiar items play a key role in establishing trust in a system when examining user preferences. Trust provides the user with the feeling that the recommendation is tailored to their preferences [8] [4]. Although a user may be satisfied to hear a familiar song recommendation occasionally, they may be annoyed if every other song is familiar,

particularly in the context of music discovery.

An additional important evaluation metric mentioned in the two papers was user preference. In both papers the users were asked to rate each song based on how well they liked it. A high liking of the recommended items, strongly correlated with the same users' perceived usefulness and acceptance of the technology [40]. Thus, user preference for an item during online evaluation could be a critical metric to measure.

There has also been a strong focus on diversity within recommendation items [42]. Diversity (see section 2.4) in the music recommendations improves user satisfaction because it reduces over-personalisation [29] [32]. Music diversity is proven to be beneficial to music discovery [43], while also correlating with a high user satisfaction [40].

3.2 Artificial Immune Systems

3.2.1 AIS algorithms

The immune system is highly complex, and many different takes on transferring the aspects of the biological immune system to a computational approach exist. Some methods try to mimic the immune system as closely as possible, while others tailor their algorithms to its application domain and partly mimic the biological immune system. This section briefly describes the different types of AIS models that exist within the field as well as how they work.

Negative selection

ARTIS [23] is one of the more known computational approaches and has a bio-plausible implementation on many different parts of the biological immune system. The negative selection part of ARTIS gathers inspiration from the negative selection algorithm [19]. The NSA generates random negative detectors and destroys those that match any of the strings in the self-set. The amount required to activate a negative detector is determined by the calculated affinity between a detector and the encountered self or non-self string. If a negative detector survives over a time period of T in the provided environment, the negative detector is evaluated as mature. The mature detector, however, still needs to receive a second signal in order to stay alive. ARTIS does this with a human operator.

3.2.2 Clonal selection

The concept of clonal selection is based on the idea that only antibodies capable of binding to an antigen will proliferate [55]. The technique was first popularised by de Castro and Von Zuben, who developed an algorithm called CLONALG [11]. The algorithm revolves around a loop where a randomly selected antigen is presented to an antibody population [11]. The affinity between each antibody and the antigen is then calculated, and the n antibodies with the highest affinities are then submitted to affinity maturation. Affinity maturation involves cloning the antibody proportional to the affinity and mutating inversely proportional to the affinity. From this set of mature clones, the antibody with the highest affinity to the presented antigen is selected as a memory cell candidate. The memory cell candidate will replace the previously stored memory cell if the affinity with the antigen is greater. CLONALG is relatively low in complexity compared to other AIS systems and has few user parameters. The model is also known for its ability to solve both optimisation and classification, even though it is sub-optimal for the latter [49]. Therefore, Sharma and Sharma [49] proposed an improved version called CLONAX, which selects antibodies based on their accuracy of connected antigens.

Another widely known clonal selection algorithm is the Artificial Immune Recognition System (AIRS) [57] which can resemble CLONALG in the sense that both algorithms are concerned with developing a set of memory cells for classification. AIRS also employs affinity maturation and somatic hypermutation schemes that are similar to what is found in CLONALG. In addition, the algorithm uses population control mechanisms and has adopted the use of an affinity threshold for some learning mechanisms. It works in the following way: The algorithm presents every training antigen to the current memory cell population, which in turn identifies the best matching cell from the current memory cell population [57]. The best matching cell will then undergo affinity maturation to expand the ARB population. ARB stands for Artificial Recognition Ball and consists of an antibody, a count of the number of resources held by the cell and the current stimulation value of the cell. The affinity of each ARB from the updated population is then examined to determine the number of resources to allocate accordingly. The survived ARBs will then go through the training process, where they undergo rounds of affinity maturation and compete for resources. This will repeat until the average affinity value for all the existing ARBs with the antigen reaches the stimulation threshold. Once the training process for the current antigen is fulfilled, the best matching ARB with the same class as the antigen is nominated as the candidate memory cell. It will be added to the memory cell population and become a long-lived memory cell if it matches the antigen better than the previous one. Watkins et al. indicates that the algorithm does not sacrifice accuracy

while providing data reduction capabilities.

AIRS has achieved great success in solving optimisation and classification problems. However, the initial version of AIRS suffers from high computation cost, the exponential growth of generated data and the algorithm's complexity [36]. As consequence, there have been proposed multiple improvements on the model such as [20],[25] and [26]. A newer one from 2021 called revised-AIRS (RAIRS) [36], introduces some new mechanisms: deleting inactive memory cells to avoid data explosion, adding the concept of weight and lifetime counter for each memory cell to improve quality, and selecting only the best representative cells. In addition, slight modifications in the AIRS functionalities were made, like the mutation function and the memory cell introduction mechanism. RAIRS has proven to be the most effective version of AIRS as well as outperforming other state-of-the-art methods on the UCI datasets.

In AISLFS [16], each antibody secretes n clones, which then undergo mutation inversely to their affinity. These clones will go through a tournament selection. The clone that binds with the highest number of antigens and has the fewest selected features is chosen to replace the parent solution. Additionally, the algorithm includes an elimination process that iteratively removes redundant antibodies until the accuracy of the classifier decreases.

Evolutionary AIS

The clonal selection algorithm resembles the evolutionary algorithm, as they rely on selection, reproduction and mutation mechanisms. Vote-ALlocating Immune System (VALIS) [30] is an AIS that utilises these evolutionary capabilities. The algorithm starts initialising a set of antibodies by randomly selecting from the antigens. Then at each generation, it selects parents according to the antibody's local classification accuracy and a sharing factor. Subsequently, the selected parents produce new antibodies through crossover and mutation operators. The antibodies with the lowest fitness value will be replaced with the newly produced offspring. The VALIS algorithm performed on par with several established classification algorithms in experiments conducted on six popular benchmark problems. Additionally, the system demonstrates emergent global behaviours that are due to local antibody interactions. Although individual fitness is the basis of the training, the population as a whole converges towards a higher collective classification accuracy. The algorithm is also simple and has few parameters because it relies on the self-organisation of the antibody population.

3.2.3 Initialisation

The most widely used method to generate candidate solutions (initial population) is random initialisation if no information about the solution is available. This can be seen in many AIS applications, including ARTIS [23], CLONALG [11] and AIRS [57]. The difference is that ARTIS randomly generate antibodies while clonal selection algorithms [11][57][36][16] randomly select antigens to the initial antibody population. However, random initialisation can have a negative impact on the rate of convergence and the quality of the final solution. RAIRS [36], solves this by proposing a semi-random initialisation. The process includes a mean vector based on the average antigen vectors in the randomly initialised population. As a result, RAIRS will provide a good foundation for the model, thus increasing the likelihood of generating an individual of high quality. AISLFS [16], on the other hand, initialises the initial population with all antigens. This prevents antibodies from being placed in regions without antigens, which is useful in sparse or small datasets.

Recognition region (RR) radius initialisation

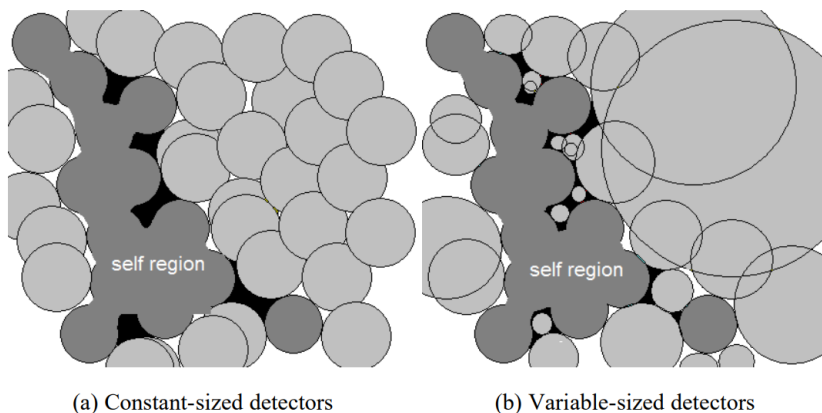


Figure 3.1: Variable-sized detectors cover more of the non-self space with fewer detectors [27]

Each antibody's recognition region radius is also defined during initialisation. The radius is usually determined by a user-defined parameter, making it challenging to decide since there are many factors to consider. For instance, large recognition regions can misclassify antigens. Further, the combination of too few

antibodies and too small regions can leave too much space in the shape space, leaving many antigens unclassified. The use of the same RR for all antibodies also poses a problem due to the presence of holes, which are common in negative selection algorithms [18] [23]. The V-detector algorithm [15] demonstrated that these holes could be filled by initialising the detectors with different RR radiuses. In this context, antibodies are referred to as detectors. The algorithm achieves this by assigning a variable radius based on the minimum distance to each detector that will be retained and matching the threshold rule with a self-antigen. The number of detectors is also reduced, which will decrease the time complexity of the algorithm and require less space. The ability of detectors with different RR radiuses to cover holes can be seen in figure 3.1) [15].

AISLFS also initialises the antibodies with different RR radiuses. In this case, the antibodies are automatically set at their highest possible RR radius, without containing antigens from a different antigen class. RR radius is re-calculated based on the same criteria at each iteration, which may make the process computationally intensive. In VALIS, the recognition threshold is initialised to the distance between the antibody and a random same-class antigen in the training set [30]. This increases the probability that every antigen will be covered by at least one antibody, that is, be within the recognition region of at least one antibody. Alternatively, if the antigens of the same class are not concentrated in the feature space, it may lead to antibodies covering antigens of a different class. The authors of MAIM [5], however, found significant improvements in accuracy after implementing the same initialisation scheme.

Recognition shape

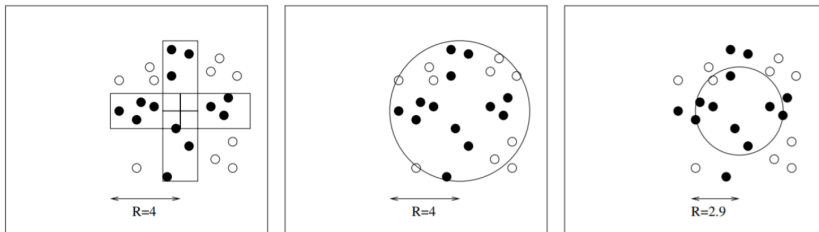


Figure 3.2: The shape of recognition shape can affect the classification accuracy [22]

The recognition region is specified by the antibody, the affinity function, and the radius. As mentioned, the shape of recognition regions depends on the application domain, but the most common is a hypersphere. However, [22] demonstrates that complex shapes can outperform hyperspheres in some cases. The figure 3.2 illustrates this point with an antibody having a cross-shaped recognition region that separates the two classes of data points, but not with an antibody that has a spherical region. Thus, the placement of antigens in the feature space may favour a shape that differs from traditional hyperspheres. In contrast, it may not be advantageous in the case of a small population.

According to a study by Ozsen and Yucelbas, dynamic recognition regions are also beneficial when used in the context of evolvable elliptical regions [38]. The recognition regions are subjected to three different mutation operators consisting of changing the ellipsis' centre, length, and orientation. Despite not having great improvement in solving linearly separable data sets, the algorithm seems to perform well and even better than other algorithms in terms of both training times and accuracy in complex nonlinear data sets. In addition, another RR algorithm that employs alternative shapes is AISLFS, which uses a variety of regional shapes such as spheres, cubes, and cylinders [16]. The majority of these are however only employed for a certain subset of dimensions as some shapes are inefficient when applied at higher dimensionalities.

3.2.4 Diversity

To facilitate global exploration and avoid poor performance caused by premature convergence, population diversity is an essential component of evolutionary algorithms. The following sections describe how different methods contribute to increasing the diversity of a population.

Parent selection strategies

Despite the fact that the AIS does not employ a parent selection strategy similar to the EA, there is still a selection phase to determine which antibodies will be used in the profiling. In clonal selection, the best antibodies are usually selected for proliferation, resulting in premature convergence and reduced diversity [21]. CLONALG, for instance, picks n antibodies that have the highest affinities with the presented antigen, while AIRS only selects the best matching memory cell for affinity maturation. The clones in AIRS will also have a chance to affinity mature if they are able to survive the training process, where they will compete for resources [57]. Consequently, the model may reduce diversity and converge prematurely even further. On the other hand, it would also assist in the exploitation of the offspring. For AISLFS, each antibody will undergo affinity maturation,

not discriminating the proliferation only to the fittest parents [16]. Compared to the other clonal selection methods, VALIS selects the parents, based on the antibody's individual local classification accuracy and sharing factor [30].

Mutation strategies

The clones are usually mutated inversely proportional to their affinity with a given antigen after proliferation, which gives the population the variation [21]. The mutation strategy depends on the problem domain but normally involves manipulating the feature values with some probability. In VALIS, the recognition radius is mutated using a log-normal random multiplier. In contrast, the feature values of the antibodies are mutated by adding a random variable with a log-uniform density and a random sign. The mutation process will be repeated until at least one mutation occurs [30]. In the original version of AIRS, the feature vector produced after the mutation is assigned randomly to a class which causes an increase in false classifications [36]. In order to overcome this problem, RAIRS propose to label the new clones with the class of the nearest mean vector of the antigens generated. As opposed to most AIS models, AISLFS does not change feature values but simply changes the subset of features by including or excluding one of them [16]. Consequently, each clone will have a different feature set than its parent, which changes the recognition region shape. Because mutations do not affect the positions of antibodies, the model relies on accurate training samples to place the antibodies strategically in the feature space.

Survivor selection strategies

While hypermutation does help with population diversity, it guides the model towards local optima. It is through survivor selection the population can avoid local optima which will attain the diversity in the population and explore new search regions [21]. CLONALG selects survivors by adding mutated clones to the population and reselecting some of them as memory cells. Memory cells are the antibodies for classification. In AIRS, the antibody with the highest affinity will be selected as a memory cell if it has a greater affinity than the previously selected memory cell [57]. It would then replace the previously selected memory cell with the new memory cell.

In RAIRS, the same principle of the AIRS [57] is kept except for the number of introduced memory cells in the updating process [36]. Indeed, the proposed mechanism improves the system and provides a high opportunity to produce a more representative model by adding all memory cell candidates with higher stimulation than the previous memory cell. AISLFS tries to replace the parent with the best-performing clone regardless if it has a lower fitness value [16]. This

allows the algorithm to escape from the local optima, boost diversity and prevent premature convergence.

Fitness sharing

Fitness sharing is a technique used by EAs to maintain the diversity of certain properties within the population (see section 2.1.8). This reduces the effect of premature convergence [52]. Although the traditional AIS models do not facilitate this technique, VALIS uses a mechanism similar to fitness sharing. This method is a part of the antibody's fitness calculation, which promotes diversity [30]. As a result, the antigen space is covered more uniformly, which leads to faster convergence of high-fitness areas.

The AIRS [57] system also employs a form of fitness sharing known as resource sharing. The aim of resource sharing is to filter the antibodies to keep only the most representative ones. A number of resources will be allocated based on the affinity of each antibody. There is a maximum resource allocation for each antibody class, and if the total resources allocated in a class exceed their maximum allocation, additional resources will be removed from the least stimulated antibody for that class.

3.2.5 Classification

The term classification strategies refer to the methods used for assigning antigen labels after the model has been trained. Normally, an antigen is classified into an antibody's class if it falls within its recognition region and has the highest affinity binding with that antibody [23] [11]. The antigens in AIRS are, however, classified in the opposite way. It instead applies a classification algorithm, such as k nearest neighbour (kNN), which classifies the antigen based on the majority of antibodies with the same class within a predefined radius. However, this method is not without drawbacks. When using kNN, for instance, the processing of all antigens is done iteratively, increasing the search cost [36]. Therefore, RAIRS proposed using the kd-tree structure for the memory cell set. This enables migration from sequential to binary search, which speeds up the search for kNN since the complexity is reduced from sequential to logarithmic. Nevertheless, classification algorithms face the same challenge in tuning the radius parameter as recognition regions. As opposed to most common class kNNs, VALIS uses independent votes based on binding weights [30]. As it relies on antigen-antibody interactions and not distance-based sorting, it does not have to consider the radius parameter like AIRS and RAIRS. In addition, VALIS only considers the antibodies that are actually connected to the antigens when classifying it, as opposed to looking at all the closest antibodies regardless of their existing connection between them.

