Lise Jakobsen & Anna Johanne Holden Jacobsen

Modeling the Impact of Recommendation Diversity on Misinformation Spread in Social Networks: An Agent-Based Approach

Master Thesis in Computer Science
Supervisor: Özlem Özgöbek
Co-supervisor: Önder Gürcan
June 2025

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

Abstract

Social media has become society's main source of information, and the threat of online misinformation is greater than ever. Recently, the role of recommendation algorithms in spreading misinformation through social networks has become more central in misinformation mitigation research. This thesis presents an agent-based model implemented with Mesa, a Python framework, to simulate the impact of different recommendation algorithms on misinformation propagation. The SEIS epidemiological model is used to capture user states over time. The algorithms evaluated in this work include popularity-based, collaborative filtering, content-based filtering, and random. Increasing diversity in recommendations has been hypothesized to break up echo chambers and limit misinformation spread. This thesis explores this statement by also testing how increased diversity, using the post-processing approach Maximal Marginal Relevance (MMR), impacts the different recommenders' effect on misinformation spread. The interplay between diversity and echo chambers is also studied. The results of the experiments show that popularity-based algorithms have the worst impact on misinformation propagation and often amplify misinformation in recommendations. Algorithms centralized around item similarity, such as content-based filtering and item-based collaborative filtering, showed the most efficiency in limiting the spread of misinformation. Increasing diversity in recommendations using MMR is more beneficial for some algorithms than for others. For popularity-based algorithms, the increase in diversity resulted in less amplification of misinformation, although propagation stayed the same. Although item-based collaborative filtering and content-based filtering both exhibited slightly higher amplification of misinformation exposure, they simultaneously decreased the propagation of misinformation. User-based collaborative filtering remained stable with and without increased diversity and performed similarly to the baseline algorithm, random. In general, the increase in diversity shows minimal change in the propagation and exposure of misinformation. However, increasing diversity results in a decreased echo chamber effect across all algorithms, indicating that although diversity alone cannot mitigate misinformation, it can help break up echo chambers.

Sammendrag

Sosiale medier har blitt den viktigste kilden til nyheter og informasjon, og trusselen om feilinformasjon på nettet er større enn noensinne. I den forbindelse forskes det stadig mer på hvordan anbefalingsalgoritmer bidrar til å spre feilinformasjon gjennom sosiale nettverk. Denne oppgaven presenterer en agentbasert modell, utviklet med Mesa, et Python-rammeverk, for å simulere hvordan ulike anbefalingsalgoritmer påvirker spredningen av feilinformasjon. Den epidemiologiske modellen SEIS brukes for å fange opp brukernes tilstander over tid. Algoritmene som evalueres i dette arbeidet er popularitetsbasert, collaborative filtering, content-based filtrering og tilfeldig (random). Det er foreslått en hypotese om at økt mangfold i anbefalinger kan bidra til å bryte opp ekkokamre og begrense spredningen av feilinformasjon. For å undersøke dette, testes hvordan økt mangfold påvirker de ulike anbefalernes effekt på spredningen av feilinformasjon ved hjelp av etterbehandlingsmetoden Maximal Marginal Relevance (MMR). Oppgaven ser også nærmere på samspillet mellom mangfold og ekkokamre. Resultatene fra eksperimentene viser at popularitetsbaserte algoritmer har størst negativ innvirkning på spredning av feilinformasjon og ofte forsterker feilinformasjon i sine anbefalinger. Algoritmer som bygger på innholdslikhet, som content-based filtering og innholdsbasert collaborative filtering, viste seg å være mest effektive til å begrense spredningen av feilinformasjon. Økt mangfold i anbefalingene ved hjelp av MMR hadde ulik effekt avhengig av algoritmetypen. Mangfold reduserte forsterkningen av feilinformasjon for popularitetsbaserte algoritmer, selv om selve spredningen forble uendret. Innholdsbasert collaborative filtering og content-based filtering økte eksponeringen for feilinformasjon, men spredningen av feilinformasjon ble redusert. Brukerbasert collaborative filtering var stabil både med og uten økt mangfold og presterte omtrent som den tilfeldige algoritmen. Selv om økt mangfold viser begrenset innvirkning på den faktiske spredningen og eksponeringen av feilinformasjon, indikerer resultatene at det bidrar til redusert ekkokammereffekt på tvers av alle algoritmer. Dette antyder at selv om mangfold i seg selv ikke er tilstrekkelig for å forhindre feilinformasjon, kan det bidra til å bryte opp ekkokamre.

Preface

This master thesis was written at the Norwegian University of Technology and Science (NTNU), Department of Computer Science (IDI).

We express our sincere gratitude to our supervisor, Özlem Özgöbek for providing valuable feedback and guidance throughout the thesis. From the initial exploration of research gaps to the final revisions of the thesis. Secondly, we want to thank our co-supervisor, Önder Gürcan, for his assistance and his experience with agent-based modeling. Both supervisors offered constant encouragement and helped us reflect on our choices and directions.

Lise Jakobsen & Anna Johanne Holden Jacobsen
Trondheim, June 17, 2025

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

ABM	Agent-Based Modeling
CBF	Content-Based Filtering
CF	Collaborative Filtering
DI	Diversity Increase
DPP	Determinantal Point Process
DS	Diversity Score
EC	Echo Chamber Effect
IB-CF	Item-Based Collaborative Filtering
ICM	Independent Cascade Model
IR	Infection Rate
LTM	Linear Threshold Model
MC	Misinformation Count
MMR	Maximal Marginal Relevance
MR	Misinformation Ratio
MRD	Misinformation Ratio Difference
NP	Non-Personalized
ODE	Ordinary Differential Equation
OSN	Online Social Network
Pop	Popularity-Based
RA	Recommendation Algorithm
Rnd	Random
SEIR	Susceptible–Exposed–Infected–Recovered
SEIS	Susceptible–Exposed–Infected–Susceptible

SIR Susceptible–Infected–Recovered

UB-CF User-Based Collaborative Filtering

Chapter 1

Introduction

This chapter introduces the thesis and highlights its key aspects. Section 1.1 presents the background and motivation for the study. Section 1.2 defines the problem that influenced the objective of this thesis, which is detailed in Section 1.3. A summary of the research method is given in Section 1.4. The main contributions of this thesis are summarized in Section 1.5. Finally, Section 1.6 presents the overall structure of the thesis.

1.1 Background and Motivation

Social media platforms have become a central source of information, changing the way people engage with news and information [1, 2]. Although these platforms give exposure to diverse perspectives, they have also become major channels for the spread of misinformation [2]. This misinformation could alter the reliability of the information ecosystem, making it difficult for users to distinguish between true and false content [2].

Part of the problem is driven by the way the social media platforms recommend content. RAs are designed to maximize user experience by showing content that aligns with a user's past behavior and preferences. This can end up exposing users to more misinformation that supports their existing views. Over time, this process can lead to the formation of echo chambers and filter bubbles, where users primarily encounter information that aligns with their current beliefs [3, 4]. By failing to promote diverse, accurate perspectives, today's recommendation systems risk enhancing polarized viewpoints, and can slow down the kinds of coordinated responses the world needs for social and environmental sustainability.

Although a significant amount of research has focused on detecting and moderating misinformation, less research has been conducted on how the design

of recommendation systems contributes to its spread. This is an important gap, especially since recommendations are one of the few areas that platforms have direct control over before content is shared [5]. Failing to address the role of RAs in the spread of misinformation has significant consequences [4]. Beyond immediate societal harms, such as increased polarization and diminished trust in institutions, the unchecked spread of misinformation undermines the credibility of digital platforms themselves. As RAs continue to shape public opinion on critical issues such as elections and public health, the responsibility of researchers to develop solutions that mitigate the propagation of misinformation has become more urgent.

1.2 Problem Outline

The spread of misinformation on social networks is not solely the result of user behavior or social intent. It can also be affected by algorithmic curation, which determines which content is shown to whom. Although platforms cannot always control what users post, they can influence how content spreads by adjusting their recommendation strategies. However, traditional RAs focus mainly on improving the precision of the recommendation to achieve user satisfaction, often at the expense of diversity and novelty [6]. This over-personalization can lead to echo chambers, where users are repeatedly exposed to content that reinforces their opinions [7].

Analyzing how the different RAs affect the spread of online misinformation is therefore important to evaluate new approaches for mitigation. Existing studies describing the effect of different RAs on the spread of misinformation have laid a good foundation for future research [4, 8, 9]. Furthermore, research has examined the correlation between increased diversity in recommendations, echo chambers, and less propagation of misinformation [3, 10]. However, these studies usually focus on specific platforms or utilize static datasets, which limits the generalization of their findings.

This study differentiates itself with the use of Agent-Based Modeling (ABM) to simulate a realistic social network environment. This enables a controlled investigation of how recommendation systems and increased diversity impact the spread of misinformation.

1.3 Goals and Research Questions

The general goal of this study is to investigate the role of RAs in the amplification of misinformation and their influence on the formation of echo chambers. Furthermore, the study explores whether increasing diversity in content recommendations

can reduce these negative effects.

This study is guided by the following research questions:

- **RQ1:** How do different RAs affect the spread of misinformation?
- **RQ2:** How does increasing diversity in recommendations affect the spread of misinformation?
- **RQ3:** How does the echo chamber effect change in relation to RAs and increased diversity?

1.4 Research Method

To address the research questions, a simulation-based approach using ABM is applied. This allows for investigating social dynamics and recommendation systems in an environment where key components can be systematically varied.

Simulations are implemented using the Python framework Mesa [11]. RAs are developed using LensKit in cases where the required input data is available within the simulated environment and the underlying computations are complex enough to justify the use of a dedicated library [12]. The agents in the model represent regular users, bots, and influencers, each with unique behavior patterns and content preferences. The social network is fixed across runs to ensure fair comparisons.

Each simulation scenario evaluates a specific RA. The propagation of misinformation through the network is tracked over time, and the following metrics are measured:

- Infection Rate (IR)
- Misinformation Ratio Difference (MRD)
- Misinformation Count (MC)
- Echo Chamber Effect (EC)

By comparing the performance of different RAs using these metrics, the analysis highlights how recommendation systems influence the propagation of misinformation on social networks.

1.5 Contributions

The work in this thesis contributes to multiple research fields, including the modeling of social networks, recommender systems, and the propagation of misinformation.

First, it introduces an agent-based model [1] that brings to life a social network of diverse actors: everyday users, bots, and influencers. Built in Python with the Mesa framework and LensKit, this platform allows for the integration of recommender-system influence. By embedding temporal dynamics and feedback loops, the model reflects how misinformation and algorithmic curation evolve together over time. The ABM also comes with a visualization dashboard, which can be used to further develop and analyze how RAs affect the propagation of information on social networks. The code for the implementation is open-source, ensuring that insights from this model can inform ethical design standards and media literacy, which are important to support sustainability.

Second, the thesis evaluates four distinct recommendation strategies, examining how each affects the flow of misinformation: popularity-based, collaborative filtering, content-based, and random. It further explores how increasing content diversity with the approach MMR can alter a recommender's impact on misinformation propagation. By examining how different recommendation strategies influence the spread of misinformation, the thesis can identify sustainable design choices that help promote factual information.

Finally, a new metric is proposed for measuring echo chambers directly within this simulation to further analyze the relationship between misinformation, diversity, and echo chambers.

In addition to the previous contributions, a long paper titled *"Agent-Based Exploration of Recommendation Systems in Misinformation Propagation"* has been accepted for the Social Simulation Conference 2025[2] in collaboration with the supervisors.

1.6 Thesis Structure

The rest of the thesis is structured as follows, where sections about the background theory and related work are based on the previous completed specialization project:

- **Chapter 2: Background Theory**

  Provides definitions and theoretical context around misinformation, social media dynamics, and recommendation systems.

- **Chapter 3: Related Work**

  Summarizes research conducted on the relevant themes for this thesis.

- **Chapter 4: Methodology**

[1] `https://github.com/lisejakobsen/MasterRaFakeNews` Last Accessed: 27.05.25
[2] `https://ssc2025.tbm.tudelft.nl/`

Describes the tools used to implement the model, the overview of the ABM, the approach to increase diversity, and the evaluation metrics.

- **Chapter 5: Experiments**

  Outlines the experimental setup and a detailed description of the two experiments carried out in this thesis.

- **Chapter 6: Results**

  Presents the results from the experiments, organized by metric.

- **Chapter 7: Discussion**

  Reflects on the implications of the results, answers research questions, and discusses the limitations of the work performed, including the impact on sustainability.

- **Chapter 8: Conclusion and Future Work**

  Summarize the main findings related to each research question and outline future research directions.

Chapter 2

Background Theory

This chapter provides an overview of the theoretical background that is needed as a prerequisite for the experiments in this thesis. Section 2.1 introduces misinformation and its presence on social media platforms. Section 2.2 presents an overview of the different RAs. Section 2.3 defines Agent-Based Modeling (ABM), while Section 2.4 outlines Ordinary Differential Equations (ODEs) and epidemiological models. Finally, Section 2.5 describes diffusion models.

2.1 Misinformation

The term "misinformation" has been defined in the literature in various ways depending on the domain and context. This makes clarification important to ensure consistency throughout the thesis. First, different definitions will be presented before providing the definition that will be used in this thesis. In general, it is common to distinguish between misinformation, malinformation, and disinformation as different definitions of "fake news" in the literature:

- **Misinformation** often refers to false, incorrect, inaccurate, or misleading information that is shared without malicious intent [13].

- **Disinformation** is false information created with the intention of deceiving or manipulating users [13].

- **Malinformation** is used about true information that is used maliciously. This can include accurate information taken out of context with the intention of harming individuals or groups [13].

Other definitions of fake news also include aspects such as irony or satire, while some exclude these definitions due to non-malicious intent. Another perspective considers what identity the content publisher has, separating between false information spread by individuals and fabricated stories that are created by organized entities. In this thesis, a broad definition is used, and "misinformation" is further defined as: *all content that is factually incorrect but is shared as a factual statement, with or without malicious intent (excluding satire and irony), regardless of source or medium.* Although "misinformation" is used as the primary term throughout the thesis, the term "fake news," which refers to the same concept, will also appear in the context of related work and where code variables are named accordingly.

2.1.1 Misinformation and Social Media

Social media is a central aspect of modern society, but it has also become a major channel for the spread of misinformation. This problem can have a great influence on political events and outcomes. During the 2016 US presidential election, more than one million tweets were related to misinformation about the "Pizzagate conspiracy theory" [2], and the most shared fake election stories generated more than 8.5 million user interactions on Facebook [14]. Political misinformation can have a big influence on people's political views, and in the months leading up to the election, misinformation that was "pro-Trump" was shared 30 million times, compared to "pro-Clinton" posts shared only 8 million times [15]. The reach of political misinformation is also significantly larger than that of misinformation from other categories, indicating an uneven distribution of misinformation on different topics online [16]. Such examples show how powerful online platforms are in amplifying the propagation of misinformation and highlight the potential impact of such amplification.

Social media consists of regular users, but influencers and bots have also gained traction in social networks. Influencers can be defined as users with a large number of followers and have emerged as authoritative figures with a greater impact on information dissemination than regular users [17]. Influencers are usually more strategic about their social media activity and carefully manage their content to make a greater impact [18].

Bots, or social bots, can be defined as machine programs online that automatically produce content and interact with humans, trying to emulate and possibly alter their behavior [19]. The number of bots on social media has increased, as more sophisticated programs can be designed to mimic human behavior on social networking sites. For the last decade, online bots have been used in an effort to propagate misinformation and propaganda [19]. Bots will be continuously more active on social media at all times, showing consistent high activity [20].

Users who consistently distribute a significantly large amount of low-credibility content can be defined as superspreaders. These superspreaders are responsible for a disproportionate share of misinformation spread [21]. Superspreaders can be malicious bots, regular users, and influencers who propagate misinformation to their large following.

Social networks further amplify the propagation of misinformation through features such as algorithmic timelines, retweets, and likes. By optimizing user engagement, users will be at risk of being repeatedly exposed to content that aligns with their previous beliefs. The tendency of a person to prefer information that confirms their preexisting beliefs is termed confirmation bias [2]. In addition, this bias can create filter bubbles, where a user is only exposed to content similar to what the user has previously interacted with [3].

Users with similar interests are more likely to follow each other on social media, which means that topic similarity between users can predict social links [22]. Therefore, social networks include clusters of users with similar preferences, which can create a good foundation for the formation of filter bubbles.

Because filter bubbles reduce exposure to diverse perspectives, they can lead to intellectual isolation and greater vulnerability to misinformation. An extension of filter bubbles is the **echo chamber effect**, where users repeatedly encounter opinions and information that reinforce their own [7]. This creates a group of like-minded people that amplify shared beliefs and exclude alternative perspectives. As this process continues, a feedback loop is created that can lead to greater polarization [2].

Misinformation also induces psychological effects that make users more likely to engage with it than with regular content [23]. Filter bubbles and echo chambers, along with misinformation being highly engaging, can therefore lay a foundation for the rapid spread of misinformation.

2.2 Recommendation Systems

Recommender systems are software techniques that generate suggestions of different types of items for a user, such as what music to listen to or what online news to read [24]. Recommendations are usually personalized, where algorithms offer a ranked list, which is a prediction of the most suitable items for the user, based on their preferences and constraints [24]. There are also non-personalized algorithms, which are usually easier to generate [24].

2.2.1 Non-Personalized Recommendation System

Modern recommendation systems often personalize content with the use of individual preferences, but some simpler approaches do not take these preferences into

account when recommending content. Non-Personalized (NP) recommendation systems, such as Random (Rnd) or Popularity-Based (Pop) algorithms, suggest recommendations without analyzing user-specific interactions. An example of a Pop recommendation is to suggest the top 10 movies. These methods are often used as benchmarks in the research of recommendation systems rather than as practical solutions [24].

2.2.2 Personalized Recommendation Systems

Content-Based Filtering

Content-Based Filtering (CBF) recommends items to users by identifying similarities between items previously liked and new content [25]. The algorithm extracts features from the user history and compares them with attributes of unseen data. For example, if a user positively rates a movie in the comedy genre, the system can learn to recommend other movies in this genre [24]. CBF is widely applied in domains such as news platforms, online articles, and television programming [25].