Traditional binary or multi-class classifications typically lead to a bias towards the class(es), with the most instances in severely imbalanced datasets. Modelling and detecting minority classes under such conditions is extremely difficult. For instance, in a paper conducted by Serapiao et al., it was suggested that an imbalanced dataset had a negative impact on clonal selection classifiers' performance as opposed to neural network models. On the other hand, in a study from 2007, AIRS have shown to improve significantly the performance when the data are imbalanced and achieve comparable performance with ANN for relatively balanced data [58]. The negative selection algorithm is also a model that addresses the imbalanced dataset problem since it is typically implemented as a one-class classifier [46]. In one-class classification, the classification problem is addressed by examining and analysing instances of only one class, usually the one of interest. There have been successful approaches to recommendation systems with NSA, such as [15] for music, due to its one-classifier capabilities.

Chapter 4

Model and Architecture

This chapter introduces the architecture of the proposed model. Section 4.1 describes the dataset and its features. Further, section 4.2 presents the model structure, the representation and calculations, and the algorithm flowchart. Additionally, the section describes each process that takes place within the model in detail. Finally, it discusses the various similarity measures used in the generation of Recommended Song Sets (RSS). The last section 4.3 presents the online evaluation questionnaire designed for the proposed model. The source code for the proposed model can be retrieved from [37].

4.1 Dataset

The datasets presented in section 3.1.4 each had their own strengths, which meant the proposed model would have required different approaches in response to those differences. Ultimately, it was decided to use the Million Playlist Dataset (MPD) [9]. The limitation with MPD is that there are only nine audio features available. This is a disadvantage when using a recommendation system using content-based filtering. On the other hand, the few features from the dataset ensured that curse of dimensionality was less of a problem than expected. This enabled the proposed model to run more effectively and ensured that more extensive testing of the algorithm could be conducted.

The dataset can be downloaded from the AICrowd site [59] by participating in the contest. By default, the dataset consisted of JSON files, which included the playlist name, the track names with the track-URL, and other meta-data not too relevant to the model. An example of how the data was organised is shown in figure 4.1. The dataset by itself does not contain audio features; these have to be

taken from Spotify's API. Spotipy, a Python implementation of Spotify's API, made it easy to download the audio features, while also offering features that could automate the process of creating a Spotify playlist of the recommended songs. Implementations like these made the MPD an effective choice in terms of being able to early in the implementation process test the results.

```
{
  "name": "disney",
  "collaborative": "false",
  "pid": 1000,
  "modified_at": 1457827200,
  "num_tracks": 189,
  "num_albums": 16,
  "num_followers": 1,
  "tracks": [
    {
      "pos": 0,
      "artist_name": "Original Broadway Cast - The Little Mermaid",
      "track_uri": "spotify:track:5IbCV9Icebx8rR6wAp5hhP",
      "artist_uri": "spotify:artist:3TymzPhJTMypk7P5xkAhM",
      "track_name": "Fathoms Below - Broadway Cast Recording",
      "album_uri": "spotify:album:3ULJeOMgroG27dnp27MDFS",
      "duration_ms": 154506,
      "album_name": "The Little Mermaid: Original Broadway Cast Recording"
    }
  ],
}
```

Figure 4.1: Data-structure of a random playlist from MPD

Audio features are the lowest level of content, see section 2.6. Spotify's audio features, however, aren't as low-level as "traditional" audio features, instead they consist of a variety of features. A prominent example is "danceability", which is determined on the basis of musical elements such as tempo, rhythm stability, beat strength, and overall regularity. All of these audio features are described in more detail in the following table 4.1.

Audio features	
danceability	Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.
energy	Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.
loudness	The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typically range between -60 and 0 db.
speechiness	Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.
tempo	The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.
instrumentalness	Predicts whether a track contains no vocals. "Ooh" and "aah" sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly "vocal". The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.
liveness	Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.
valence	A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).
acousticness	A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.

Table 4.1: Description of audio features obtained from Spotify API

4.2 Model Architecture

4.2.1 Model Structure

The proposed model is inspired by the Artificial Immune Recognition System (AIRS), but in this case, altered to be a content-based filtering music recommendation system which solves the Automatic Playlist Continuation (APC) problem.

There are two alternatives to the proposed model presented: MAIRS 1.0 and MAIRS 2.0, which are compared and tested in the experiment plan. The main objective of MAIRS 1.0 is to adapt the original AIRS to APC while remaining true to the algorithm and encouraging diversity. MAIRS 2.0, meanwhile, deviates more from its original model since it strives to make up for the similarity lost through adaptation. The design decisions in this version involve more novel design ideas.

Each section is divided into two parts. Part one describes how the algorithms operate in both models. The second part examines the differences between MAIRS 1.0 and MAIRS 2.0 and explains some of the design decisions that were made.

4.2.2 Chromosome representation

The proposed model consists of two types of chromosome representations: one for antigens, memory cells (MC), and antibodies, and one for artificial recognition balls (ARBs).

In the original AIRS, each chromosome structure includes a class label, as it is designed to be a multi-classification model. In contrast, another AIS music recommendation model [15] did not consider this factor. As a negative selection algorithm, this model only had to classify positive instances. In light of the proposed model being also a recommendation system, it was decided not to include class labels. The main reason for this was that the proposed model that classifies items as "similar" did not need to track more than one label. It is also likely that the imbalanced data set would result in a stricter self-space with multiple labels, which will have a negative impact on diversity (see section 3.2.5).

Antigen and antibody representation

Antigens and antibodies are analogous in their representations as they only consist of a feature vector. The difference is that an antigen is what is presented to ARBs for stimulation or response, whereas an antibody is what is contained

within an ARB or a memory cell. Furthermore, the antigens are not mutated throughout the algorithm, which means that it remains unchanged throughout.

Memory cell (MC) representation

Memory cells also consist of only a feature vector. The features vector are derived from antigens during memory cell initialisation or from the memory cell candidate(s) (MC candidates) during memory introduction. The memory cell is used to classify items. In MAIRS 2.0, the MCs include a label that identifies whether the MC represents an outlier (see 4.2.7).

Artificial recognition ball (ARB) representation

ARB is represented by an antibody, the number of resources it possesses, and a value that indicates the current stimulation level. In the original AIRS, resources are equivalent to fitness. Each ARB is assigned resources based on its stimulation value and clonal rate. The purpose of resources is to limit the number of ARBs allowed in the population. This is further discussed in section 4.2.9.

4.2.3 Affinity, Affinity Threshold, Stimulation Calculation

In the proposed model, three calculation measures are employed:

Affinity calculation

The affinity measure used in the proposed model is the Euclidean distance. Affinity measures the degree of similarity between the features of an antigen, an antibody or a memory cell. Two individuals with low affinity for each other appear similar and close in their feature space. The affinity measure is a determining factor for the other two measurements.

$$Affinity(p, q) = d(p, q) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (4.1)$$

Affinity threshold

The affinity threshold metric is calculated upon initialisation and measures the average affinity of all antigens. It is used during the introduction of memory cells to determine whether the current most stimulated memory cell should be replaced with a new memory cell. Calculation of the affinity threshold is as follows:

$$AT = \frac{\sum_{i=1}^n \sum_{j=i+1}^n Affinity(ag_i, ag_j)}{\frac{n(n-1)}{2}} \quad (4.2)$$

where n is the number of antigens and affinity between two pairs of antibodies or antigens.

Stimulation Calculation

The stimulation is defined as the inverse affinity and measures the degree to which an antigen is able to stimulate a cell. The measure is used when generating mutated clones and identifying the most stimulated memory cell and potential memory cell candidates.

$$Stimulation(ag, mc) = 1 - Affinity(ag, mc) \quad (4.3)$$

Playlist range calculation

The playlist range is a list of the maximum and minimum values for each feature in the antigen vector. It is used at two points during the algorithm: mutations in affinity maturation (see section 4.2.8) and the range method in classification (see section 4.2.11). The playlist range is used to determine whether the i th feature of a vector falls within the playlist $range_i$. To filter out the outliers, the playlist range is positioned between the first and third quartile of the original feature range. Each feature range in the playlist is calculated as follows:

$$range(f_i) = [Q1(f_i), Q3(f_i)] \quad (4.4)$$

f_i represents the i th feature; Q1 is the first quartile of the feature value across all antigen vectors, while Q3 is the third quartile. The playlist range is also calculated upon initialisation. Note that this calculation is not included in the original AIRS.

4.2.4 Proposed model flow chart

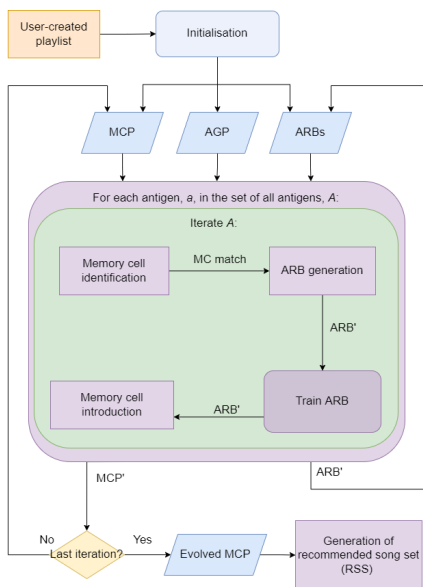


Figure 4.2: MAIRS 1.0

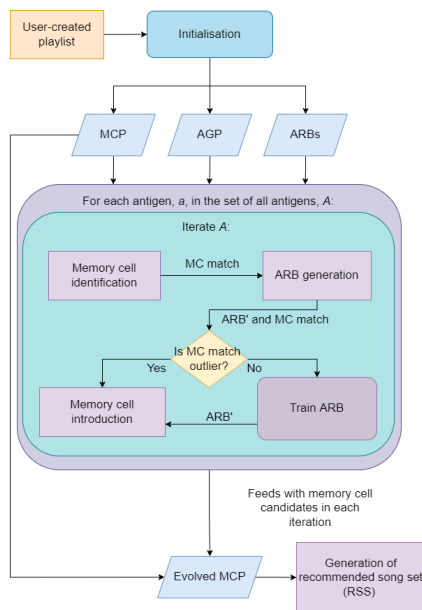


Figure 4.3: MAIRS 2.0

Figures 4.2 and 4.3 illustrates the general overview of the two proposed models, MAIRS 1.0 and MAIRS 2.0. The sections below provide further details regarding each process in the figure.

MAIRS 1.0 begins initialising the antigen population (AGP), memory cell population (MCP) and ARB population (see section 4.2.6). The process then enters the main loop within the figure where one of the antigens is presented to the MCP to locate the memory cell most stimulated by it called (MC match). This procedure is known as memory cell identification (see section 4.2.7). Upon ARB generation, the most stimulated memory cell will undergo affinity maturation, in which the mutated clones become members of the ARB population (ARB'). Subsequently, the ARB population is trained to develop potential MC candidates (see section 4.2.9), which during memory cell introduction might replace the MC match in the MCP (MCP') (see section 4.2.10). The updated ARB' and MCP' are then passed to the next iteration, and the loop continues to present the next

antigen from the AGP. This process within the main loop will repeat until the last antigen is presented. As soon as the last iteration has been completed, the MCP' is set as the evolved MCP. The evolved MCP is then used to classify items as "similar" (see section 4.2.11). In this thesis, the items classified as "similar" are considered the most relevant song recommendations for the user-created playlist and are referred to as Recommended Song Set (RSS).

As opposed to MAIRS 1.0, MAIRS 2.0 has an empty ARB population for each iteration. Additionally, MAIRS 2.0 feeds the empty evolved MCP with MC candidates after each iteration, as well as adds the entire MCP at the end. It also checks whether the MC match is an outlier during memory cell identification before deciding whether or not to train the ARBs generated upon ARB generation. Compared to MAIRS 1.0, the procedure for initialisation, memory cell identification, and memory cell introduction differ slightly in MAIRS 2.0. The following sections provide further details.

4.2.5 Model Parameters

The parameters of the proposed algorithm are presented and explained in table 4.2, as well as which model they are associated with.

Parameter	Explanation	Model
Maximum number of resources	Maximum number of resources allowed in the proposed model	Both
Mutation rate	Indicates the likelihood of an individual in the ARB population being mutated or not.	Both
Clonal rate	Combined with Hypermutation rate, clonal rate determines the number of mutated clones that an ARB produces.	Both
Hypermutation rate	Combined with clonal rate, hypermutation rate determines the number of mutated clones that an ARB produces.	Both
Stimulation threshold	A parameter between 0 and 1 used to indicate if the training of a specific antigen is achieved or not.	Both
Affinity threshold scalar	A value between 0 and 1 used to substitute the memory cells in the memory cell introduction stage.	Both
Size of initial memory cell population	Specifies the initial size of the memory cell population A size greater than the size of the playlist will result in duplicate memory cells. duplicated.	MAIRS 1.0
Size of initial ARB population	Determines the initial size of the ARB population.	MAIRS 1.0
Outlier distance	A value between 0 and 1 is used to determine if a memory cell is an outlier	MAIRS 2.0
Memory cell candidate size	The number of memory cell candidates to include in the evolved memory cell population after each iteration	MAIRS 2.0

Table 4.2: The parameters of the proposed models

4.2.6 Initialisation

The AT and the playlist range are calculated during initialisation, as mentioned in 4.2.3. Then, the proposed model starts initialising the AGP with the feature vectors from the original playlist. The original playlist consists of a feature vector

taken from Spotify API (see section 4.1). The feature vectors (antigens) are normalised such that the distance between two feature vectors is always between 0 and 1. Specifically, the Min-Max normalisation is applied separately for each feature value. It is calculated as follows:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (4.5)$$

where x represents the old value, x' represents the new value. The $\min(x)$ and $\max(x)$ values in the formula denote the minimum and maximum value across all the data samples' i th feature vector index, where $0 \leq i \leq 9$. Thus, the feature values at the i th and $(i + 1)$ th indices are considered independent when performing normalisation.

Two MCP and ARB initialisation strategies have been considered for the proposed model (see section 3.2.3). The first is the standard initialisation procedure of AIRS [57], which involves selecting n random antigens from the AGP to fill the MCP and ARB population. The population is sized according to parameters in table 4.2. The second is RAIRS [36] initialisation, which also fills the populations with randomly selected antigens. The difference, however, is that RAIRS also includes a mean vector, which is based on all the antigens, in MCP.

MAIRS 1.0 use the RAIRS initialisation approach. This is because the mean vector will serve given a good starting point, which increases the likelihood of a good match being found.

MAIRS 2.0, on the other hand, initialises all antigens as the MCP and leaves the ARB empty. Since MCP is essential for learning the model, the random initialisation strategies can result in a lack of or insufficient representation of some memory cells. This may affect the quality of the results obtained. When the AGP is small, it is essential to include most of the available antigens in the MCP. Considering that the AGP is initially small for the application domain, the proposed model does not benefit from sampling randomly among the antigens but instead includes all of them.

4.2.7 Memory cell identification

MAIRS 1.0 introduces memory cells in the same manner as the original AIRS. Memory cell identification can be viewed as parent selection (see section 3.2.4). As mentioned in section 4.2.4, the process locates the memory cell that is most stimulated by the presented antigen. The most stimulated memory cell is called the MC match. This applies regardless of whether the feature vector in the

memory cell and the presented antigen are identical.