To give recommendations, the CBF creates a model of user preferences, often by treating the task as a classification problem to help identify user interests based on training data. Feedback for these models can be collected explicitly, such as users rating a movie with stars, or implicitly with interactions such as clicks, views, and purchases. Explicit feedback is typically more accurate but often sparse, while implicit feedback is easy to collect but less accurate, thus less related to user satisfaction [26]. Common classification techniques used in CBF include decision trees, nearest neighbor algorithms, and linear classifiers.

CBF struggles when the model has only insufficient data to separate between user preferences, which makes it less effective if user interests are ambiguous or difficult to categorize. In these scenarios, collaborative filtering may provide a better alternative [25].

Collaborative Filtering

Collaborative Filtering (CF) is a widely used approach that generates recommendations based on the user behavior and preferences of similar users [27]. Unlike CBF, which depends on item attributes, CF identifies patterns by looking at user-item interactions. These interactions are stored in a user-item interaction matrix, where each row corresponds to a user and each column to an item. The entries represent user feedback collected explicitly or implicitly. This enables CF to provide recommendations beyond the content itself, which can lead to more unexpected and diverse suggestions.

Although CF is effective in capturing hidden patterns in user behavior, it requires a great deal of interaction data to function optimally. This reliance

Table 2.1: User-item matrix

	Item A	Item B	Item C
User 1	1	1	0
User 2	1	1	1
User 3	0	0	1

on user activity can be a limitation compared to CBF, which can generate recommendations based solely on the characteristics of the item [25]. CF methods are generally classified into two main types: Memory- and model-based methods. Model-based methods extract patterns using the user-item matrix [27]. Methods such as matrix factorization are among the most popular model-based methods that decompose the interaction matrix into latent factors that represent user preferences and attributes of items. Latent factors may, for example, capture a user's preference for a movie or an item's popularity.

Memory-based CF relies on similarity metrics to identify relationships between users or items. These approaches are also known as neighborhood-based since they seek the best neighborhood users who share similar interests [28]. The method can be divided into two subtypes:

User-Based Collaborative Filtering (UB-CF) identifies users similar to the target user based on the preferences of the shared items [28]. Each user and item is represented in a user-item interaction matrix, and similarities between users are calculated using distance metrics such as cosine similarity [25]. Once the system identifies users with similar behavior, it recommends items that those users have rated but that the target user has not previously seen.

This is illustrated in Table 2.1. In UB-CF, the similarity between users is computed on the basis of the items they have interacted with. For example, for User 1, the similarity with User 2 is calculated using the following vectors:

- User 1 = [1, 1, 0]
- User 2 = [1, 1, 1]

Since these two vectors are quite similar, User 2 is identified as a good match for User 1. Because User 2 has rated Item C, which User 1 has not, the system would recommend Item C to User 1. This similarity computation is repeated across all user-user pairs to find the most relevant neighbors.

Item-Based Collaborative Filtering (IB-CF) focuses on the similarity between items, which is calculated using the user ratings for those items [28].

Returning to Table 2.1, this time the similarity is calculated between the vectors of the item instead of the vectors of the user. Considering User 1 as the target user, it has rated Items A and B but not Item C. To determine whether or not Item C should be recommended, it is compared with items User 1 has already rated. For example, the similarity is computed between:

- Item A = [1, 1, 0]
- Item C = [0, 1, 1]

Because Item A and Item C have some similarities and are both rated by User 2, User 1 might also be interested in Item C. As a result, Item C will be recommended to User 1. This similarity is repeated for all unrated items, based on their relationships with the items the user has interacted with. One commonly used similarity metric is cosine similarity.

Cosine similarity The computation of similarity between users or items is based on the user-item matrix. The similarities between users or items can be calculated using various similarity measures. Among them, cosine similarity is a popular choice due to its simplicity and effectiveness [28]. The cosine similarity between two vectors A and B is defined as:

$$\text{cosine\_similarity}(A, B) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{2.1}$$

where A and B are vectors for the user or item. The similarity score ranges from -1 to 1, with a higher score indicating greater similarity between the vectors.

Hybrid Systems

Hybrid recommendation systems are a combination of several recommendation techniques with the aim of maximizing their strengths while mitigating their weaknesses [29]. Traditional methods such as CF and CBF each have limitations such as cold start, sparsity problems, and the ability to discover novel content. Hybrid systems are there to address these challenges by integrating multiple models.

Integration can occur in various forms, one of which is the combination of CF and CBF. This hybrid approach can be implemented by improving CF using data from the CBF user profile data or allowing CBF to take advantage of collaborative insights of CF to identify users with similar preferences and improve personalization [29]. Another approach is to combine the results of multiple recommenders using weighted averages or voting. Some systems further adapt by switching between models, depending on the context or available data.

Neural Networks

Neural network approaches in recommendation systems are often used to obtain extra data, especially in situations where it is challenging to understand user preferences with their historical data [29]. Adapting neural networks, and deep neural networks, can aid in solving the common issues in CF, including sparsity in the user-item matrix, and the cold start problem [29].

2.2.3 Cold-start problem

Recommendation systems can face several challenges when deployed, such as scalability, privacy concerns, and bias in recommendations. One of the most central issues in RAs is the cold-start problem, which occurs when there is insufficient data to generate meaningful recommendations. This issue appears primarily in three scenarios [25]:

1. **New users**: Without an interaction history, the system struggles to provide relevant suggestions.
2. **New items**: Freshly introduced content lacks ratings or engagement, making it difficult for the algorithm to recommend it.
3. **New communities**: When a recommendation system is deployed in a new environment, there may be too little data to identify reliable patterns.

The cold-start problem is especially important because it directly affects the system's ability to provide useful recommendations during initial use. Effective integration of new users and items is essential to maintain user engagement and content diversity.

2.2.4 Diversity in recommendations

Diversity in recommendation systems often refers to the variety or dissimilarity among the items suggested to avoid overly homogeneous lists of recommendations. In practice, this concept is often measured as the average dissimilarity between all pairs of items in a recommendation list [30]. A highly diverse list might include items from different genres or topics, rather than many very similar items. Diversity can be considered at the individual level (intra-list diversity for each user's recommendations) and at the system level (aggregate diversity across all users). Other closely related concepts that often are discussed alongside diversity are novelty and serendipity. Novelty focuses on recommending items that are new to the user, with the intention of surprising the user with content they have not previously seen [25]. Serendipity builds on novelty because it not only recommends

new items but also ensures that these items are unexpectedly satisfying [31]. A recommendation is considered serendipitous when it surprises the user with both novel and valuable content. Ensuring high diversity recommendations helps to present content to users that is not over-specialized and can also introduce novelty in users' consumption of content online [30].

Diversity in recommendation systems is, in general, about balancing relevance with variety. The system should deliver relevant suggestions that are not all alike, thereby keeping users engaged and avoiding the redundancy or "echo chamber" effect of overly narrow personalization.

Increasing Diversity

Over the years, many techniques have been proposed to increase the diversity of recommendations without having to sacrifice accuracy [30]. These methods can be categorized based on when they intervene in the recommendation pipeline: pre-processing (adjusting the input data), in-processing (modifying the learning algorithm or objective), or post-processing (adjusting the final recommendations).

Pre-processing approaches improve diversity by altering training data or input before training the recommendation model, and recognize that often there is an issue with the data itself [32]. The idea is to reduce biases in the data (e.g., popularity bias) that lead to monotonous recommendations. One common strategy is to resample or reweight the data. For example, the system may oversample underrepresented items, or downsample overrepresented items so that the model is trained on a more balanced dataset [33]. This can prevent the model from simply learning to always recommend the most popular items. Popularity bias, a tendency to overrecommend popular items, is essentially a class imbalance problem that harms diversity by overshadowing "long-tail" items [34].

By making adjustments to the data distribution (such as giving more weight to niche items or rating users with less mainstream preferences), the model can learn to recommend a wider variety of content to its users. An advantage of pre-processing methods is that they are model-agnostic: by making the input data more diversity-friendly, any standard RA trained on those data is likely to produce more diverse outputs [33]. However, it is important to be careful to maintain enough relevant information. Aggressive resampling or data manipulation can degrade accuracy by removing important signals along with the bias [33].

In-processing methods integrate diversity-promoting mechanisms directly into the RAs learning phase. A common approach is to modify the loss function or objective to explicitly include a term for diversity [33]. In a multi-objective

optimization framework, the training objective can be formulated as:

$$L_{\text{total}} = L_{\text{rec}} + \lambda L_{\text{diversity}}$$

where L_{rec} is the loss of the standard recommendation (e.g., based on rating prediction or ranking error), and $L_{\text{diversity}}$ is a term that penalizes the lack of diversity (or rewards diversity) in the recommended list, with λ controlling the trade-off [33]. By tuning λ, accuracy and diversity can be balanced, where a higher value λ encourages the model to spread recommendations across different items or categories even if that slightly sacrifices relevance.

There are various forms of diversity regularizers. $L_{\text{diversity}}$ can be defined to penalize similarity among the top-N items recommended to the same user, which effectively encourages intra-list diversity. This could be implemented by using vectors of the characteristics of the item and adding a term that minimizes the average cosine similarity between the pairs of recommended items [33]. Another approach is to regularize against popularity bias by down-weighting the loss contributions on popular items, thereby making errors on niche items more costly. This encourages the model to focus more on getting those right [34]. This kind of regularizer aims to expose users to different viewpoints, representing a specific interpretation of diversity.