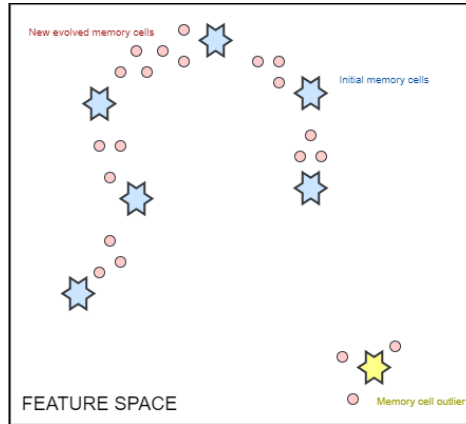


Figure 4.4: The feature space in MAIRS 2.0 after the main loop has been iterated

MAIRS 2.0 also performs memory cell identification in a similar manner as AIRS, although there are a few differences. The first difference is that the MAIRS 2.0 does not allow the presented antigen to stimulate a memory cell that has the same feature vector. In other words, since MAIRS 2.0's MCP is initialised with all the antigens, the presented antigen cannot choose itself as an MC match. This way, the presented antigen will generate and stimulate ARB clones with the closest memory cell and cover the space between them. Figure 4.4 illustrates this point. MAIRS 2.0 also labels the MC match as an outlier when the stimulation threshold with the presented antigen is out of range based on the outlier distance from table 4.2. Consequently, the MCP will not be affected by the outliers and will only create ARB clones around itself. The ARB generation made from the MC match will not undergo train ARB process.

4.2.8 ARB generation

Once the MC match has been identified, the MC match generates new ARBs through affinity maturation. Affinity maturation is the process of when the cell produces clones that eventually mutate in response to stimulation with the presented antigen. The clones become ARBs that consist of the same feature vector as the MC match. The ARB's stimulation and resource value are later updated during train ARB (see section 4.2.9).

A uniform mutation operator is employed during affinity maturation, controlled by the mutation rate parameter as shown in table 4.2. The mutation operator is applied to each feature value in the vector of n dimensions, with a probability of $1/(1+n)$. When a feature value is selected for mutation, it is multiplied by a random real number between 0 and 1. The mutation value is valid if the new $feature_i$ falls between the values in the playlist $range_i$. Otherwise, it will continue until the criteria are met.

If one feature of the vector is mutated, the clone will be added to the ARB population. The clones of the MC match are put repeatedly through this mutation procedure until the number of mutated clones is produced. The mutated clones are then added to the population of ARBs.

The number of mutated clones is determined by the dot product of hypermutation rate, clonal rate and the MC match's stimulation value. If the MC match is an outlier, which only applies in MAIRS 2.0, the stimulation value is not included in the number of mutated clones calculation.

4.2.9 Train ARB

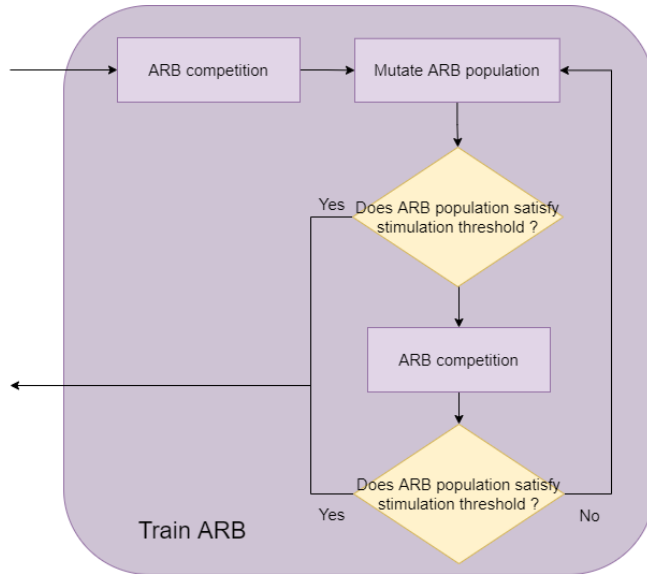


Figure 4.5: The flow chart of process Train ARB

Train ARB is the process by which the ARBs in a population evolve into more stimulated cells in response to the antigen presented. In order to achieve this, the ARB population must first undergo competition. The remaining ARBs will then undergo affinity maturation. ARB competition and affinity maturation will then repeat until the stimulation threshold is reached. The flow diagram of train ARB can be seen in figure 4.5 and is further described in the following sections.

ARB competition is based on the fitness sharing (see section 3.2.4) found in evolutionary algorithms. The goal is to refine the ARB population so that only the ARBs most stimulated by the presented antigen are retained. The ARB competition proceeds by first computing the stimulation level for each ARB. Resources are then assigned to each ARBs based on the calculated normalised stimulation value and clonal rate, which is computed as follows:

$$\begin{aligned} ab.stim &= \frac{ab.stim - minStim}{maxStim - minStim} \\ ab.resources &= ab.stim \times clonal\ rate \end{aligned} \quad (4.6)$$

The resources are removed from the least stimulated ARBs until the maximum number of resources is reached. The maximum number of resources is defined by the model parameter in table 4.2 determines how many resources are available to the population as a whole. Those ARBs left with zero resources are removed from the ARB population. See figure 4.5 for an illustration of the ARB competition.

After the ARB competition, the ARB population consists of those who were successful in acquiring resources. Each ARB then has a chance to produce mutated offspring to increase the diversity of the population, which is similar to the affinity maturation described in section 4.2.8. There is, however, a subtle difference between affinity maturation of surviving ARBs and affinity maturation of MC matches. Instead of a random number determining the mutation operator, it will depend on whether the stimulation level of each ARB is greater than the random number. The number of clones is calculated by multiplying the stimulation level with the clonal rate.

Next, the stopping criterion is calculated to evaluate if the stimulation of the ARBs with the antigen is sufficient. If the stopping is reached, the proposed model proceeds to the next step in the main loop; otherwise, the ARB competition and affinity mutation processes are repeated sequentially until the stimulation threshold is achieved. Figure 4.5 shows that the stop criteria are examined after each process. The stimulation threshold is reached if and only if:

$$\begin{aligned}
S &\geq \textit{stimulation threshold} && \text{where} \\
S &= \frac{\sum_{j=1}^{|ARB|} arb_j \cdot stim}{|ARB|}, && arb_j \in ARB
\end{aligned} \tag{4.7}$$

4.2.10 Memory cell introduction

The process of memory cell introduction is similar to survivor selection (see section 3.2.4). During memory cell introduction, the trained ARBs in the population have the opportunity to become memory cell candidates (MC candidates) and replace the MC match in the existing MCP. The ARBs with a higher stimulation level than MC match with the presented antigen are considered MC candidates. If the affinity between the MC candidate and MC match is less than the dot product of the AT and the ATS, the MC match is replaced.

Several memory cell introduction alternatives have been considered for the proposed model. One of the methods is the original AIRS, which substitutes the first MC candidate for the MC match. Thus, it is a 1-on-1 substitution. RAIRS, on the other hand, replaces the MC match with all MC candidates with a higher stimulation level to provide a higher opportunity to produce a more representative model. Moreover, the AISLFS replacement method has also been considered for the memory cell introduction. In this case, it will replace the parent with the offspring regardless of the stimulation level to enhance diversity.

MAIRS 1.0 follows the original AIRS memory cell introduction for simplicity. In MAIRS 2.0, however, memory cells are introduced in a similar manner to RAIRS. The model will include a selected number of MC candidates (based on parameter in table 4.2), except that they will not replace the MC match.

4.2.11 Generation of Recommended Song Set (RSS)

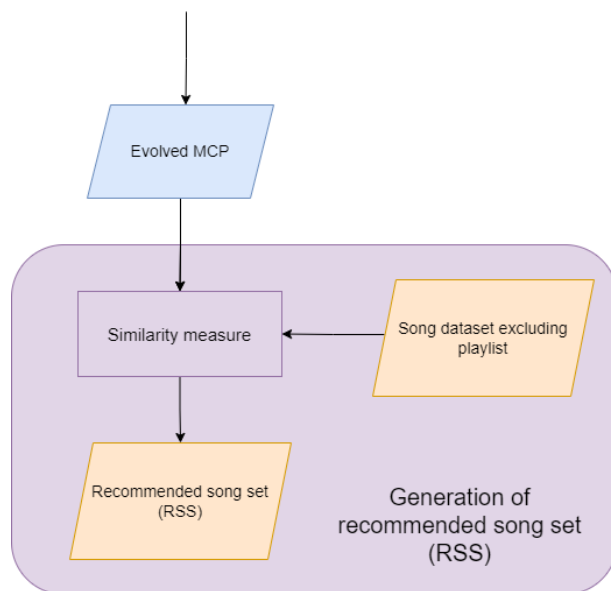


Figure 4.6: The flow chart of generation of RSS

In the original AIRS, the memory cells from the evolved MCP were used for classification after the main loop was completed. The classification was accomplished using a kNN (k nearest neighbour) classifier. The traditional classification methods, however, did not perform well with the proposed model. This was due to the proposed model being a one-class classifier and the dataset being imbalanced, which is further explained in preliminary testing (see section 5.1). Therefore, alternative "classification" methods were suggested called similarity measures.

Generation of RSS can be viewed as the same process as class prediction of items in the original airs and is illustrated in Figure 4.6. The evolved memory cell in the proposed model is presented to the dataset of all songs as the vectors. The vectors of the evolved memory cells then identify which song should be classified as "similar". There are various methods that can be used to conduct this classification, referred to as similarity measures. There are four similarity measures available in the proposed model, which are presented in the following sections. All songs that are classified as "similar" are considered the most relevant songs

for the original playlist. The classified items are also ranked in order to evaluate the proposed model.

Closest affinity

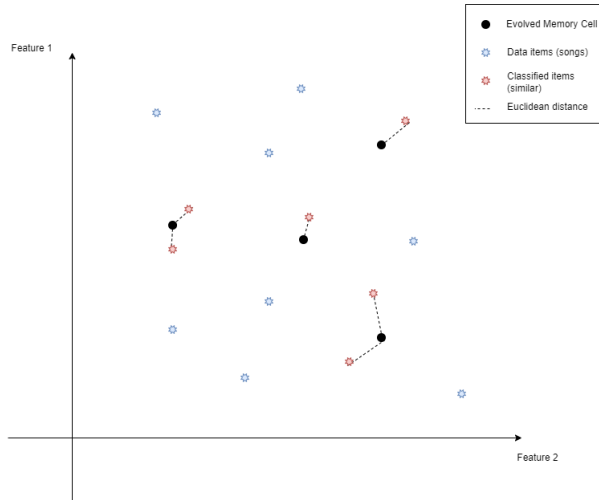


Figure 4.7: Closest affinity - Classified items is the closest ones to the evolved memory cells

The "closest affinity" method will classify a data item as "similar" if it has the highest feature space similarity relative to another evolved memory cell. These are the same items that are closest to one of the evolved memory cells in the feature space. See figure 4.7. The method is based on the fact that the songs in the same playlist for APC (see section 3.1.3) tend to share the same feature characteristics. As the evolved memory cell is derived from the original playlist songs, it is reasonable to assume that the closest songs would fit the original playlist. Therefore, the items with the shortest Euclidean distance relative to an evolved memory cell will be the most similar for that playlist. A drawback with this method is that it could limit diversity, as there is a chance that the recommended songs have a high similarity with the same songs.

Average affinity

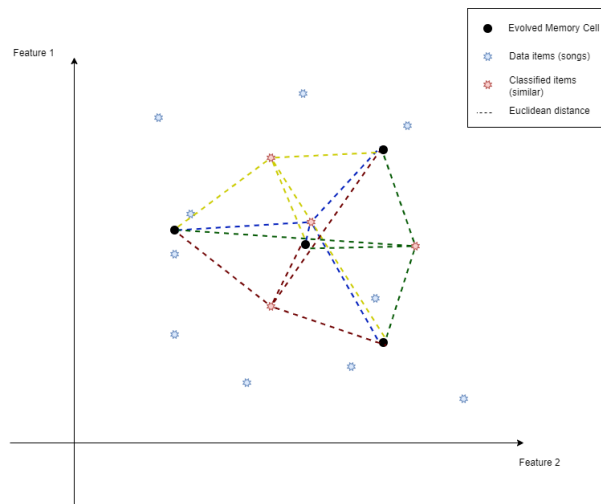


Figure 4.8: Average affinity - Classified items is between the evolved memory cells

Similarly to the "closest affinity", the "average affinity" method classifies items with the shortest Euclidean distance, except that it takes the average of all the evolved memory cells. This is illustrated in figure 4.8. The idea behind the "average affinity" method is that for APC, a "similar" song cannot be determined by only one song but by all of them in a playlist. Therefore, it considers all evolved memory cells rather than just a few. The classified songs will also possess playlist characteristics rather than specific song characteristics. This method, however, is susceptible to outliers as it considers all of the songs in the original playlist. In other words, the average affinity method would not be effective if a song in the original playlist does not fit the playlist characteristics.

Range method

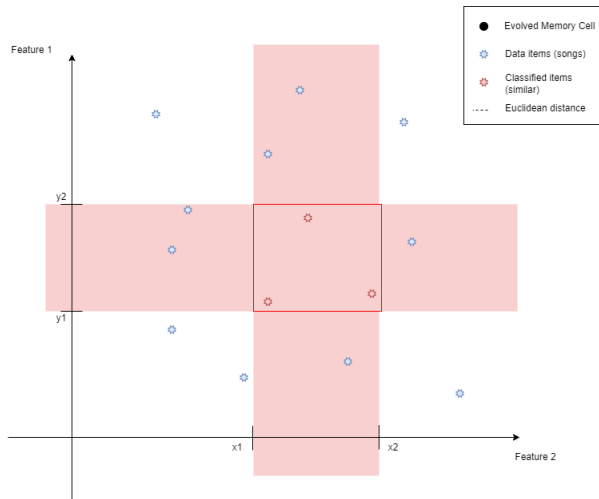


Figure 4.9: Range Method - Classified items is where the various playlist range overlaps

Each feature in the dataset has its own range, which is between the lowest and highest feature value in the distribution. However, for a given playlist, this feature range is restricted. Generally, songs in a playlist have similar features, so the interval between them tends to be smaller, resulting in a more limited feature range. The restricted feature range would therefore be a reasonable indicator of the characteristics of a given playlist. The range method will classify a data item as "similar" if the $feature_i$ falls within the values of the playlist $range_i$ (see section 4.2.3). The n top relevant songs will be the ones with the highest number of features that fall within that calculated feature range. Figure 4.9 illustrates the method. It is important to note here that this similarity measure is only applied when antigens is set as memory cells, due to time restrictions.

Recognition region

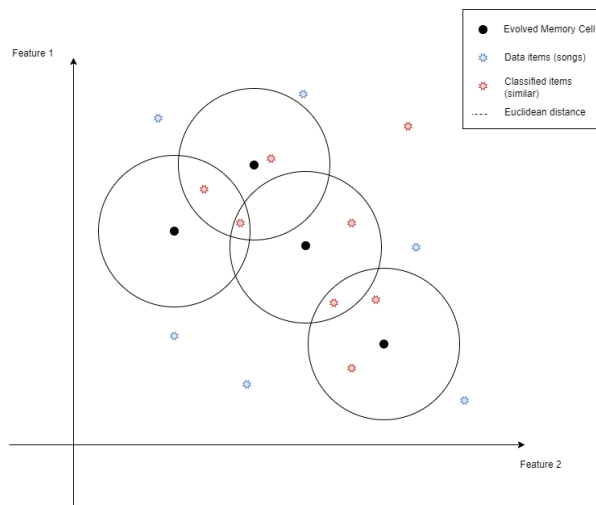


Figure 4.10: Recognition region - Classified items is (usually) where the recognition regions overlaps

The recognition region approach classifies items similarly to the classification scheme of ARTIS and CLONALG (see section 3.2.5). In the proposed method, items are classified as "similar" if they fall within a pre-defined radius of an evolved memory cell, which is illustrated in figure 4.10. A key objective of this approach is to produce the same results as average affinity while avoiding outliers. Given that the dataset is relatively large, the recognition region approach is likely to classify a large number of items as "similar". To find the most relevant songs for the original playlist, the model ranks the items by the number of recognition regions it falls into.