The key part of the in-processing idea is that the recommendation model is explicitly trained to consider diversity as part of its goal, rather than treating it as an afterthought. The challenge with in-processing approaches is ensuring that the added diversity term does not excessively impact the model's ability to capture relevance, and make the model less interpretable [32]. This issue requires the optimization function to be accessible, replaceable and modifiable, which can be challenging itself [32].

Post-processing methods, also referred to as re-ranking or result diversification algorithms, operate after the initial recommendation list or candidate set has been generated [33]. These methods adjust or reorder the top-N results to increase diversity while trying to retain high relevance. As post-processing methods only needs access to predictions, and not the actual algorithms, these methods are applicable for black-box scenarios where not all of the elements in the recommendation pipeline are exposed [32].

A classic algorithm in this category is Maximal Marginal Relevance (MMR), originally proposed in information retrieval but widely used in recommendation systems [35]. MMR iteratively builds a diverse list by selecting items that are both high in relevance and dissimilar to the items already selected. Essentially, at each step, MMR selects the item that maximizes a weighted score: score = $\lambda \cdot$relevance$-(1-\lambda)\cdot$similarity-to-selected, where λ controls the emphasis between relevance and novelty.

Another common approach for post-processing diversification relies on Determinantal Point Processs (DPPs). DPPs are probabilistic models that naturally prefer diverse sets of items. They assign a higher probability to item sets that have a lower correlation with each other. In the context of recommendations, a DPP-based re-ranker will choose a subset of candidates that maximizes the determinant of a kernel matrix derived from item features or similarity measures [36]. This mathematically encodes the idea that the selected items should be relevant on their own and collectively diverse. Fast algorithms are developed for the maximization of DPP enabling its use in top-N recommendations [37]. In practice, DPP approaches have been implemented in industry as a way to improve catalog coverage and user satisfaction.

Post-processing diversification techniques like MMR and DPP operate on the output rankings to inject diversity after the initial recommendation has been generated, which is convenient because they can be applied on top of any RA without changing their fundamental functionality. They can offer explicit control over the diversity level using parameters or objective functions, though they may lead to some loss in immediate relevance or precision as a trade-off.

Maximal Marginal Relevance

The classical MMR approach iteratively selects items that maximize the following objective function [35]:

$$\text{MMR} = \underset{d_i \in R \setminus S}{\arg\max} \left[\lambda \cdot \text{Sim}_1(d_i, q) - (1 - \lambda) \cdot \max_{d_j \in S} \text{Sim}_2(d_i, d_j) \right] \tag{2.2}$$

where:

- R represents the candidate set of recommendations
- S represents the set of already selected items
- q represents the user's preference vector
- d_i represents a candidate item not yet selected
- d_j represents an already selected item
- $\text{Sim}_1(d_i, q)$ measures the relevance of item d_i to the user's preferences
- $\text{Sim}_2(d_i, d_j)$ measures the similarity between items d_i and d_j
- λ is a parameter that controls the trade-off between relevance and diversity

In this implementation, both similarity functions utilize cosine similarity between topic vectors.

2.3 Agent-Based Modeling

Agent-Based Modeling (ABM) is a computational approach that is used to simulate the actions and interactions of autonomous agents within a defined environment. Each agent operates according to a set of rules, and, through these low-level interactions, system-level behavior can emerge [38]. ABM has been used, in particular, to study systems such as populations, markets, and organizations [38]. Unlike traditional top-down modeling approaches, ABM creates a system bottom-up, further enabling the representation of heterogeneity and adaptive behavior. This gives researchers the opportunity to capture agent characteristics and local decision-making processes, a flexibility that makes ABM well suited to explore scenarios or policy interventions [38].

ABM has become a useful tool for studying the diffusion of information in social systems. In social networks, agents can be designed to represent users with behavior influenced by personal preferences, social connections, prior experiences, or exposure to content [5]. Many such models are based on epidemiological frameworks to simulate how information spreads similarly to infectious diseases [39]. More explanation of the epidemiological frameworks can be found in Section 2.5.

ABM allows for detailed modeling of agent heterogeneity, social structures, and behavior mechanisms [40]. Synthetic populations can, for example, be made with attributes such as age, occupation, or susceptibility to misinformation before being integrated into multilayer networks to represent families, workplaces, or online communities [40]. This gives ABM the opportunity to extract both user behavior on a micro-level as well as macro-level trends, which makes it particularly appropriate for modeling aspects such as misinformation spread, opinion shifts, and the development of echo chambers.

2.4 Ordinary Differential Equations

ODEs are mathematical tools for modeling the evolution of dynamical systems and have been extensively used to model the dynamics of information diffusion on online social networks [41]. An ODE is an equation that relates a function of an independent variable, usually time, to its own derivatives [42]. ODE models mainly consist of one or more equations that describe how the state variables of a system change over time. By specifying the rate of change for each variable as a function of the current state, ODEs can provide a continuous and deterministic representation of system dynamics [42]. ODE-based models generally offer a deterministic framework for investigating how information or innovations propagate through a population over time. These models are particularly useful for providing a global and continuous, time-continuous perspective on the collective behavior of users, even in the absence of detailed interaction-level data [41].

A central ability of ODEs is how they can break systems into separate compartments or categories and then model the flows between them. This approach is often used in epidemiology and network science to model contagion-like processes. A classic example is the SIR model, which partitions the population into three compartments and uses ODEs to describe state transitions over time [41]. Such equations encapsulate the mass-action assumption that each individual has an equal probability of contacting any other, making the infection rate proportional to the product $S \cdot I$ [41].

ODEs can be an alternative to ABM, as ODE-based compartment models have been extensively applied in other research fields, including information diffusion and social contagion, precisely because they represent how fractions of a population change state over time under average interactions [41]. However, ODE models are deterministic mean-field approximations and come with certain assumptions. An example is how they assume a well-mixed population and often linear proportionality between encounter rates and population fractions [41]. In contrast, ABMs simulate each user and their discrete interactions, capturing randomness and network structure explicitly.

ODE models can be used as an alternative to ABM by providing a simplified aggregate-level description of the system. They sacrifice some realism, such as ignoring detailed network topology or individual variability, but in exchange for analytical simplicity and speed [43].

2.4.1 Epidemiological Models

A widely used approach to model information spread is built on ODEs and involves comparing it with disease transmission. Here, misinformation is treated similarly to a virus that "infects" users as it is spreading. Epidemiological models, which were originally used for disease outbreaks, are now also used in social network analysis [41]. In the following paragraphs, some of the most adapted epidemiological models in social dynamics will be explained.

SIR

The Susceptible–Infected–Recovered (SIR) model usually acts as the baseline for epidemiological models, and subsequent studies have extended it by adding or removing elements [39]. SIR assumes that individuals transition from being Susceptible to misinformation (S), to Infected (I), meaning they spread misinformation, and eventually enter a Recovery state (R) where they are no longer active participants. This means that SIR assumes permanent immunity, which does not necessarily reflect real-world social behavior, where users might engage with misinformation several times [39]. The classical formulation is given by differential equations such as:

$$\frac{dS}{dt} = -\beta SI \tag{2.3}$$

$$\frac{dI}{dt} = \beta SI - \gamma I \tag{2.4}$$

$$\frac{dR}{dt} = \gamma I \tag{2.5}$$

where β is the infection rate, and γ is the recovery rate [41]. In a network context, a S node becomes I with probability β when connected to an I neighbor, and an I node recovers (becomes R) at rate γ. In terms of misinformation, S represents a user who has not encountered false information, I corresponds to a user currently sharing or convinced by false information, and R represents a user who has stopped believing the misinformation, possibly due to debunking or loss of interest, and will not share it further [39].

SEIR

The Susceptible–Exposed–Infected–Recovered (SEIR) model further extends the SIR by adding an *Exposed* (E) state between S and I. An individual marked as *Exposed* has encountered misinformation but has not yet actively spread it. The SEIR model captures the contagion with incubation, adding a more realistic delay between exposure and infectiousness [39], which in social networks can correspond to the time users take to verify or gradually accept misinformation. The formulation for SEIR is similar to SIR, but adds the complexity of an additional state. Transitions typically include an exposure rate and an activation rate: for example, $S \to E$ when coming into contact with misinformation (at rate β), then $E \to I$ at some rate σ, representing the delay before the person starts spreading it, and $I \to R$ at rate γ. The formal equations can be written as [44]:

$$\frac{dS}{dt} = -\beta SI \tag{2.6}$$

$$\frac{dE}{dt} = \beta SI - \sigma E \tag{2.7}$$

$$\frac{dI}{dt} = \sigma E - \gamma I \tag{2.8}$$

$$\frac{dR}{dt} = \gamma I \tag{2.9}$$

SEIS

Previous models assume that a node ends up in a final recovered state, but the Susceptible–Exposed–Infected–Susceptible (SEIS) model is a variant in which there are no permanent states [45]. This version combines the exposure delay with no possible immunity, reflecting real-life scenarios in which users really never become permanently immune to misinformation. It also adds the functionality for users that stop actively spreading misinformation to be able to be infected again in the future. The mathematical formulation of the differential equations can be defined as [45]:

$$\frac{dS}{dt} = -\beta SI + \gamma I \tag{2.10}$$

$$\frac{dE}{dt} = \beta SI - \sigma E \tag{2.11}$$

$$\frac{dI}{dt} = \sigma E - \gamma I \tag{2.12}$$

which is quite similar to the previous equations but adds a dependency to the recovery rate (γ) times I in the differential equation for the *Susceptible* state.

2.5 Information Diffusion Models

Information diffusion models provide a framework for understanding how information propagates in social networks [46]. These diffusion models are probabilistic graph-based models with a discrete-time nature, compared to epidemiological models whose foundations are built on ODEs, which are deterministic and continuous.

In diffusion models, individuals are represented as nodes and their interactions as edges, following principles from graph theory [47]. Individuals cycle through different states, which are defined by transition rules that can capture how information propagates through a network. By mimicking real-world communication processes, diffusion models provide valuable insights into the dynamics of information propagation, helping researchers understand how content travels through interconnected systems.

2.5.1 Linear Threshold Model

One information diffusion model is the Linear Threshold Model (LTM) [47], which assumes that people adopt information based on the collective influence of their neighbors. Each user is assigned a threshold value between 0 and 1, which represents the minimum level of influence required to adopt the information.

The influence of neighbors is modeled through weighted edges, where stronger connections mean stronger pressure. A user becomes 'activated' (adopts or shares the information) if the sum of the influences from its activated neighbors exceeds its threshold [47]. This model is particularly useful for simulating peer-influenced behavior, such as users being persuaded to believe or share misinformation due to repeated exposure within connected groups.

2.5.2 Independent Cascade Model

Another model is the Independent Cascade Model (ICM), which is based on probabilistic influence. Whenever a node becomes active (Infected) at time t, it has a single opportunity to activate each neighbor that is currently inactive at time $t+1$ [48]. Each contact is successful with a certain probability (often denoted p per edge). If successful, the neighbor becomes active in the next time step. The process proceeds in discrete timesteps until no further activations occur [48]. ICM is optimal for capturing the spread of information, especially when content is shared through direct interactions, such as retweets or reposts on social networks. Unlike LTM, where influence accumulates, ICM models the immediate impact of information exposure.

Chapter 3

Related Work

This chapter presents related work on topics that are relevant to this study. It provides a foundation for research, helping with methodological choices and focus areas that align with the goal of mitigating misinformation spread through recommendation systems. First, Section 3.1 describes research on recommendation systems and misinformation. Section 3.2 outlines the societal impact of recommendation systems. Section 3.3 presents studies done on mitigation tactics using recommendation systems. Finally, Section 3.4 describes various modeling approaches used to simulate how misinformation spreads through social networks.

3.1 Recommendation Systems and Misinformation

RAs play an important role in shaping user experiences on social media platforms. However, their optimization for engagement has made researchers question their contribution to the spread of misinformation. Real-world datasets have been used to simulate the diffusion of misinformation on platform X, focusing on engagement-driven RAs [4] and [8]. The results indicate that Pop algorithms amplify the spread of misinformation, whereas CF and CBF recommenders have a smaller effect. Another study by [49] use YouTube conspiracy-related video data to test top-N recommenders, and shows that simpler CF models, such as Nearest Neighbors are more resilient to misinformation spread than complex models, such as Logistic Matrix Factorization (LMF). LMF often prioritizes improved performance over control of false content. These studies contribute particularly to the decision to explore classical algorithms in the simulations conducted in this thesis.

Biased news content can further worsen this problem. Studies show that RAs tend to reflect and amplify these existing biases, further reducing content diversity and making users more vulnerable to misinformation [2, 3, 8]. More discussion of content diversity and mitigation strategies is presented in Section 3.3.

Recommendation systems affect not only the content users consume but also who they connect with. A study by [9] investigates the role of friend recommenders, which are systems that suggest new social connections rather than content. This is done using a COVID-19 misinformation dataset based on tweets from platform X. The study shows that some friend recommendations can strengthen network homogeneity, leading to an increased risk of misinformation dissemination. These findings show that algorithms affect not only recommended content but also the structure of social interactions. The research influenced the consideration of network structure in the analysis, suggesting the inclusion of the dynamics of social connection in the simulation framework.

Another important aspect is the type of feedback signal used to train RAs. Many systems rely on implicit feedback because these signals are available in large volumes. However, [26] and [8] point out that implicit feedback could also contribute to feedback loops that promote low-quality information, which means that users may click on misleading headlines or controversial content out of curiosity. Superspreaders amplify these signals when they interact heavily with the content. In contrast, systems that use explicit feedback have the tendency to be more reliable but also suffer from low engagement. Zhao et al. [26] further suggest that hybrid approaches that combine both types of feedback show the potential to offer an approach with a better trade-off between engagement and content quality. These studies influenced the evaluation of implicit and explicit feedback mechanisms when designing experiments in this study.

Diversity and misinformation

It is hypothesized that low diversity in the recommendation feeds, such as very narrow personalization, can contribute to filter bubbles and echo chambers, where users are repeatedly exposed to the same or similar content and viewpoints [9]. This homogeneous exposure to information can reinforce users' beliefs and make them more vulnerable to misinformation, as they are less likely to encounter content that challenges false or misleading narratives [9]. Therefore, increasing the diversity of recommended content might reduce the echo chamber effect and thereby mitigate the spread of false information. This hypothesis largely inspired this thesis' work, as it investigates how different levels of diversity in RAs affect the spread of misinformation.

A recent study provided direct evidence that increasing the diversity of recommendations can reduce the spread of misinformation [10]. These results suggest

that diversity can act as a form of "immune response," introducing enough varied content into each user's feed so that misinformation does not spread as easily or as widely. The study concludes that incorporating diversity-aware objectives in social media recommenders could be a promising strategy for misinformation intervention, helping to mitigate the viral spread of fake news at the network level. The insights on diversity guided the choice to incorporate diversity levels into the experiments conducted in this study.

3.2 Recommendation Systems and Societal Impact

Recommendation systems offer several significant benefits, such as personalized content delivery and an enhanced user experience, but their broader societal impacts can have severe consequences.

RAs can result in the formation of filter bubbles and echo chambers, repeatedly showing users content that aligns with their past behaviors and beliefs [3]. Although this increases short-term user satisfaction, it can lead to increased polarization, reduced tolerance for alternative perspectives, and even radicalization in some online communities. Furthermore, a user-centered survey investigates how recommendation systems influence user experience and how users influence recommendations. The findings indicate that individual experiences with recommendations vary widely. This means that users may undergo changes in satisfaction and trust over time, even when it is not visible on the larger scale [50].

However, it is also important to consider the broader societal consequences. Repetition of exposure to false or polarizing content not only misinforms individuals but can also challenge the trust in democratic institutions, undermine public understanding of science, and contribute to social fragmentation.

Another challenge is the lack of transparency surrounding how these systems operate. Most major platforms do not disclose how their algorithms rank content or weigh different signals. This makes it difficult for users to understand why they are encountering certain recommendations and for researchers to study the effects of these systems. It also limits public accountability, especially when recommendation systems contribute to harmful social outcomes. Several ethical concerns are highlighted in a study of recommendation systems in digital mental health applications [51]. This includes the need to ensure explainability, protect user privacy, and respect user autonomy over data and interaction history. Another study on the larger impact of recommendation systems emphasizes that user data extracted is often not clear or easy to understand [52]. For users, it is hard to control the data being used or that can be seen by the recommendation system. Further, the systems can limit the user's freedom by pushing content on the user

based on what the algorithm thinks the person likes, often without permission from the user. Because many recommendation systems are owned by companies that keep their methods secret, it is difficult for users to challenge the way recommended content is exposed to them.

In response to concerns raised about the unintended effects of recommendation systems, researchers have focused more on the need for ethical algorithmic development and frameworks that can be regulated. Some suggestions include making the RA processes easier to understand, integrating fairness and trustworthiness regarding how items are ranked, and allowing others to review or check the system independently. An ethical recommendation framework is suggested by [53], connecting each stage of the development of the recommendation system with the ethical challenges raised. The study emphasizes the importance of giving the user control over their recommendations, such as options to limit data sharing, reset algorithmic goals, and manage content filtering. Principles such as transparency and "ethics-by-design" can help mitigate bias and privacy violations. These measures are important for protecting user rights and also for promoting trust in recommendation systems, ensuring that RAs prioritize societal interests in a responsible manner.

Although recommendation systems optimize user engagement, they can also lead to societal concerns such as polarization and the amplification of misinformation. This gives them great responsibility in influencing what people believe, who they trust, and how communities interact. This means that their design is not just a technical matter but a societal one.

The insights from examining the societal impacts of RAs have influenced the choice of algorithms and methodologies used in this study.

3.3 Mitigating Misinformation with RAs

This thesis investigates the possibility of mitigating the online spread of misinformation utilizing recommendation systems. Multiple approaches aim to reduce the propagation of misinformation online; some are user-centered, and some focus on veracity-checking content. Some approaches are more intrusive than others; where certain models simply try to "nudge" users to make better decisions when engaging in social media content, others remove content deemed "fake" completely.