4.3 Online evaluation questionnaire

Diversity
<i>Genre</i> : 'The genre of this song suits the original playlist'
<i>Mood</i> : 'The mood of this song is similar to the mood of the original playlist'
<i>Musical style</i> : 'This song sounds similar to the songs in the original playlist'
Liking
<i>Personal preference</i> : 'I could see myself adding this song to my own playlists'
<i>Personal preference</i> : 'I would like to listen to this song again'
<i>Personal preference</i> : 'I like this song (independent of the original playlist)'
Relevance
<i>Playlist fit</i> : 'This song fits the original playlist'
<i>Playlist fit</i> : 'I would add this song to the original playlist'
Familiarity
<i>Artist</i> : 'This artist is familiar to me'
<i>Song</i> : 'I have heard this song before'

Table 4.3: Questionnaire designed to measure the four different evaluation metrics: diversity, liking, relevance and familiarity

To evaluate the proposed model, an online evaluation was designed and conducted (see section 3.1.5). This is mainly due to difficulty measuring accuracy through offline evaluation with the chosen dataset which is explained further in section 5.1.2. Online evaluation is also preferred over offline evaluation as it provides more accurate results when it involves real users.

The chosen methodology of the online evaluation was a survey. A total of 10 statements were asked for each song in the RSS in order to collect information related to these metrics. Each statement was graded on a Likert scale between 1-7. A number of the statements are different versions of one another in order to minimise how the interpretation of a single statement could influence the results. The statements and their related evaluation metric is presented in the following table.

The evaluation metrics were determined on the basis that the most significant factors that contribute to a positive perception of a playlist are diversity, coherence, and a common theme (see section 3.1.3). Liking plays an important role as well, and the playlist should be familiar in the theme and genre or mix familiar and unfamiliar songs.

The proposed evaluation also draws its main inspiration from the survey conducted by [7] and [28]. For instance, the metrics familiarity and liking are measured in both evaluation surveys. These metrics have been considered important metrics to measure since they provide insight into the users' trust, acceptance, and utility of the model. The proposed also chosen to evaluate similarity as the [28] example does. Thus, the participants are asked to rate how strongly they agree that the song fits the playlist and if they would add it to the playlist. It was found that diversity is beneficial to music discovery, which is strongly correlated to user satisfaction [40]. Therefore, the diversity metric is also included in the evaluation and is based on mood/reference, genre and musical style.

Chapter 5

Experiments and Results

This chapter first explains the preliminary tests conducted during the development of the algorithm in section 5.1. Following up with a brief explanation of the visualisation tools used to view the results in section 5.2. Section presents 5.3 the experimental plan of the tests. Meanwhile, section 5.4 presents all the parameters and data essential to be able to repeat the experiments. Lastly, section 5.5 describes in detail the findings of the experiments.

5.1 Preliminary tests

The final version of the model was influenced by several discoveries made during the development process. This section contains the experiments' results and discoveries made during this development.

5.1.1 Effects of KNN as similarity measure

KNN is a classification method but was intended to serve as a similarity measure in this project. Initially, this similarity measure was believed to provide good results. The idea was to divide the dataset into a test- and training set. The training set would consist of a number of songs from the original playlist, decided by a split percentage which AIRS would expand into evolved memory cells. In theory, this would make the training set consist of artificially yet representative songs (in terms of audio features). On the other hand, the test set would consist of all songs in the dataset except the songs in the training set. Each song in the test set would then be labelled as either "similar" or "not similar". Songs in the training set would be labelled as positive instances (similar songs), while songs not a part of the training set would be labelled as negative instances (not similar

songs).

Even though AIRS the training set was extended, there was still a significant imbalance between positive and negative instances. Negative instances would then likely cover a large part of the feature space. The imbalance would favour negative instances, leading to more negative neighbours upon kNN-voting. This result in almost no songs being classified as "similar".

On top of this, all songs had the same score in relevance, making it impossible to separate the relevance of the songs. This discovery led the research towards other similarity measures that would enable the possibility of separating the relevance of the songs (see section 4.2.11).

5.1.2 Accuracy measurement of the RSS

The second discovery that had an impact on the development process was the finding that accuracy as an evaluation method served no purpose. The initial plan was to divide the original playlist into two. One part (OA) represents the training-set, while the other part (OB) would serve as subjects to evaluate the accuracy of the algorithm when the model recommends songs. If all of the songs from OB were included in the recommended songs set (RSS), the RSS would receive a prediction accuracy of 100 percent since it would have predicted all songs that originally appeared in the playlist. However, when OB (5-30 songs), is part of a test-set consisting of 2.2 million songs, and the recommendation size was limited to a normal user session (5 to 40 songs), it was discovered that the RSS in almost all cases had an accuracy of 0 percent. Even though the size of the RSS should represent a normal user listening session, it was tried to increase the RSS size to 500 songs, in order to bump up the accuracy score. However, the accuracy was still too low to be able to assess the performance of the RSS.

These discoveries made it essential to find other ways to assess the quality of the RSS. Two methods were evaluated. The first method consisted of counting the genre-hit percentage for each of the songs in the RSS compared to the five most representative genres of the original playlist. Even though genre hit percentage does not essentially describe the relevance of the songs in RSS, it could be used as a tool to effectively give an indication of whether a new model implementation or parameter adjustment was worth looking further into. The method was, however, eliminated due to Spotify's API ultimately rejecting the necessary API calls to obtain the genre for all of the songs in the dataset.

A second, and ultimately the final method of evaluating the quality of the RSS

was by manually listening to the tracks and subjectively assessing their relevance to the original playlist. The subjective assessment of the results would adversely affect the project's credibility. However, it was merely a means to determine whether the model was heading in the right direction following various adjustments and additions.

5.1.3 Audio features and playlist type

Throughout the development of the model, the RSS was continuously evaluated. Early on, it was discovered that playlists containing music with higher relevance to the audio features (see table 4.1) would give the RSS a much higher score in terms of relevance than other types of music. Playlists containing acoustic traits like piano or country (guitar) seemed to consistently score better than playlists where the audio features were not that relevant to the genre of the original playlist, like rap or hip-hop. These results gave confidence in that the model could recommend relevant songs to any playlist as long as the audio features provided were relevant to the playlist. Due to the confidence that the model could provide highly relevant recommendations within these types of playlists, it was easier to evaluate the amount of diversity in the models within these genres.

5.2 Visualisation tools

Instead of comparing the values of numbers, graphs have been used to provide a basis for evaluating and comparing the results of the experiments. The most natural way to obtain results from the users was by using Google forms. It was decided to store the results from the users in a Google Spreadsheet since the data from google forms is easily transferable between these tools. A spreadsheet was finally made to combine the results in order to extract and visualise the data of importance from the experiments.

5.3 Experimental Plan

The experiments are designed to address the research question presented in section 1.2. The experiments are structured in such a way that they must be conducted chronologically. This is because implementation decisions for the following experiment are based on the results of the preceding experiment. In all of the experiments, the performance of the different models was evaluated by either the researchers or users through online evaluation from section 4.3.

5.3.1 Overview of experiment plan

Description	Hypothesis
Exp.1	
<p>Four evaluation metrics are compared based on their relevance to the original playlist. This experiment aims to investigate the ability of different similarity measures to recommend relevant songs based on the original playlist. In that sense, the experiment is conducted without MAIRS.</p>	<p>Range method is designed to capture the characteristics of a playlist within a set range for each audio feature. Since music is diverse and even songs labelled within the same genres can have different audio features, a range for each feature is expected to describe and filter similar songs to a higher degree (RQ1).</p>
Exp.2	
<p>Exp.2 seeks to investigate whether MAIRS 1.0 can recommend diverse songs while maintaining the same relevance score as the winner in Exp.1. This study aims to assess how encouraging diversity in an MCP affects the final RSS. Furthermore, to determine whether the proposed model is comparable to or superior to the winner from the first experiment in recommending "similar" songs.</p>	<p>MAIRS is expected to be on par with Exp.1 winner in terms of relevance and outperform in diversity, as the evolved MCP is more population diverse (RQ2).</p>
Exp.3	
<p>MAIRS 2.0 is investigated in this experiment to determine whether the design decision to decrease the recommendations' diversity has an impact on the final RSS. The results are compared to MAIRS 1.0, which lacks these new design decisions. The performance is evaluated based on evaluation metrics ??, but with an extra focus on the difference in similarity between the two methods.</p>	<p>MAIRS 2.0 is expected to recommend less diverse songs than MAIRS 1.0, while outperforming it in terms of relevance. This is because MAIRS 2.0 is designed to explore search spaces between the evolved memory cell (see figure 4.4) population resulting in a more strict search space than MAIRS 1.0 (RQ3).</p>
Exp.4	
<p>The final experiment evaluates the performance of the evaluated best model based on the previous experiments, compared to Spotify's recommendation algorithm. A user survey with 12 users is conducted to obtain the most objective results possible. The users will be asked questions regarding liking, diversity, relevance, and familiarity with each recommended song.</p>	<p>Spotify's algorithm is expected to outperform the proposed model in terms of relevance. Their model is more advanced and makes use of collaborative filtering combined with content-based filtering. As of the time of writing, this combination is considered state-of-the-art in the MRS field. However, the proposed model is expected to outperform in the area of diverse recommendations (RQ4).</p>

Table 5.1: Experiment plan overview

Related experiment	Methods compared
Exp.1	Range method — Closest Affinity — Average affinity — Recognition region (0.1, 0.2, 0.5)
Exp.2	MAIRS 1.0 + Exp.1 winner — Exp.1 winner
Exp.3	MAIRS 1.0 + Exp.1 winner — MAIRS 2.0 Exp.1 winner
Exp.4	Exp.3 winner — Spotify recommendation algorithm

Table 5.2: The methods compared within the different experiments

5.4 Experimental Setup

This section presents the parameters and other important details to keep in mind to be able to repeat the experiments. The table 5.5 shows the default parameters applied in all the experiments, and stays constant unless stated otherwise.

Parameter	Value
Default experiment parameters	
Dataset size	1 000 000 Playlists
Playlist id's	1017, 589 and 89
Random seed	123
Number of song recommendations	5
Playlist split percentage	0.1
Audio features	danceability, energy, loudness, speechiness, acousticness, instrumentalness, liveness, valence, tempo
Default MAIRS parameters	
Mutation rate	0.5
Clonal rate	500
Hypermutation rate	100
Stimulation threshold	0.8
Initial ARB population size	10
Affinity threshold	0.1
Recognition region radius	0.2
Mutation range	True

Table 5.3: The default parameters for all the experiments

Due to time constraints and the manual workload involved in evaluating the model, the default MAIRS parameters were primarily motivated by those in AIRS [57].

A total of three playlists were selected for the model’s evaluation: playlist 1017 piano, playlist 589 rap and playlist 89 80s. These playlists represent a wide range of music types, allowing the experiments to compare the generality of the model to some degree. The ultimate choice of including playlist 1017 was based on it containing acoustic piano songs, which were discovered during preliminary testing to give good results 5.1.3. Poor results on this playlist could indicate a too high degree of exploration introduced in the models or other design flaws. Further, playlists 589 and 89 were selected because they consist of a different degree of similar music. The rap songs in playlist 589 are considered to have more distinct similarities than playlist 89, which contained songs from the 1980s. Playlist 89 will have less strict inclusion criteria regarding the type of genres included in the playlist. Thus, playlist 89’s songs would be more diverse in terms of their audio features, making it more difficult to provide relevant song recommendations. Therefore, the model is expected to perform better in relevance on playlist 589 than on playlist 89.

5.4.1 Exp.1

Experiment 1 was conducted with the similarity measures applied to the original playlists. In other words, the whole antigen population is set as the evolved MCP. Therefore, the only parameters that changed between each sub-experiment were the parameters concerning the similarity measure set to run.

5.4.2 Exp.2

Parameter	Value
MAIRS 1.0	
Initial Memorycell Size	Playlist size
Maximum number of resources	Playlist size * Clonal rate
Recognition Region Radius	0.2

Table 5.4: Parameters and their respective value for Exp.2

5.4.3 Exp.3

Parameter	Value
MAIRS 1.0	
Initial Memorycell Size	Playlist size
Total resource size	Playlist size * Clonal rate
Recognition Region Radius	0.2
MAIRS 2.0	
Memory cell candidate size	3
Maximum number of resources	1000
Outlier distance	0.2
Recognition Region Radius	0.2

Table 5.5: The individual parameters for both models

5.4.4 Exp.4

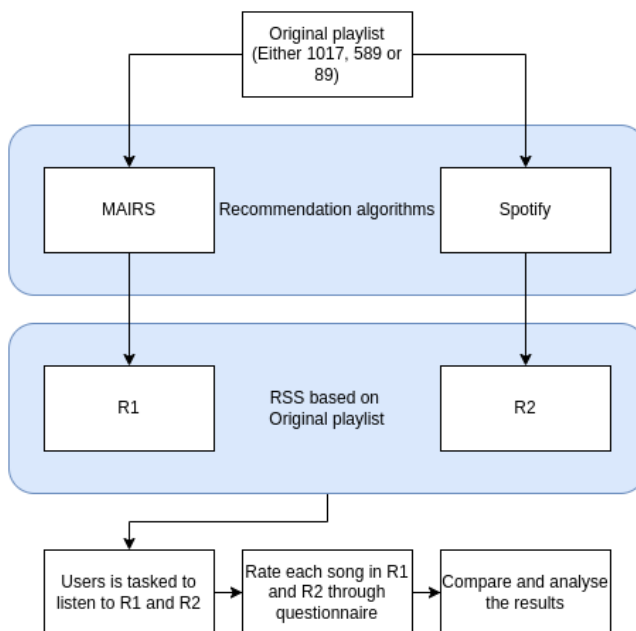


Figure 5.1: Overview of how exp.4 is conducted

Exp 4.A

The initial plan in Exp.4 was to compare Spotify's algorithm with the proposed model with the user's own playlist. By conducting tests on the user's own playlist one could ensure that the users had familiarity with the original playlist that the RSS originated from. Further, by letting the users select their own playlist, it could be assumed that their liking of their original playlist was somewhat equal among the users. This would ensure that the foundation for the results behind the liking metric was equal for all users. Unfortunately, issues with Spotify's API occurred in the later phases of the project. Possible due to a rate limit ban from the many requests sent to the API, and it was no longer possible to retrieve data regarding the users playlist. In the following section it is explained how the issue was resolved and how exp.4 finally was conducted.

Exp 4.B

Exp 4.B, from now on referred to as exp.4, was conducted in the same way as Exp.4.A with the only difference being the original playlists. Playlist 1017, 589 and 89, being in the MPD (Million Playlist Dataset), ensured that the experiment could be conducted without the use of Spotify's API. Due to the rate limit ban from Spotify, the RSS from Spotify was created by manually adding the 5 first suggested tracks in the APC (Automatic Playlist Continuation) section of the original playlist to its own single playlist, labeled as R2 in the experimental results. As seen in 5.1, R1 and R2 was then sent out to be rated by the users through the online questionnaire.

5.5 Experimental Results

5.5.1 Exp.1

The results presented in Exp.1 will investigate the effect of the different similarity measurements; Range method (RM), Average affinity (AA), Closest affinity (CA) and Recognition region (RR) (see section 4.6). The Recommended song set (RSS) is evaluated in terms of relevance and diversity. If found, other interesting facts seen in the graphs will be pointed out.

Playlist independent comparison

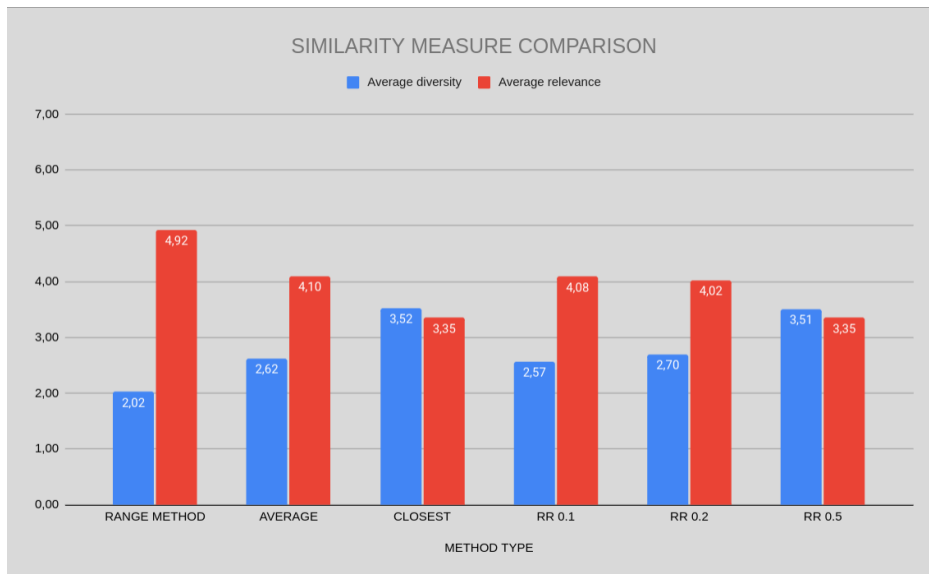


Figure 5.2: Exp.1 - Playlist independent results

The presented results in the graph consist of the average diversity and relevance scores from all three playlists based on the questions presented in table 4.3. In this experiment, it is preferable to have a lower diversity score and a higher relevance score.

Figure 5.2 supports the hypothesis regarding RM outperforming the other methods in terms of both lowest diversity and highest relevance, despite not being statistically significant. Moreover, a comparison of the results between RM and CA may indicate that the similarity measure should take into account the features of the entire playlist as opposed to the features of individual songs. The CA will recommend the songs with the lowest euclidean distance to one of the songs in the original, which the results indicate is not optimal. The validity of this claim is further supported by the results of AA, which outperforms CA. Similarly to RM, AA considers all the songs in the original playlist when classifying songs as "similar".

Playlist dependent comparison

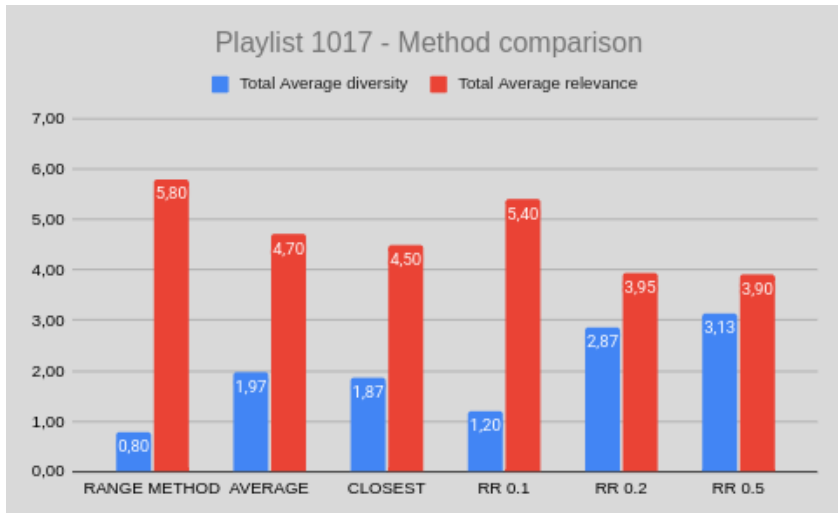


Figure 5.3: Playlist 1017 — Piano characteristics

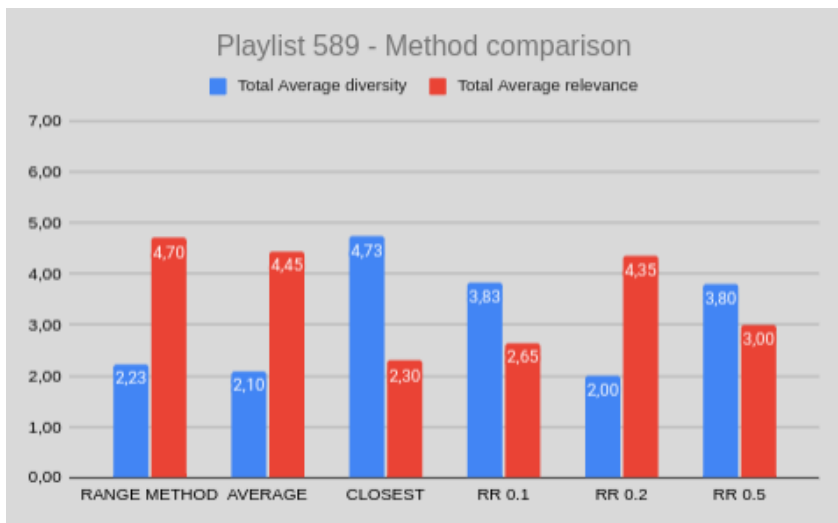


Figure 5.4: Playlist 589 — Rap characteristics

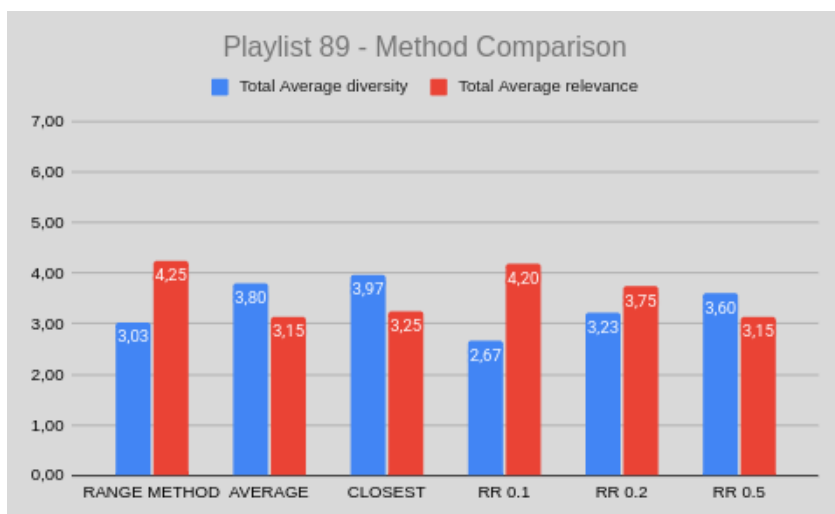


Figure 5.5: Playlist 89 — Song from 80s characteristics

Several interesting trends are revealed by comparing the similarity measures between the different playlists.

First, it appears that adjusting the radius of the recognition region according to the type of playlist affects the relevance and diversity. In figure 5.3 playlist 1017 with Piano characteristics, RR 0.1 outperforms RR 0.2 due to its lower diversity and greater relevance. In contrast, the RR 0.1 scores considerably lower in playlist 589 with Rap characteristics compared to playlists 89 and 1017. As RR 0.1 has a smaller recognition region, this may result in a stricter and smaller search coverage, lowering the total number of recognition regions RSS can fall within (votes). Thus, irrelevant music is more likely to appear in RSS since relevant songs may be overlooked in the strict search space.

On the other side, figure 5.2 illustrates that RR 0.5 has a worse score than RR 0.1 and RR 0.2. The large radius of the recognition regions in RR 0.5 may cover too much of the search space. Thus, irrelevant songs may receive votes, and some of these songs may be sufficiently voted to qualify for RSS inclusion.

The fluctuating similarity and diversity scores across the types of playlists in figure 5.3, 5.4 and 5.5 indicate that RM is also the most stable method in general. The method performs better than or on par with the other similarity measurements on all three playlists and shows robustness for a wide range of playlist

types. In addition, if the radius specified in the RR method has been overlooked and the method is generally considered, at least one RR method can be observed to perform on par with RM across three types of playlists. In that case, one might also argue that RR is also robust as a similarity measure for playlists of different types. Accordingly, RR may also be an appropriate similarity measure for APC, provided that the radius can be accurately specified for each type of playlist.

In conclusion, although not statistically significant, RM is evaluated as the best method, both on average and for each playlist type. Unfortunately, due to time constraints, it has not been possible to integrate RM as a similarity measure with MAIRS, so RR was selected instead. This is because the RR, independent of the specified radius, has shown to be able to recommend results on par with the RM. RR 0.2 was ultimately chosen as the similarity measure for MAIRS in the following experiments to encourage more diversity.

Similarity measure independent comparison

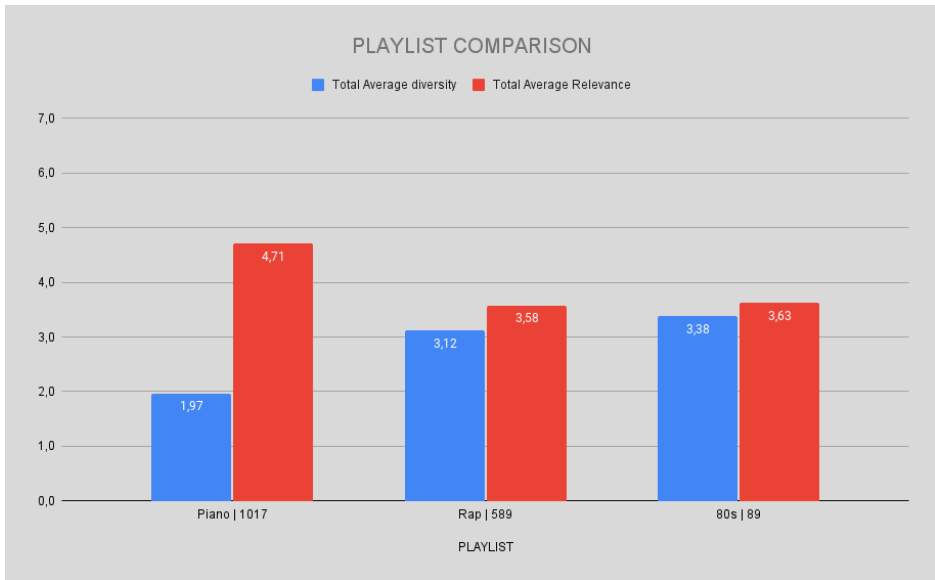


Figure 5.6: The average score of all similarity measure within the different playlists

Figure 5.6 presents the average score of diversity and relevance within the different playlists across the similarity measures. The results indicate that, as observed in the preliminary test (see section 5.1), a playlist with songs that are more similar in terms of audio features is more likely to get more successful recommendations, independent of the similarity measures. This may indicate that the playlist that are instrumental playlists, such as 1017 piano, are well represented in the features. A reason that playlist 589 rap and playlist 89 80s does not perform as well is that the "similar" data items (songs) are more scattered in the search space in comparison. Consequently, similarity measures struggle to classify the relevant songs since these songs are more likely to be farther away from the evolved memory cells (songs in the original playlist) in the search space. This hypothesis is strengthened when comparing the results for RR 0.1, RR 0.2 and RR 0.5 for playlist 1017 in figure 5.3, where RR 0.2 and RR 0.5 seems to get a lower similarity score than RR 0.1. RR 0.2 and RR 0.5 explore more of the search space, hence increasing the chances of recommending songs not entirely relevant to the playlist compared to RR 0.1.

5.5.2 Exp.2

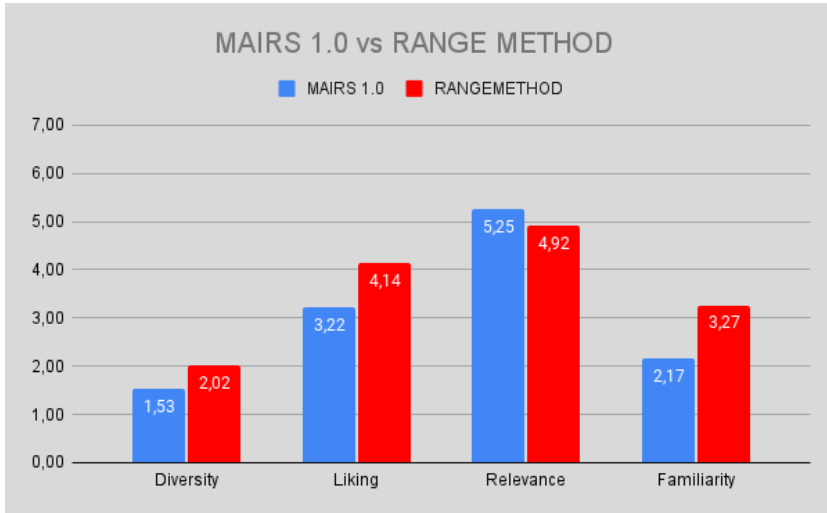


Figure 5.7: The average score of evaluation metrics of MAIRS 1.0 and RANGE METHOD

In figure 5.7, the average evaluation from MAIRS 1.0 as well as the results from RANGE METHOD (from Exp.1) are presented. It is apparent that the metrics in MAIRS 1.0 are on the lower half of the scale with the exception of relevance. According to the figure 5.7, the RANGE METHOD (RM) appears to be superior to MAIRS 1.0 in all metrics except relevance, although it is not statistically significant. There is, therefore, little evidence to support the hypothesis regarding MAIRS 1.0 outperforming RM (see table 5.1).

MAIRS 1.0 has a lower diversity metric compared to the RM in figure 5.7. This was not expected since the proposed model aimed to promote diversity among the population, which should have resulted in more musically diverse song recommendations. Thus, the diversity metric should have been higher or equal to RM, which was not the case. However, the difference is not statistically significant. This could indicate the results regarding diversity are accidental outcomes.

On the other hand, the lower diversity and higher relevance in MAIRS compared to RM would suggest that the song recommendation is more successful. In other words, the song recommendations in MAIRS 1.0 are possibly more relevant for the playlist. It can be assumed that with a more diverse MCP, song recom-

mentations will become more similar and relevant to a playlist. This interesting finding should be further investigated to determine whether the population diversity and music diversity correlate in the opposite direction.

In figure 5.7, it is also be seen that MAIRS 1.0 have a higher similarity whilst having a lower familiarity compared to RM. One may conclude that MAIRS 1.0 is able to classify serendipitous songs, as the song recommendation was less known but relevant. Consequently, MAIRS 1.0 assist the user in discovering new music. Therefore, it could be speculated that the proposed MAIRS 1.0 model solves the problem of cold starts through content-based filtering. The liking is also however compare also observed to be lower compared to RM for MAIRS 1.0, which could also indicate that the reason the liking is low is due to the songs being unknown.

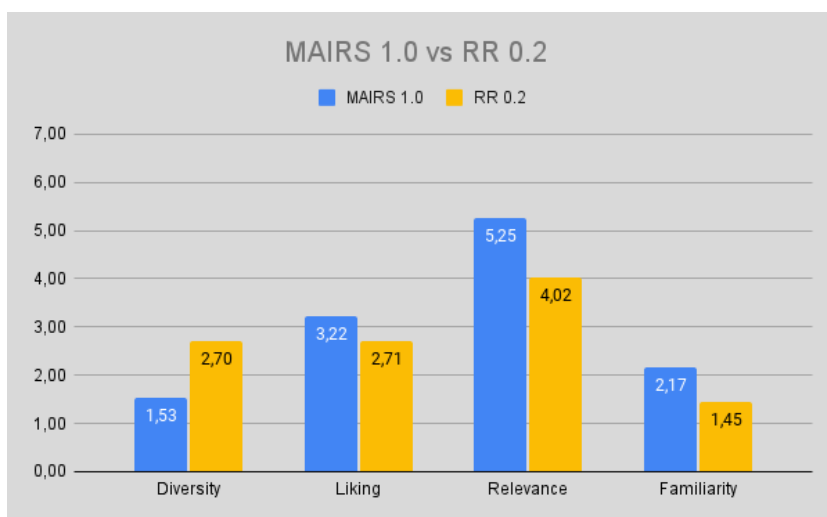


Figure 5.8: The average score of evaluation metrics of MAIRS 1.0 and RR 0.2

The results from MAIRS 1.0 and RR 0.2 (from Exp.1) are presented in figure 5.8. In comparison with RR 0.2, MAIRS 1.0 appears to perform better in all evaluation metrics except diversity. This is noteworthy since RR 0.2 was the chosen similarity measure (see section 5.5.1) in MAIRS 1.0. Therefore, with that proposed model, MAIRS 1.0 was able to create a more diverse song representation of the original playlist, to then achieve more successful song recommendations. The design decisions that were made in MAIRS 1.0 should be investigated further to identify made potential differences. Further, there is not much difference between the final MCP size between the two methods, which supports the indication even

more.