3.3.1 Nudging users to make better choices

Some researchers state that directing users toward higher-quality information can mitigate misinformation, maintain user trust, and avoid more interfering approaches such as content moderation. The use of "soft nudges" in recommendation systems is researched by [54] to nudge users toward more credible

news without directly challenging the user's trust or belief. The study designs a recommendation system that incrementally introduces users to trusted sources, that are closely related to their preferences. After several iterations simulating this model, it successfully nudges users to more reliable and less subjective sources over time. The results indicate that users exposed to these trust nudges adjust their news consumption profiles, reducing their engagement with lower-quality sources, thereby helping to counter misinformation in a non-restrictive manner. Some other studies focus more on contradicting fake news by recommending fact-checked news content when a user is exposed to fake news. The results of one of these publications show that the delivery of fact-checked URLs, that align with the topic of a questionable article, makes the user less likely to share or engage in misinformation, contributing to a reduction in the propagation of misinformation [55]. These approaches demonstrate that directing users to more reliable information can be an effective approach to improving information quality while maintaining user autonomy. The study highlights that by using behavioral economics in recommendation systems, it can help users choose trustworthy information, which could help reduce misinformation on social media.

3.3.2 Directly mitigating misinformation

Recent work introduces more direct approaches to remove misinformation before users are exposed to it. These methods incorporate mechanisms to identify and filter out misinformation while recommending credible content. Two relatively new models that have produced great results are Rec4Mit [56] and FANAR [57].

Rec4Mit

The Rec4Mit model, proposed by [56], is a veracity-aware and event-driven recommendation system designed to mitigate the spread of fake news. Unlike other systems that focus solely on user preferences, Rec4Mit also distinguishes between true and fake news. In the model, a veracity classifier is used to evaluate the authenticity of news content in order to ensure that only verified news is recommended. The model aims to recommend corrective news to users who have previously been exposed to misinformation. A central aspect of Rec4Mit is its event-driven nature. The system tracks a user's engagement history with respect to different events, such as the presidential election, making it easier to detect user interests. The model separates veracity checking from event-related information using the event-veracity disentangler, which allows the system to recommend user-relevant news that is also verified. Focusing on specific events ensures that users who have read fake news about a specific topic are later recommended credible news content about the same topic. Compared to other baseline methods for news recommendation, Rec4Mit significantly outperforms the other methods

and nearly only recommends true news. This is likely due to the high accuracy of the veracity classifier, but it is important to note that the other methods do not aim to mitigate misinformation and are centered around recommendations based solely on user preferences.

FANAR

FANAR is also a news recommendation system that aims to reduce misinformation with a different objective to achieve its goals. Unlike Rec4Mit, which uses actual content verification to reduce misinformation, FANAR uses a trust-based approach. In the study by [57], the recommendation system focuses on the trustworthiness and behavior of users rather than the content itself. The key concept is to improve CF by incorporating a trust model that evaluates users based on their history of interactions with fake news. FANAR then filters out users categorized as "untrustworthy" to prevent users spreading misinformation from influencing recommendations.

FANAR extends regular CF by introducing a Beta Trust model, which estimates the reliability of users based on their previous engagement. Users who frequently interact with false information are assigned lower trust scores and are then removed from the recommendation process during the candidate generation step. Another aspect of FANAR that distinguishes it from most other recommendation models is its incorporation of visual context in the news representation. This multimodal approach outperforms models that rely solely on textual content. Although FANAR does not directly classify the veracity of news content, the paper presents results showing that the model efficiently mitigates the spread of misinformation with the focus on removing unreliable users rather than content.

The strategy is effective without requiring a large-scale fact-checking system. The paper shows how FANAR outperforms all other baseline approaches when it comes to recommending a low percentage of fake news. FANAR also outperforms Rec4Mit, which is the only other recommendation system that aims to mitigate fake news in this comparison. FANAR appears to be one of the best recommendation system for mitigating the spread of fake news, at least up until the end of the year 2024. However, it is important to note that FANAR is most useful in social networks, where user behavior plays a significant role in how news content propagates.

The mitigation strategies in Rec4Mit and FANAR show that modifying *what* and *who* influences recommendations can effectively reduce the spread of misinformation. Although the experiments in this thesis do not implement these specific mitigation strategies, understanding their effectiveness supports the investigation of diversity as an alternative approach to improving information quality in user feeds.

3.3.3 Diversifying Recommendations

Another approach to mitigate the spread of misinformation involves diversifying recommendations to reduce the echo chamber effect. The study by [10] analyzes existing recommendation systems to determine which provides higher levels of diversity. The study presents that CBF and CF have greater diversity than Pop. The same study concludes that the lower diversity in the recommendations contributes to a stronger EC, propagating misinformation at a higher rate than that of the higher diversity.

Similar results are found in a study by [3], who implements post-processing diversification techniques to investigate how they affect users' exposure to misinformation. The study compares CF with and without post-processing methods such as MMR. The results show that MMR with IB-CF clearly reduces content homogeneity and user interaction with misinformation, as well as association with filter bubbles. This is done without a major loss in accuracy. However, the improvements in both diversity and misinformation interaction are modest, suggesting that while MMR can help break up echo chamber loops, it cannot alone fully eliminate filter bubbles or misinformation spread.

Several technical approaches are proposed to mitigate negative effects, such as popularity bias or polarization in recommendation systems, many of which also serve to increase diversity. Ekstrand et al. [58] explore the use of re-sampling user data to mitigate popularity and demographic biases in CF models. The approach results in more balanced and diverse recommendations, particularly for underrepresented user groups. Similarly, [59] introduces the concept of "antidote data," strategically crafted training examples that are added to the dataset to counteract unwanted outcomes such as polarization. Although the technique aims primarily at improving fairness, it also increases diversity by broadening the training distribution and encouraging the model to recommend a wider range of items.

Taking the concept of diversity a step further, some researchers explore how introducing serendipity can further help minimize echo chambers and filter bubbles. Serendipitous recommendations expose users to diverse viewpoints or topics that they might not actively seek out. This increase in exposure disrupts the filter bubble effect. How serendipity enhances user satisfaction is highlighted by [31] while simultaneously increasing the diversity of content consumed.

These insights motivate this study to focus on diversity and guide the selection of RAs that have been studied in the context of diversity.

3.4 Modeling Misinformation Propagation

Several modeling approaches have been used to simulate how misinformation propagates through social networks, including traditional epidemiological models, information diffusion models, and agent-based simulations. These different approaches vary in complexity, underlying assumptions, and flexibility in integrating realism and heterogeneity among agents. In the following section, related studies are outlined by their modeling approach to provide an overview of the multiple strategies that have been adapted to research misinformation propagation using simulation tactics. All of these approaches were considered during the development of this thesis.

3.4.1 Epidemic Models and ODE-based approaches

Ordinary Differential Equations (ODEs) are commonly used to model social contagion processes, including the spread of misinformation. Although early applications are inspired by epidemiological models, recent research uses ODEs to explore more complex social dynamics, including the role of platform features such as recommender systems, user sentiment, and behavioral feedback loops.

In a study done by [60], a compartmental ODE model is developed to simulate the competition between misinformation and debunking content on online platforms. The model, called the Rumor Spreading Debunking (RSD) model, extends classical SIR-style equations by introducing two interacting subsystems: one for the rumor and one for the corrective information. Individuals can move between states such as Susceptible, Infected, Debunked, and Removed. The model is a system of ODEs where each equation describes the rate of change of individuals in each state. For example, the number of misinformed individuals grows based on interactions with others but declines when debunking has worked successfully. This system is used to simulate real world scenarios, including the spread of a rumor during Hurricane Harvey. The findings indicate that early and widespread debunking is critical if corrective information is introduced too late. The study demonstrates how ODEs can be used to evaluate the timing and effectiveness of intervention strategies.

Another relevant study uses system dynamics with a SIR-based modeling approach to examine how platform features such as algorithmic recommendation and echo chambers influence disinformation [61]. In the model, individuals move through compartments that represent awareness, belief, sharing, and disillusionment, with transitions governed by differential equations. The effects of recommender systems are modeled by introducing feedback loops that reinforce belief within closed communities, a sort of echo chamber. For example, users who believe a piece of disinformation are more likely to encounter reinforcing content, which increases the probability of continued sharing. The simulation results

show that in systems with strong echo chamber effects, even small disinformation campaigns can maintain high levels of influence over time. The ODE framework enables varying platform parameters, such as the strength of recommendation bias, and observing how these influence long-term dynamics. This is especially relevant for this thesis, as it highlights the feedback between user behavior and content exposure in recommendation-driven environments.

A third study proposes an ODE-based model that combines psychological factors such as emotional response and cognitive resistance to the spread of fake news [39]. The proposed model is also based on epidemiological models and includes compartments for Susceptible, Exposed, Positively Infected, Negatively Infected, and Restrained individuals. The transitions between these states are modeled using ODEs with rates that depend on parameters such as emotional engagement and resistance. The results of the study indicate that even small changes in cognitive bias or emotional response can significantly affect the level and duration of propagation of misinformation. This work shows how ODEs can be used to include more nuanced and behaviorally accurate dynamics, something more comparable to ABM approaches.