Comparison between methods in each playlist

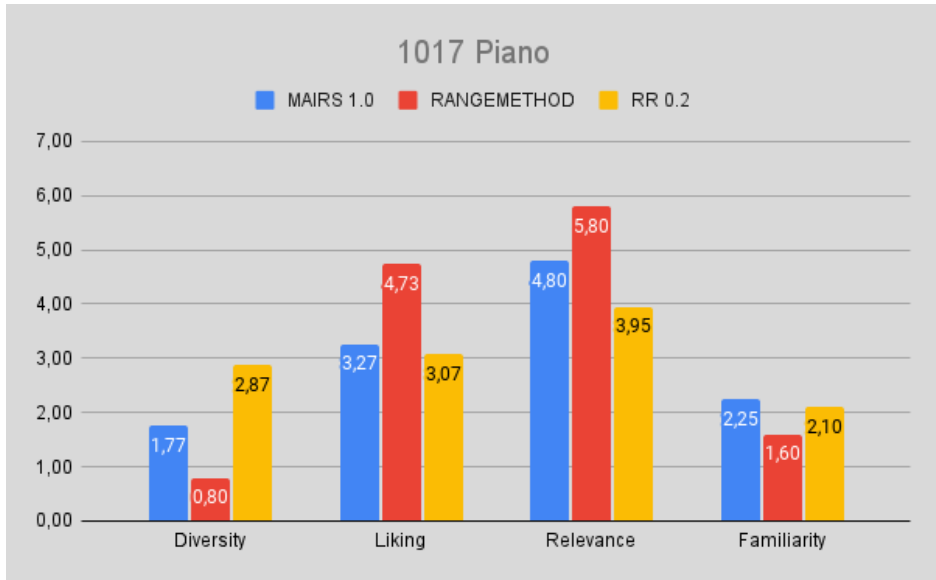


Figure 5.9: 1017 piano - The average score of evaluation metrics for MAIRS 1.0, RANGE METHOD and RR 0.2

figure 5.9 compares the MAIRS 1.0, RM and RR 0.2 in evaluation metrics for the 1017 piano playlist. It is evident that the RM performed best in terms of relevance and generated the least diverse recommendations out of the three methods. As discussed in Exp.1 5.5.1, this may be due to the features in piano-themed songs being very similar to each other, and the interval between the feature values is small. Meanwhile, MAIRS 1.0, which still had positive relevance results, had a much higher diversity in piano song recommendations. Thus, for the 1017 piano playlist, the model was able to promote diversity while retaining diversity. However, the liking is significantly lower in MAIRS 1.0 compared to the RM. This could indicate that although the song recommendations are more diverse, they do not correspond with the users' preferences.






#	TITLE	ALBUM
1	 Concerto No. 20 in D Minor for Piano a... Wolfgang Amadeus Mozart, Berliner Sympho...	Concert Series: Mozart - Concertos ...
2	 Tchaikovsky: Swan Lake, Op. 20, Act II:... Pyotr Ilyich Tchaikovsky, André Previn, Londo...	Tchaikovsky: Swan Lake, Op. 20
3	 Piano Quintet in E-Flat Major, Op. 44: I:... Robert Schumann, Alexander Melnikov, Jerus...	Schumann: Piano Quintet; Piano Qu...
4	 Stella By Starlight Morten Haxholm Quartet, Lage Lund, Morten...	Viridian
5	 Piano Concerto No. 5 in E-Flat Major, O... Ludwig van Beethoven, Evgeny Kissin, James ...	Evgeny Kissin - The Complete Conc...

Figure 5.10: MAIRS 1.0 Song Recommendations - Playlist 1017






#	TITLE	ALBUM
1	 Patience The Lumineers	Cleopatra
2	 Milk Henri Bardot	Blue Night
3	 Intro Black Elk	Sparks
4	 Habits Kenneth August	Slower Ballad Covers
5	 Earnestly Yours (feat. Ren Ford) Keaton Henson, Ren Ford	Romantic Works

Figure 5.11: Range Method Song Recommendations - Playlist 1017

According to figure 5.10 and 5.11, the majority of the song recommendations in MAIRS 1.0 were orchestral, whereas RM recommended more piano pieces. Therefore, the lower relevance in MAIRS 1.0 can be attributed to the fact that, despite being instrumental, the 1017 search space is more strict compared to other playlists. RM is able to recommend relevant songs within that strict search space due to it taking into account through playlist range.

MAIRS 1.0 exceeds RR 0.2 in terms of relevance and liking. The difference is, however, not significant here either. It could still indicate that despite promoting diversity in the population may result in lower relevance and liking, there still needs to be a degree of population diversity to achieve relevant enough results. As mentioned in Exp.1, given the limited range of features in the 1017 piano, RR 0.1 may be a better similarity measure to use with MAIRS 1.0. Therefore, for further work, the proposed model should analyse the features range in a given

playlist and select the most fitting RR according to that.

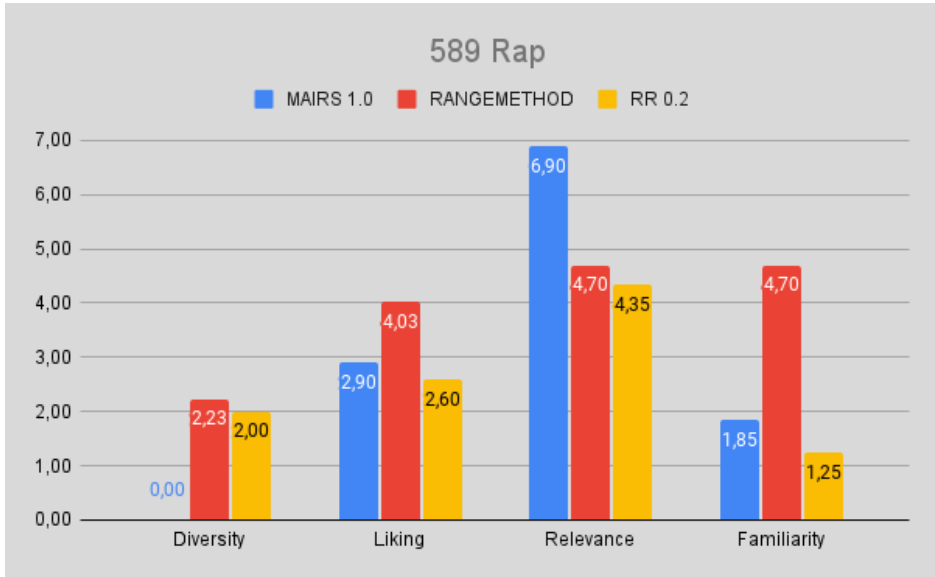


Figure 5.12: 589 rap - The average score of evaluation metrics for MAIRS 1.0, RANGE METHOD and RR 0.2

According to figure 5.12, MAIRS 1.0 outperforms both RR 0.2 and RM significantly. Perhaps this is a result of the interval in playlist range in the RM being too large and general, which targets too many diverse and less relevant songs. This gives MAIRS 1.0 an advantage, since it may have a more restrictive search space. Furthermore, this could indicate that the features in rap songs may be more diverse as opposed to piano music and that the diverse population within MAIRS 1.0 is able to reflect that through evolved memory cells.

5.5.3 Exp.3

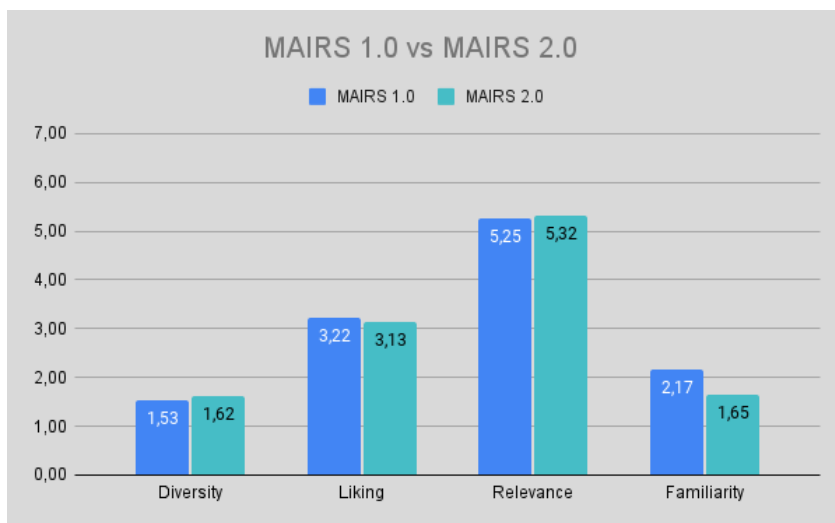


Figure 5.13: The average score of evaluation metrics of MAIRS 1.0 and MAIRS 2.0

The results of MAIRS 2.0 are presented in figure 5.13 along with those from MAIRS 1.0 (from Exp.2). The table and figure demonstrate that there are almost no differences in evaluation metrics between the two proposed models. There is, therefore, little support for the hypothesis that MAIRS 2.0 increases relevance metrics while maintaining diversity. Furthermore, the final MCP in MAIRS 2.0 was adjusted to be three times larger than the original playlist size. Results in figure 5.13 suggest that the data expansion capabilities in MAIRS 2.0 are insufficient when the original playlist is large. Additionally, a diversity metric should have been minimised because the initialisation in MAIRS 2.0 is not random. Perhaps the initialisation approach in MAIRS 2.0 has a small impact when the initial population size in MAIRS 1.0 is set as the length of the playlists.

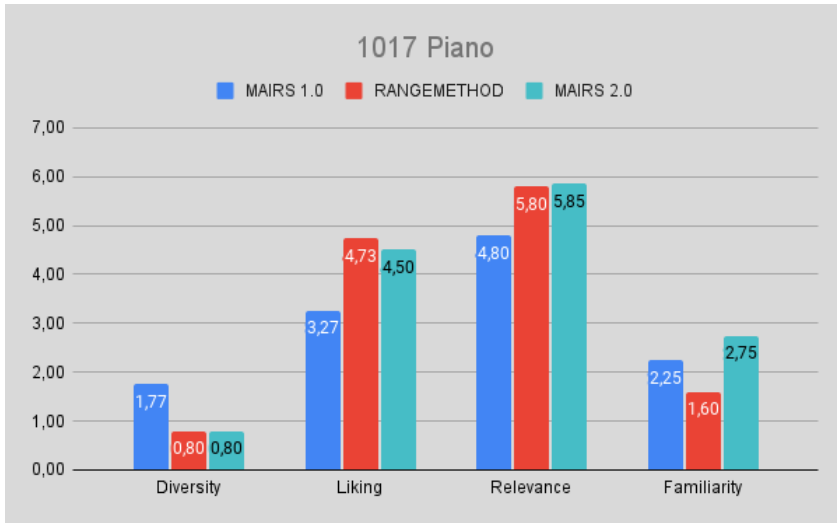


Figure 5.14: 1017 piano - The average score of evaluation metrics for MAIRS 1.0, RANGE METHOD and MAIRS 2.0

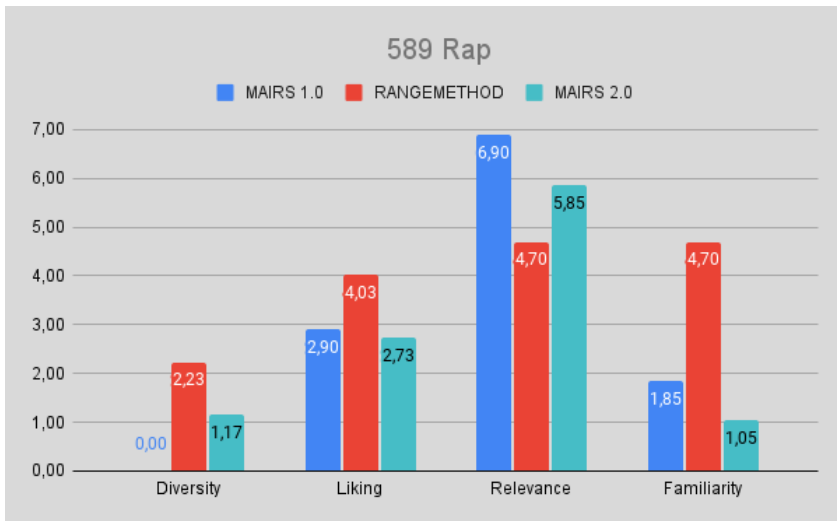


Figure 5.15: 589 rap - The average score of evaluation metrics for MAIRS 1.0, RANGE METHOD and MAIRS 2.0

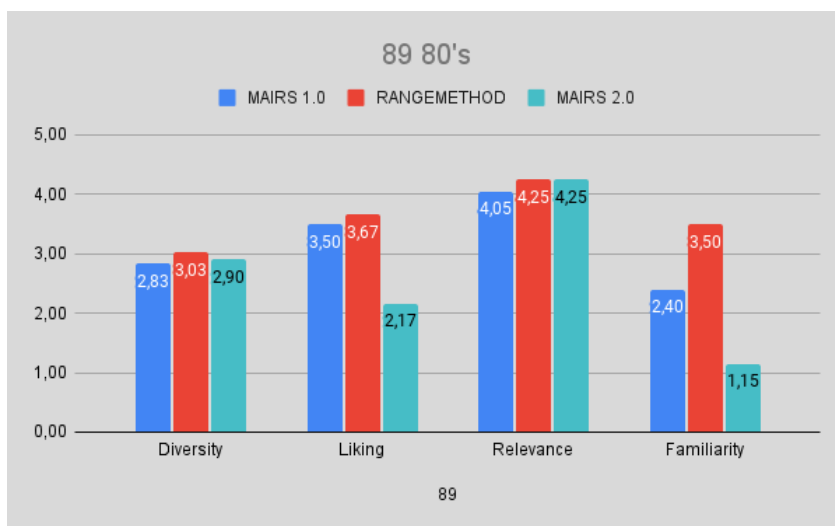


Figure 5.16: 89 80's - The average score of evaluation metrics for MAIRS 1.0, RANGE METHOD and MAIRS 2.0

The differences between the models become more apparent when taking a look at the results of each playlist type, as seen in figure 5.14, 5.15 and 5.16. MAIRS 1.0 outperforms MAIRS 2.0 in playlist 589 rap in terms of relevance. Thus, it is apparent that MAIRS 2.0 introduces more musical diversity in the recommendations. The situation is reversed when it comes to the 1017 piano playlist.

By comparing the range method to each model, however, a difference becomes apparent. As shown in figure 5.14 and 5.15, MAIRS 2.0 beat RM in playlist 589 rap and is able to match it in playlist 1017 piano in terms of relevance. MAIRS 1.0, on the other hand, beats it significantly in the 589 rap playlist and loses to it in playlist 1017 piano. The relevance metric in playlist 89 80s is almost identical in all methods. The reason MAIRS 2.0 was able to defeat MAIRS 1.0 in playlist 1017 was may due to the population being a bit less diverse, yet diverse enough to defeat RM in playlist 589. Therefore, it would suggest that the level of diversity in the population is important to consider when developing a model that takes all types of playlists into account.

Another important point is that the proposed models are still struggling with the 89 80s playlist, as seen in figure 5.16, which consists of songs with varied genres. This kind of playlist may be representative of an average user's playlist. The individual may compile a playlist of the songs that they enjoy, which may

include different genres and characteristics. Further work should there focus on playlists that have the same characteristics as the 89 80s playlist.

One may conclude that MAIRS 2.0 outperforms the other models in general by at least matching or beating RM in the relevance metric across all playlist types. It is also suspected that MAIRS 1.0 is overfitted to playlists that employ the same characteristics as the 589 Rap playlist. Therefore, the chosen model for Exp.4 is MAIRS 2.0.

5.5.4 Exp.4

The final experiment aims to compare the final model with Spotify’s recommendation algorithm, based on a user questionnaire (see section 4.3) conducted on 12 people. The final proposed model, MAIRS 2.0, is constructed based on the results obtained in previous experiments. The results presented are based on diversity, liking, relevance and familiarity. In contrast to the previous experiment, this experiment evaluates the performance of the systems in relation to how they perform as music recommender systems. Hence the evaluation metrics are looked at in relation to each other and not as independent of each other as in the previous experiments.

Playlist dependent comparison

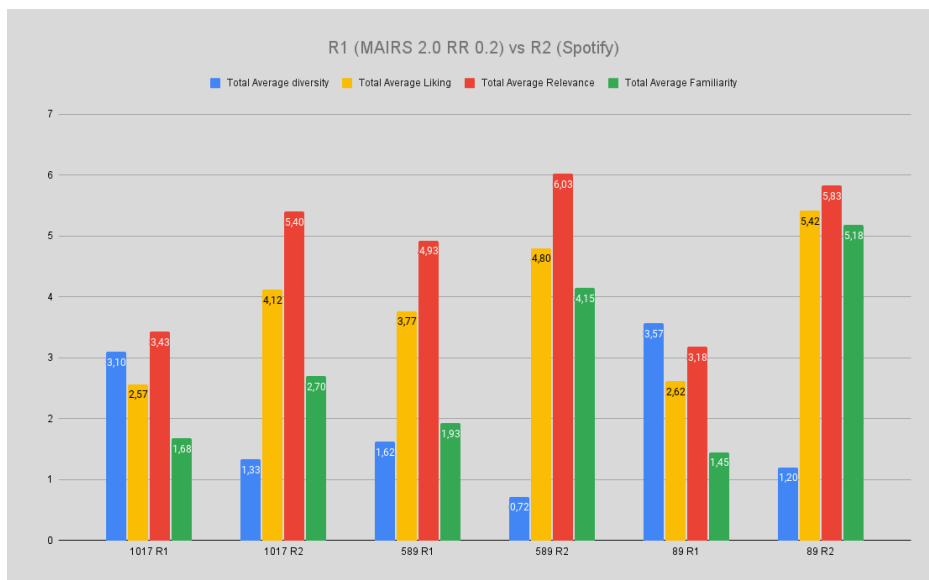


Figure 5.17: Playlist dependent comparison of MARIS 2.0 and Spotify’s algorithm

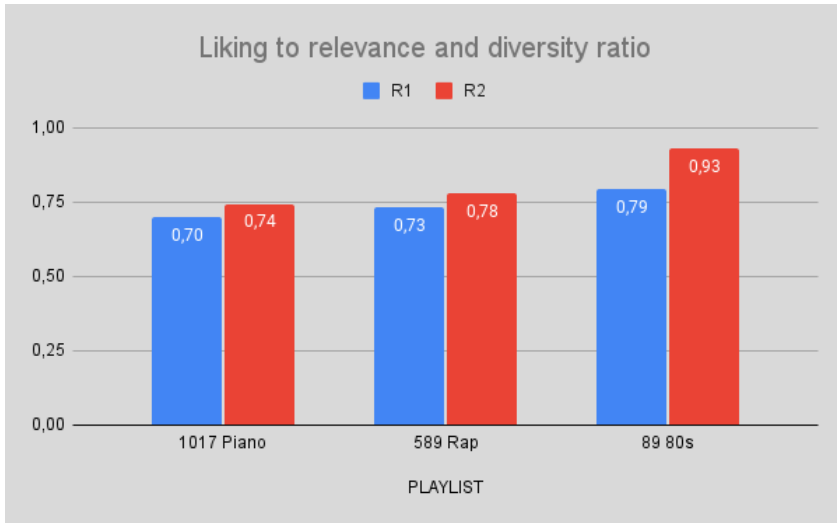


Figure 5.18: Liking in relation to diversity and relevance in the RSS

Several interesting trends are observed in the figure 5.17. R1 refers to the song recommendation provided by MAIRS 2.0, while R2 refers to the song recommendation provided by Spotify. As elaborated in section 5.4, playlist 1017 was selected for evaluation of the proposed model because the RSS seemed more relevant than the RSS of other playlists. Therefore, playlist 1017 was expected to exhibit less diversity and more relevance than playlists 589 and 89.

Surprisingly, the results from figure 5.17 show that 589 R1 recommends far more relevant songs compared to the other two playlists. In fact, playlist 1017 scored almost on par with playlist 89. However, as elaborated in exp.3, the RSS from R1 in playlist 1017 consisted of a large amount of orchestral music. Considering MAIRS 2.0 is a pure content-based system and should encourage diversity, these results might not be as surprising as they seem. The nature of orchestral music is acoustic and instrumental, similar to the nature of piano music. Therefore, one could argue that the song recommendations are still relevant to the playlist but promote more diversity than R2. Furthermore, orchestral music does not represent a completely different genre or type of music from piano music.

Playlist 89 also yields interesting results. As mentioned previously, playlist 89 represents music from the 1980s and may have a broader range of music than the other two playlists. The diversity in the playlist increases the challenge for content-based filtering systems, which is prevalent in figure 5.17, where R2 out-

performs R1 in relevance and diversity. This is possibly due to the fact that Spotify uses other techniques such as collaborative filtering and metadata to identify the type of music in this playlist. Collaborative filtering and metadata would, for instance, be particularly beneficial for playlist 89 since songs from the 1980s are not determined by the audio features but rather the decade in which they were created. The high familiarity and liking score for 89 R2 further strengthens that collaborative filtering was one of the main components of the RSS, as this technique is known to recommend more popular and familiar songs.

In figure 5.17, it can be seen that a higher relevance and lower diversity in the RSS result in a higher liking factor, which can be seen for both 1017 R1 and 89 R1. One of the overall goals of MAIRS 2.0 was to recommend more diverse music but with a similar style and mood to the songs in the original playlist. An effective measure of whether the diversity introduced in the RSS has been successful is comparing the average liking of a playlist with the diversity and relevance score. Figure 5.18 shows this ratio. In this figure, a higher score is evaluated as better. A score of 1 means that the average diversity and relevance of the song is the same as the liking score. Ideally, the proposed model would outperform R2 in all three playlists, which was not the case. However, in terms of promoting diversity in the RSS, the results from figure 5.18 show that R1, at least for playlists 1017 and 589, performs on par with R2. This would indicate that R1 for some playlists can introduce diversity while perceiving the same relation between liking and diversity.

Playlist independent comparison

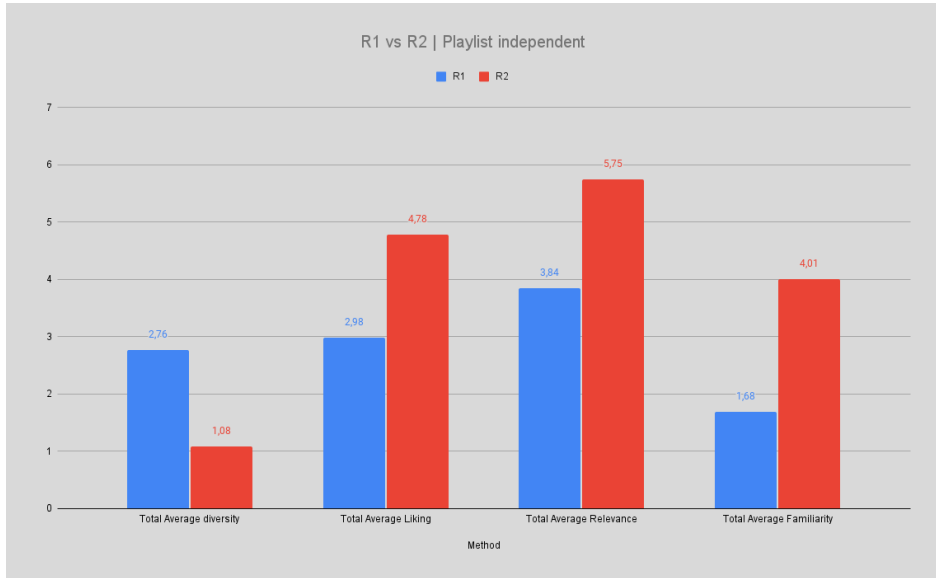


Figure 5.19: Playlist independent comparison of MAIRS 2.0 and Spotify’s algorithm

Figure 5.19 shows the playlist independent scores of the two methods. The difference between R2 and R1 was statistically significant in all the evaluation metrics. It was expected that R1 would have higher relevance and lower diversity than R2, as MAIRS 2.0 was designed to encourage diversity in the RSS. The higher liking and familiarity seen at R2 could be due to collaborative filtering, as this method favours popular music.

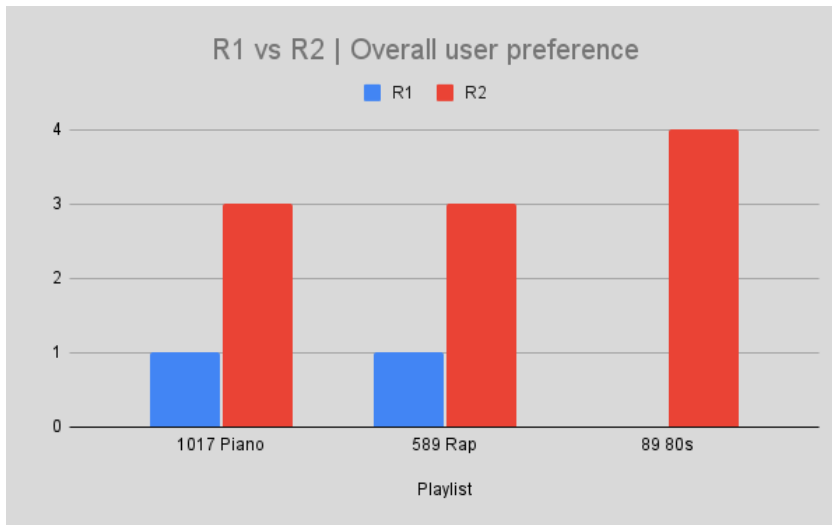


Figure 5.20: Overall user preference between R1 and R2

The users were also asked which of the RSS they preferred as a final question in the questionnaire. Figure 5.20 presents these preferences. In relation to the average high liking score seen for R2 in figure 5.19, it was expected that the users mostly preferred R2. However, it was a surprise to see that some users preferred MAIRS 2.0's RSS over Spotify's, given the number of resources that Spotify has available compared to MAIRS 2.0. This indicates that the proposed model has the potential to function as a music recommendation system, and it should be further explored in light of the results found during the study.

Chapter 6

Evaluation and Conclusion

This chapter discusses the results in light of the research questions in section 6.1. In section 6.2 the limitations are presented followed by section 6.3 showing the contributions from the project. Finally, future work is presented in section 6.4.

6.1 Discussion and goal evaluation

In this thesis, the overall objective is to *investigate the applicability of an Artificial Immune System (AIS) with content-based filtering (CBF) for Automatic Playlist Continuation (APC) task with limited features*. Four separate research questions were developed to address different aspects of this overall goal. These research questions are then discussed in light of the results obtained through testing the proposed model.

Research question 1 *What similarity measure should be applied to ensure that songs in a playlist can be classified as 'similar' despite a limited number of features in the representation?*

The results suggested that the Range Method was the most effective method for classifying the most "similar" songs to a given playlist. It was observed both when looking at the average similarity metric across playlists as well as when examining each playlist type individually. This may indicate that the similarity measure should consider the features of the playlist as a whole rather than the features of each individual song. Accordingly, a song does not necessarily need to have a low Euclidean distance with the songs in a playlist to qualify as "similar". Instead, it should fall within most of the playlist's feature range. The result, however, does not appear to be statistically significant, indicating that it might be a fluke. Furthermore, it is also possible that the RM is more effective for the

type of playlists selected for testing.

Similarity measures differed in performance depending on the playlist type, indicating that similarity measure is playlist-dependent. Furthermore, all similarity measures had greater difficulty classifying "similar" songs to playlists without distinct characteristics. For instance, the 89 80's playlist contained songs with a wide range of genres and more variation in feature values.

Time constraints prevented a way applying the range method as a similarity measure for the proposed model was not found. Thus, further work should investigate this possibility. A similar approach was applied to the mutation range with the proposed model as compensation. Due to its almost equal performance to the range method, RR 0.2 was the similarity measure used in the proposed model.

Research question 2 *How should AIRS be refined for content-based filtering of Automatic Playlist Continuation while taking music diversity into account?*

The intention was to exploit the properties of AIRS to promote diversity within the population, which in turn promotes music diversity in recommendations. Furthermore, the proposed model was continuously tailored to meet the needs of APC through architectural design choices. The proposed model is called MAIRS 1.0. To encourage diversity, the original AIRS model was converted into a one-classifier not to restrict self-space. Additionally, the default mutation rate was increased to provide a greater degree of diversity.

The results demonstrate that the diverse population of evolved MCPs encourages music diversity in some playlists, such as 1017 piano, despite trading off relevance. However, MAIRS 1.0 outperforms RM significantly in playlist 89 rap in terms of relevance and providing no music diversity. This may indicate that the design decisions made in MAIRS 1.0 positively influence similarity and relevance despite the intention to promote music diversity. Rap playlists may also contain more diverse features than piano playlists, so MAIRS 1.0 can do well on playlists with features that vary slightly. Comparing the results with the RR 0.2 applied to antigens as evolved MCP, it can be seen that MAIRS 1.0 properties are in fact favourable, as diverse populations are able to achieve more successful song recommendations. This implies that it is necessary to investigate the design decisions made in MAIRS 1.0 to identify any potential influence they may have on music diversity and relevance.

One may also conclude that MAIRS 1.0 can encourage music discovery as it is able to recommend serendipitous songs. The results also suggest that the pro-

posed model can alleviate the cold start problem by implementing content-based filtering.

Research question 3 *How can similarity be encouraged while maintaining diversity in the proposed model?*

To increase similarity, several design choices were made for MAIRS 2.0. The results indicate, however, that there is not a considerable difference between MAIRS 1.0 and MAIRS 2.0 in terms of performance. This would suggest that the design choices may not have been advantageous.

One of them was to initialise the antigens as memory cells. This was to take advantage of all antigens in the population, as the AGP is initially small. Furthermore, the diversity metric would minimise since MAIRS 2.0 does not use a random initialisation process. The results may indicate that this would not matter if the initial population in MAIRS 1.0 is set as the size of the AGP. The evolved MCP in MAIRS 2.0 was also adjusted to be three times larger than the AGP. However, MAIRS 2.0 does not appear to be able to produce a representative expanded MCP when the original playlist is quite large.

On the other hand, it could be argued that MAIRS 2.0 still outperforms MAIRS 1.0 in similarity when compared with RM. When comparing the relevance metrics for all playlist types, MAIRS 2.0 matches or outperforms RM, while MAIRS 1.0 defeated RM significantly in only one of them. MAIRS 2.0 also outperformed MAIRS 1.0 in playlist 1017 piano in terms of relevance. This is likely a result of MAIRS 2.0 having a less diverse population, but still diverse enough to defeat the RM in the same playlist. The level of diversity in the population should, therefore, be considered when developing a model that considers all types of playlists.

Research question 4 *How can the AIS model ensure similarity whilst achieving music diversity in the recommendations?*

Compared to Spotify, MAIRS 2.0 was able to introduce more diversity to song recommendations. Considering that the objective was to introduce diversity while ensuring similarity, it is a positive development. It can be concluded that MAIRS 2.0 promoted more diversity due to its design choices. Even so, there is still a need to achieve a greater degree of similarity and relevance. This may be accomplished by incorporating collaborative filtering, making it hybrid, and including features such as metadata. Moreover, the proposed model should have been tailored to

the type of playlist rather than having a general one, since the performance of similarity differs between playlist types.