All of these studies show that ODE-based models provide a way to explore the high-level dynamics of misinformation spread. In contrast to this thesis project, they abstract away from individual user behavior and network structure. The simplicity of ODEs makes them useful for testing hypotheses, identifying thresholds for intervention, and understanding how systemic features like recommender systems can impact the spread of misinformation.

3.4.2 Information Diffusion Models

The Linear Threshold Model (LTM) is used by researchers to study how exposure from multiple peers can lead to misinformation infection. For example, [62], uses a modified version of the LTM that enables nodes to change their beliefs, allowing misinformation and truthful information to diffuse. The study simulates two spreading processes, fake news versus true news, on the same network. The study focuses on minimizing misinformation and uses community detection and trust scores to intelligently choose seed nodes to inject true information. The LTM, along with the ability for the nodes to switch opinions, provides a realistic way to model tug-of-war dynamics between misinformation and facts.

Independent Cascade Models (ICMs) have also been a popular choice in misinformation research. A multi-campaign ICM is demonstrated by [63] to simulate the parallel spread of false information and correction. In their work, "positive" seeds are the ones that spread factual information, whereas the "negative" seeds are the ones that spread misinformation, and they both diffuse throughout the network following standard ICM rules. The results of the research underline

the importance of early intervention, due to the way certain hub nodes could effectively protect parts of the network against misinformation if they reached correct information.

3.4.3 Agent-Based Models

While ODEs enable fast and interpretable experiments that can serve as a guide for the design of more complex simulations, ABMs simulate micro-level interactions. Recent studies use ABM to investigate how misinformation spreads and how recommendation systems can contribute to or mitigate this spread. Several studies simulate the spread using epidemiological diffusion models, as they can reflect the dynamics of social networks [64].

Simulating heterogeneous agents with personalized behaviors and interactions within the community, [40] demonstrates how ABM can provide an alternative to models that treat all people the same. ABM is used to simulate the dynamics of disease in Greece, allowing a detailed evaluation of policy measures through scenario testing. Although it is focused on epidemiology, this framework shows the potential of ABM to find complex aspects of information dissemination. Another study extends epidemic models by incorporating realistic online behaviors, such as the roles of bots, influencers, and time-varying engagement, within an ABM framework [64]. The work shows that standard epidemic models fall short in capturing the nuances of fake news propagation and instead suggests using more advanced agent-based simulations that include network dynamics and differences in trust. These insights are especially relevant to this thesis, which also models heterogeneous user behaviors and interactions within a recommendation system context.

Modeling disease spread and health misinformation together, [65] takes this further by treating them as coupled but distinct processes. The ABM reveals how the prevalence of "bad advice" significantly worsens health outcomes and demonstrates thresholds, such as the proportion of the population resistant to misinformation, where the effects of fake news can be effectively mitigated. This two-part modeling approach is particularly relevant for this thesis, where recommendation systems influence user exposure to information and how they respond to it.

ABM is used in [5] to analyze how recommendation systems affect polarization of opinions and the spread of misinformation. The findings show that the design of the RAs can affect the formation of echo chambers. The results further indicate that engagement-based algorithmic curation, which can be compared with RAs that consider user preferences and previous interactions, significantly amplifies the spread of misinformation and contributes to polarized communities. While the model provides good information, the simulation simplifies complex user behaviors

and the functionality of curation algorithms. This specific study is similar to what this thesis aims to explore, making it a good source of inspiration.

Taking a more integrated modeling approach, [66] introduces a hybrid stochastic-deterministic model that combines Network-Enhanced Agent-Based (NEAB) modeling with traditional diffusion metrics. This work further confirms that hybrid ABM frameworks can capture complex nonlinear dynamics in social systems, including misinformation build-up and delayed user reactions.

Chapter 4

Methodology

This chapter presents the methodology and ABM. First, Section 4.1 describes different tools used to implement the model. Then, Section 4.2 provides an extensive overview of the model architecture. Section 4.3 outlines the implementation of the increase in diversity in the recommendations. Finally, Section 4.4 defines the evaluation metrics used in this thesis.

4.1 Tools

In this thesis, several Python libraries have been used to create the model and perform the experiments. Mesa and LensKit are explained more in-depth in the following section due to their central role in the implementation. The other tools are described more sparingly due to their familiar libraries in the computer science field. The mentioned libraries are listed below:

Mesa[1] Mesa is an open-source package providing components for developing ABMs, as well as schedulers and data collectors. It also offers a real-time visualization dashboard.

LensKit[2] LensKit is an open-source toolkit created for research and education on recommendation systems. It includes modules for data processing and evaluation and is used to implement built-in algorithms such as IB-CF using their interfaces.

[1] `https://mesa.readthedocs.io/stable/` Last accessed: 19.05.2025
[2] `https://lenskit.org/` Last accessed: 19.05.2025

NetworkX[3] NetworkX is used to create and study complex networks. In this study, it is used to build a directed social network that forms the simulation model.

Matplotlib[4] Matplotlib includes tools to generate plots and data visualization. The plots created to visualize the social network and metrics in the dashboard, as well as the ones showing experimental results, use this tool.

Pandas[5] Pandas provides tools such as DataFrames, which are used in this study to work with structured data and simplify data processing and analysis.

NumPy[6] NumPy represents multi-dimensional arrays and matrices, including built-in mathematical functions. It is often used with Pandas for numerical computations. In this implementation it is used for array operations, mathematical functions and random number generation.

Scikit-learn[7] Scikit-learn is an open-source machine learning library, which provides a set of helpful tools. In this implementation it is used for cosine similarity operations.

4.1.1 Mesa

Mesa is an open-source ABM framework written in Python that is designed to help develop, analyze, and visualize simulations composed of autonomous agents [11]. Mesa provides a user-friendly and modular approach to ABM, making it accessible to both beginners and experienced modelers.

Mesa was created to address the lack of a comprehensive Python-based ABM toolkit. It's architecture is built to be modular and extensible, allowing users to easily develop models while also having the flexibility to implement customized components [11]. An overview of the Mesa architecture is presented in Figure 4.1, which is adapted from [11].

The core structure of Mesa is organized into three main components [11]:

- **Model:** This includes the main simulation structure, where:
  - *Agents* are autonomous entities with behaviors and internal states.

[3] `https://networkx.org/` Last accessed: 19.05.2025
[4] `https://matplotlib.org/` Last accessed: 19.05.2025
[5] `https://pandas.pydata.org/` Last accessed: 19.05.2025
[6] `https://numpy.org/` Last accessed: 19.05.2025
[7] `https://scikit-learn.org/stable/` Last accessed: 04.06.2025

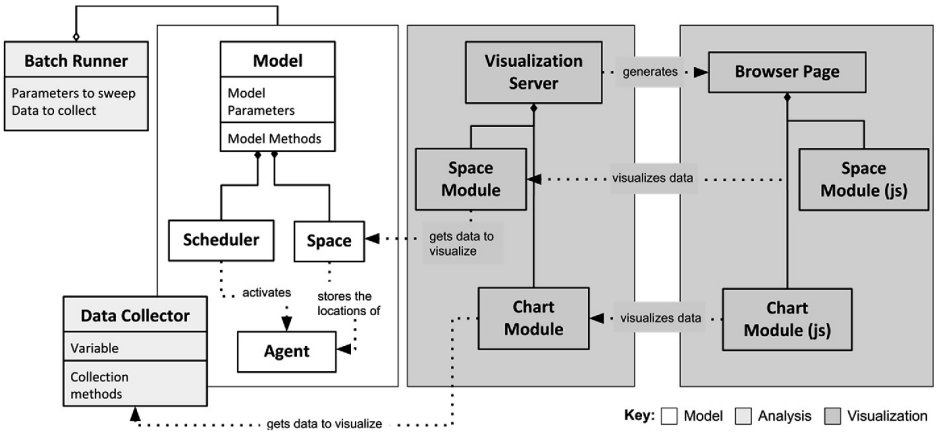

Figure 4.1: The architecture of Mesa

  - The *Schedule* determines the activation order of agents during each simulation step.
  - *Space* defines the environment in which agents operate, such as a network or grid.

- **Analysis:** Mesa provides built-in tools to collect and analyze data:
  - The *DataCollector* gathers metrics from both agents and the model.
  - The *BatchRunner* allows for automated experimentation with different model parameters.

- **Visualization:** Mesa includes an interactive visualization framework:
  - A Python server handles back-end logic and simulation updates.
  - A browser-based front end presents the model state and visualizes agent interactions in real time.

4.1.2 LensKit

LensKit is an open-source toolkit developed in Python, used for experimentation and research on recommendation systems [12]. It was originally released in 2010 as a Java framework, but has evolved into a flexible Python package that easily integrates with the scientific Python ecosystem. LensKit enables users to implement classical CF algorithms, analyze the performance of different recommendation systems, prepare data for experiments, and run batch jobs [12].

LensKit is used in different contexts, such as academic research, educational settings, or small-scale production deployments. It is designed to allow for constructed experiments that are reproducible, which is important in academia [12]. The library includes several different RAs, for example, classic neighborhood-based CF algorithms such as UB-CF and IB-CF. They also provide NP algorithms, in addition to several matrix factorization implementations. LensKit serves as a comprehensive platform for experimenting with recommendation systems [12].