There is also a need to increase the familiarity with the recommended songs. The reason for this is that it appears that users will be more inclined to favour a song recommendation if they are familiar with it. Incorporating collaborative filtering and metadata would also result in more familiar song recommendations. The proposed model, however, still contributes to music discovery and alleviates the cold start problem. Since the proposed model is content-based, this is to be expected.

Despite the various shortcoming of the proposed model, MAIRS 2.0 still has the potential as a music recommendation system based on the findings presented. The main focus of future work should therefore be on increasing similarity and relevance to be competitive with the current state of the art.

6.2 Limitations

The project was affected by two unforeseen events such that adjustments were necessary to ensure sufficient progress.

6.2.1 Spotify rate limiting

During the final phases of the project, an issue with requesting data from Spotify API was encountered. Possibly due to a rate limit ban, requests to Spotify's API regarding automatically creating playlists from the RSS and retrieving the user's playlist items were rejected. This resulted in the final experiment changing from recommending songs on the user's playlist to playlists existing in the dataset. The fact that the RSS no longer automatically could be created by the algorithm also slowed down the project as more work had to be targeted towards manually creating and labelling the playlist concerning the tests conducted.

6.2.2 Evaluation metrics

The initial plan was to evaluate the different models by measuring their classification accuracy and participating in the AICrowd challenge conducted by Spotify. An early investigation of the AICrowd challenge revealed that recommendations on many playlists were required to submit a valid entry. Thus, MAIRS was modified early on by converting parts of the code into Cython, a C implementation of Python designed to achieve the speed of C. Unfortunately, the part of the algorithm required for generating the RSS was underestimated and not optimised in

Cython. This resulted in that, except for the range method, recommendations for a single playlist usually took up to 1-2 hours. The AICrowd challenge required RSS for a total of 10 000 playlists for each submission.

6.3 Contributions

Some valuable contributions to the field were made during the implementation and testing of the proposed model.

The main contribution was the investigation into adapting an AIS with content-based filtering for the task of Automatic Playlist. As the initial version of the model did not meet the expectations in terms of similarity, a second alternative for the proposed model was also developed. There was, however, significant uncertainty associated with both the model design and the performance due to time constraints and a lack of experiments. Despite not appearing to be comparable with state-of-the-art models, the results indicate that there may be some potential in this application. In this regard, the proposed future work should be investigated further in addition to overcoming the limitations encountered during this study.

The second contribution of this thesis is comparing the ability of different similarity measures to recommend songs similar to an original playlist. The range method and recognition region have been specially tailored to the domain of music in order to test their applicability in assessing which songs fit into original playlists. The results indicate that the Range method, as well as the recognition region with the correct radius, are methods that should be further investigated when attempting to create automatic playlist continuations.

It has been shown that a limited amount of audio features in some playlists is enough information for expressing the type of music in a playlist, giving any model a correct foundation for creating RSS. This insight will likely prove helpful for systems aiming to accelerate RSS creation in an AIS model to avoid the curse of dimensionality.

Also, a contribution has been made to the field of music by creating an online questionnaire to extract users' ratings regarding their liking, familiarity, diversity, and relevancy for recommended songs. Making this questionnaire a foundation for evaluating these metrics in similar projects.

6.4 Future Work

In combination with the broadness of the topic and limitations faced during the project, many potential implementations, design decisions and evaluation metrics had to be discarded. The following subsections can serve as a foundation for further investigation in the case of interest in the topic.

6.4.1 Offline evaluation and optimisation

The current way of evaluating the proposed model was ineffective and susceptible to inaccurate and subjective results. However, time restrictions regarding delivery date and slow running times required the project to focus on other aspects. Thus, for future work, it would be interesting to see how the proposed model performs in comparison to other models and evaluation metrics other than those conducted in this project. Initially, it was planned to evaluate the final model by participating in the AICrowd challenge [59] for the employed dataset. Hence optimising the algorithm by either rewriting it in a faster language or employing techniques to run the algorithm on more suited hardware as a GPU should give the model a fair chance to run fast enough to participate in the contest.

6.4.2 Audio feature expansion

The MPD dataset contains nine audio features, which is quite a few compared to the field of content-based filtering. The lack of audio features led to the hypothesis that the characteristic of different playlist types could not be accurately described. As elaborated in chapter 5, the proposed model could more accurately predict relevant songs when the playlist type had a higher relation to the audio features used in the model. Hence combining the current model with another dataset with more audio features or using techniques to extract more audio features from the songs in the dataset could serve as additional work to the project.

6.4.3 Parameter tuning to music

Due to the difficulty in getting objective results for the proposed model, it was too time-consuming to investigate how different parameters in the AIS part of the model affected the results in addition to the planned experiments. As part of future work on the project, it would be interesting to investigate the impact of tuning these parameters specifically on the domain of music and within specific types of music playlists.

6.4.4 Investigation of other design decisions

Collaborative filtering, a standard method for recommending songs in recommendation systems, was ultimately excluded from the project. The reason is that it would be time-consuming and not contribute enough to the field. The results of the experiments and the state-of-the-art indicate that implementing content-based filtering along with collaborative filtering could improve results in some cases. Collaborative filtering could thus be a valuable addition to the model in a future project.

Further, the experiments showed that the specified recognition radius of the RR similarity measure impacted the quality of the results of the RSS. This would indicate that it could be interesting to investigate the applicability of creating a dynamic recognition radius regarding the type of the original playlist. Further work should also explore how the range method can be used as a similarity measure for the proposed model since this method appeared to be the most suitable for APC.

Bibliography

- [1] Adiyansjah, Gunawan, A. A. S., and Suhartono, D. (2019). Music recommender system based on genre using convolutional recurrent neural networks. *Procedia Computer Science*, 157:99–109.
- [2] Aggarwal, C. C. (2016). *Recommender Systems*. Springer International Publishing.
- [3] Atlas, A. (2022). 7digital - b2b music solutions for fitness, social, gaming brands. <https://www.7digital.com/>.
- [4] Barrington, L., Oda, R., and Lanckriet, G. R. (2009). Smarter than genius? human evaluation of music recommender systems. In *ISMIR*, volume 9, pages 357–362.
- [5] 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.
- [6] Bertin-Mahieux, T., Ellis, D. P. W., Whitman, B., and Lamere, P. (2011). The million song dataset. pages 591–596.
- [7] Bogdanov, D., Haro, M., Fuhrmann, F., Xambo, A., Gomez, E., and Herrera, P. (2013). Semantic audio content-based music recommendation and visualization based on user preference examples. *Information Processing Management*, 49(1):13–33.
- [8] Celma, Ò. and Herrera, P. (2008). A new approach to evaluating novel recommendations. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 179–186.
- [9] Chen, C.-W., Lamere, P., Schedl, M., and Zamani, H. (2018). Recsys challenge 2018: automatic music playlist continuation. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 527–528. ACM.

- [10] Dagdia, Z. C. and Mirchev, M. (2020). Chapter 15 - when evolutionary computing meets astro- and geoinformatics. In *Knowledge Discovery in Big Data from Astronomy and Earth Observation*, pages 283–306. Elsevier.
- [11] de Castro, L. and Von Zuben, F. (2002). Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation*, 6(3):239–251.
- [12] De Jong, K. A. (1975). *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, USA. AAI7609381.
- [13] Defferrard, M., Benzi, K., Vandergheynst, P., and Bresson, X. (2017). Fma: A dataset for music analysis. *arXiv:1612.01840 [cs]*. arXiv: 1612.01840.
- [14] Deldjoo, Y., Schedl, M., and Knees, P. (2021). Content-driven music recommendation: Evolution, state of the art, and challenges. *arXiv:2107.11803 [cs]*. arXiv: 2107.11803.
- [15] Dionisios N. Sotiropoulos, G. A. T. (2018). Artificial immune system-based music recommendation. *Intelligent Decision Technologies*, 12(2):213–229.
- [16] Dudek, G. (2012). An artificial immune system for classification with local feature selection. *IEEE Transactions on Evolutionary Computation*, 16(6):847–860.
- [17] Eiben, A. and Smith, J. (2015). *Introduction to Evolutionary Computing*. Natural Computing Series. Springer Berlin Heidelberg.
- [18] Floreano, D. and Mattiussi, C. (2008). *Bio-Inspired Artificial Intelligence. Theories, Methods, and Technologies*. Massachusetts Institute of Technology.
- [19] Forrest, S., Perelson, A., Allen, L., and Cherukuri, R. (1994). Self-nonspecific discrimination in a computer. In *Proceedings of 1994 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 202–212. IEEE Comput. Soc. Press.
- [20] Giatzitzoglou, D. G., Sotiropoulos, D. N., and Tsihrantzis, G. A. (2019). Airs-x: An extension to the original artificial immune recognition learning algorithm. In *2019 International Conference on Computer, Information and Telecommunication Systems (CITS)*, pages 1–5.
- [21] Haktanirlar Ulutas, B. and Kulturel-Konak, S. (2011). A review of clonal selection algorithm and its applications. *Artificial Intelligence Review*, 36(2):117–138.

- [22] Hart, E. (2005). Not all balls are round: An investigation of alternative recognition-region shapes. In *Proceedings of the 4th International Conference on Artificial Immune Systems, ICARIS'05*, pages 29–42, Berlin, Heidelberg. Springer-Verlag.
- [23] Hofmeyr, S. A. and Forrest, S. (2000). Architecture for an Artificial Immune System. *Evolutionary Computation*, 8(4):443–473.
- [24] Isinkaye, F. O., Folajimi, Y. O., and Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, 16(3):261–273.
- [25] Jenhani, I. and Elouedi, Z. (2014). Re-visiting the artificial immune recognition system: a survey and an improved version. *Artificial Intelligence Review*, 42(4):821–833.
- [26] Jenhani, I. and Elouedi, Z. (2017). Airs-ga: A hybrid deterministic classifier based on artificial immune recognition system and genetic algorithm. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–7.
- [27] Ji, Z. and Dasgupta, D. (2004). Real-valued negative selection algorithm with variable-sized detectors. In *In LNCS 3102, Proceedings of GECCO*, pages 287–298. Springer-Verlag.
- [28] Kamehkhosh, I. and Jannach, D. (2017). User perception of next-track music recommendations. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, pages 113–121, Bratislava Slovakia. ACM.
- [29] Kaminskis, M. and Bridge, D. (2016). Diversity, serendipity, novelty, and coverage: A survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Trans. Interact. Intell. Syst.*, 7(1).
- [30] Karpov, P., Squillero, G., and Tonda, A. (2018). Valis: an evolutionary classification algorithm. *Genetic Programming and Evolvable Machines*, 19(3):453–471.
- [31] Kumar, B. V., Connors, T. J., and Farber, D. L. (2018). Human t cell development, localization, and function throughout life. *Immunity*, 48(2):202–213.
- [32] Kunaver, M. and PoÅ¾rl, T. (2017). Diversity in recommender systems - a survey. 123:154–162.
- [33] Lee, J. H. (2011). How similar is too similar?: Exploring usersâ perceptions of similarity in playlist evaluation. *Poster Session*, page 6.

- [34] McEwan, C. and Hart, E. (2009). Representation in the (artificial) immune system. *Journal of Mathematical Modelling and Algorithms*, 8(2):125–149.
- [35] McFee, B., Barrington, L., and Lanckriet, G. (2012). Learning content similarity for music recommendation. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(8):2207–2218.
- [36] Nebili, W., Farou, B., Kouahla, Z., and Seridi, H. (2021). Revised artificial immune recognition system. *IEEE Access*, 9:167477–167488.
- [37] Nina Duong and Ola Kruge (2022). Source code ais music recommendation. <https://gitlab.stud.idi.ntnu.no/olajkr/prosjektoppgave>,.
- [38] Ozsen, S. and Yucelbas, C. (2015). On the evolution of ellipsoidal recognition regions in artificial immune systems. *Applied Soft Computing*, 31:210–222.
- [39] Petrowski, A. (1996). A clearing procedure as a niching method for genetic algorithms. In *Proceedings of IEEE International Conference on Evolutionary Computation*, pages 798–803.
- [40] Pu, P., Chen, L., and Hu, R. (2012). Evaluating recommender systems from the user’s perspective: Survey of the state of the art. *User Modeling and User-Adapted Interaction*, 22.
- [41] Read, M., Andrews, P. S., and Timmis, J. (2012). *An Introduction to Artificial Immune Systems*, pages 1575–1597. Springer Berlin Heidelberg.
- [42] Ricci, F., Rokach, L., and Shapira, B. (2011). Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer.
- [43] Robinson, K., Brown, D., and Schedl, M. (2020). User insights on diversity in music recommendation lists. page 8.
- [44] Sareni, B. and Krahenbuhl, L. (1998). Fitness sharing and niching methods revisited. *IEEE Transactions on Evolutionary Computation*, 2(3):97–106.
- [45] Schrepp, M., Hinderks, A., and Thomaschewski, J. (2017). Design and evaluation of a short version of the user experience questionnaire (ueq-s). *International Journal of Interactive Multimedia and Artificial Intelligence*, 4:103.
- [46] Seliya, N., Abdollah Zadeh, A., and Khoshgoftaar, T. M. (2021). A literature review on one-class classification and its potential applications in big data. *Journal of Big Data*, 8(1):122.
- [47] Serapiao, A. B. S., Mendes, J. R. P., and Miura, K. (2007). Artificial immune systems for classification of petroleum well drilling operations. In *Artificial Immune Systems*. Springer.

- [48] Shao, B., Wang, D., Li, T., and Ogihara, M. (2009). Music recommendation based on acoustic features and user access patterns. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(8):1602–1611.
- [49] Sharma, A. and Sharma, D. (2011). Clonal selection algorithm for classification. In Lio, P., Nicosia, G., and Stibor, T., editors, *Artificial Immune Systems*, pages 361–370. Springer Berlin Heidelberg.
- [50] Silveira, T., Zhang, M., Lin, X., Liu, Y., and Ma, S. (2019). How good your recommender system is? a survey on evaluations in recommendation. *International Journal of Machine Learning and Cybernetics*, 10(5):813–831.
- [51] Soleymani, M., Aljanaki, A., Wiering, F., and Veltkamp, R. C. (2015). Content-based music recommendation using underlying music preference structure. In *2015 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE.
- [52] Squillero, G. and Tonda, A. (2016). Divergence of character and premature convergence: A survey of methodologies for promoting diversity in evolutionary optimization. *Information Sciences*, 329:782–799.
- [53] Sverdrup-Thygeson, S. (2021). An artificial immune system for fake news classification. Accepted: 2022-01-13T18:19:37Z.
- [54] Swearingen, K. and Sinha, R. (2002). Interaction design for recommender systems. In *In Designing Interactive Systems 2002*. ACM. Press.
- [55] Timmis, J. (2007). Artificial immune systemsâtoday and tomorrow. *Natural Computing*, 6(1):1–18.
- [Turrin et al.] Turrin, R., Quadrana, M., Condorelli, A., Pagano, R., and Cremonesi, P. 30music listening and playlists dataset. page 2.
- [57] Watkins, A., Timmis, J., and Boggess, L. (2004). Artificial immune recognition system (airs): An immune-inspired supervised learning algorithm. *Genetic Programming and Evolvable Machines*, 5(3):291–317.
- [58] Xu, L., Chow, M.-Y., Timmis, J., and Taylor, L. S. (2007). Power distribution outage cause identification with imbalanced data using artificial immune recognition system (airs) algorithm. *IEEE Transactions on Power Systems*, 22(1):198–204.
- [59] Yoogottam Khandelwal, Shivam Khandelwal, S. M. (2020). Spotify million playlist dataset challenge. <https://www.aicrowd.com/challenges/spotify-million-playlist-dataset-challenge>.

- [60] Yoshii, K., Goto, M., Komatani, K., Ogata, T., and Okuno, H. G. (2006). Hybrid collaborative and content-based music recommendation using probabilistic model with latent user preferences. In *ISMIR*, volume 6, pages 296–301.

Appendices