4.2 Agent-Based Model

To investigate the impact recommendation systems have on misinformation propagation, an ABM approach is adopted. Initially, empirical data-driven approaches using real-world datasets were considered for this research. However, because of the absence of suitable and available datasets that included all necessary features for the experiments, this approach was discarded.

The ABM approach is chosen in part because of its flexibility in incorporating external components such as RAs, as well as its use of synthetic data generation and simulation modeling. Unlike alternative solutions for modeling social dynamics, such as ODEs, ABM also offers a framework for the implementation of heterogeneous agents and complex modeling of social networks and interactions. Due to these necessities and since making the simulation model as realistic as possible is central to this thesis, ABM serves as the most fitting choice.

The Mesa framework is adapted to implement the ABM simulation. The framework is chosen because of its modern implementation and the authors' familiarity with the Python programming language. Mesa also allows integration of external components, such as the LensKit library for out-of-the-box RAs.

The following assumptions are made when creating the model:

- The social network structure is static during the simulation,
- All agents start in the Susceptible state,
- A timestep (i.e., tick) represents 3 hours,
- A user shares news content based on how similar the topic is to their preferences, their naivety level, and how engaging the content is,

4.2.1 Model Specification

The model simulates the spread of misinformation in a social network, capturing the dynamics between different types of agents, content recommendation, and information dissemination. An overview of the model and the different classes, with

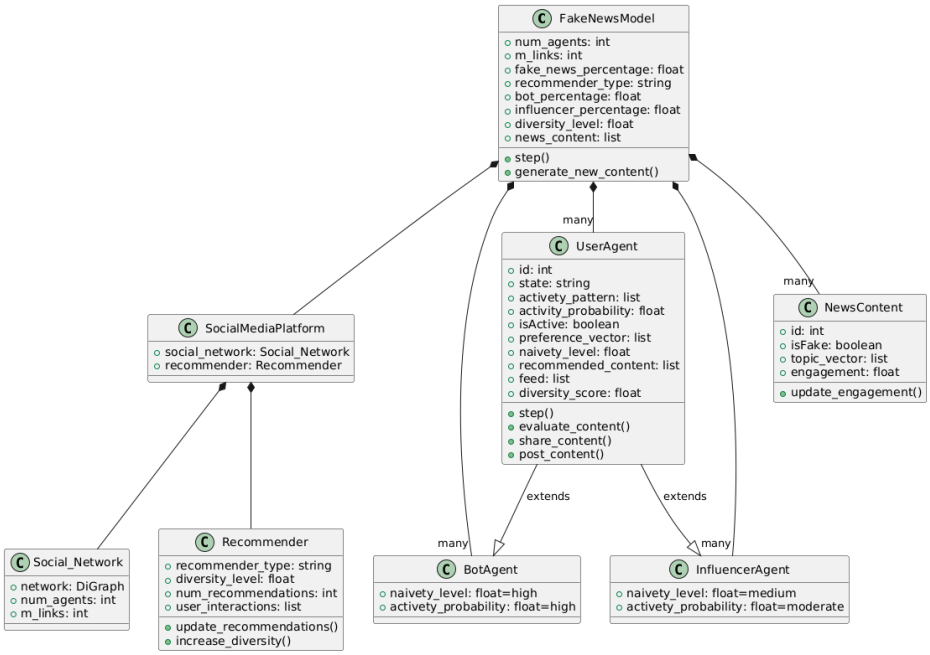

Figure 4.2: Class diagram of the model

their respective attributes, is presented in Figure 4.2. The model integrates several key components that work together to simulate the social media environment:

Fake News Model: The main simulation engine that initializes the environment, creates agents, and orchestrates the propagation of both regular content and misinformation through the network at each time step.

Social Network: A directed graph in which each node represents an agent and each edge represents a "follow" relationship, defining how information flows between users.

News Content: A collection of items, each annotated with a topic vector, an engagement score that decays over time, and a flag indicating whether it is misinformation. New items are injected periodically.

Social Media Platform: The overarching system that manages the social network structure and the content recommendation process, acting as the intermediary between agents and News Content.

Recommender System: The algorithmic component responsible for selecting which content to display to each agent, based on their preferences and the chosen recommendation strategy.

User Agent: A generic agent that models an individual user, characterized by topic preferences, activity patterns, and a level of naivety. Users consume, evaluate, share, and post content.

Bot Agent: A highly active and specialized agent designed primarily to amplify the spread of misinformation by generating and sharing large volumes of content.

Influencer Agent: A high-reach agent that simulates the behavior of influential users whose shares can dramatically boost a post's visibility across the network.

To make the model adaptable for testing multiple scenarios, several model parameters are added which control its behavior. All model parameters are described in Table 4.1.

4.2.2 User Agents

The agent population consists of regular users, bots, and influencers, and the distribution is affected by the parameters $bot\_percentage$ and $influencer\_percentage$. Agents vary in their activity patterns, naivety levels, and preferences. The preference vector is generated at random with a length of 8 and is then normalized.

Table 4.1: Model parameters

Parameter	Description	Default Value
num_agents	Number of agents in the network	200
m_links	Number of links per agent	10
news_amount	Initial amount of news content	500
fake_news_percentage	Base percentage of misinformation items in content pool	10%
recommender_type	Algorithm for content recommendations	Rnd
bot_percentage	Percentage of agents that are bots	5%
influencer_percentage	Percentage of agents that are influencers	5%
diversity_level	Content diversity in recommendations	0
num_recommendations	Number of items recommended	10
use_stored_network	Should the model use a stored network	False

The number 8 is chosen to ensure enough variety between the vectors, but not enough variety so that the differences between similarities become very small.

The probability of posting and sharing misinformation is modeled as lower for influencers and regular users compared to bots. Bots' intention is often to spread misinformation, and therefore they post and share loads of it. Taking inspiration from [64], it is modeled that the population follows average behavior with high probability, where extreme behaviors, such as a very high or very low vulnerability to misinformation, occur with low probability. Thus, attributes for each agent, such as the level of naivety and the activity probability, are set to a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ containing different means μ and standard deviations σ^2.

When agents post content, the probability of the item being misinformation is calculated differently then during normal initialization. Bots have 1.5 times the model's $fake\_news\_percentage$ of posting misinformation, as bots' usually share large amounts of misinformation online. For regular users and influencers, the probability is multiplied with the agents' naivety level, which has a Gaussian distribution.

In this model, a user profile is abstracted to differentiate from others mainly

in terms of activity pattern, naivety level, and preferences. Demographic and geographic attributes such as age, gender, and location have been excluded due to time constraints in this study. The various compositions of the agents' activity patterns, naivety level, and preferences are assumed to provide a high-level representation of how demographics and geographic attributes affect user behavior online. The base agent attributes given in the simulation are presented in Table 4.2, while the attributes that vary depending on the agent type are shown in Table 4.3.

Table 4.2: Base agent attributes and initialization

Attribute	**Description**	**Default value**
preference_vector	Vector stating user preferences	Null
state	User's current state	S
infection_start_step	Timestep of eventual infection	Null
feed	Shared or posted news content	[*empty*]
recommended_content	List of recommended content	[*empty*]
diversity_score	Diversity score in recommendations	0
is_active	Flag indicating wether a user is active	False
last_active_step	Step when the agent was last active	Null

Table 4.3: Agent attributes by type

Attribute	**Regular User**	**Bot Agent**	**Influencer Agent**
activity_probability	$\mathcal{N}(0.3, 0.1)$	$\mathcal{N}(0.7, 0.15)$	$\mathcal{N}(0.5, 0.15)$
naivety	$\mathcal{N}(0.5, 0.15)$	$\mathcal{N}(0.9, 0.05)$	$\mathcal{N}(0.5, 0.15)$
activity_pattern	1–3 peak times	High activity	Strategic timing

4.2.3 News Content

At the start of the model, a number of news items are initially set based on the model parameter *news_amount*. These items are then randomly distributed to users to start propagation. At each step of the model, a random number of news

items is generated to simulate how new content is continuously posted to social media. These are then added to the news content pool, which is utilized by the RA when generating new feeds for all user agents. Agents also have the ability to post content. The reason behind generating new content in addition to users posting, is to ensure a sufficient content pool for generating quality recommendations. The properties of news content can be seen in Table 4.4.

Table 4.4: News content attributes

Attribute	**Description**
content	Unique identifier for each content item
isFake	Flag for misinformation, true with prob. *fake_news_percentage*
topic_vector	Vector representation of the content's topic
engagement	Novelty of content, 1.5 for misinformation; 1.0 for true news
creation_step	Set to current step at creation time

To allow users to interact with news content similar to their interests, all news items are provided with a random topic vector of the same dimension as the user agent's preference vector. This is used to generate recommendations and to influence whether an agent will interact with the content.

To enable analysis of how fake news spreads on the social network, all news items are also given a Boolean $isFake$ attribute to determine their veracity. Whether the content is set as misinformation or not depends on the model parameter $fake\_news\_percentage$. The implemented check looks like this: **if** $random(0, 1) < fake\_news\_percentage$ **then** True **else** False. Additionally, some content topics have a higher probability of containing misinformation than others; for example, there is more misinformation about presidential campaigns than on the topic of a newly released movie [16]. This dynamic is implemented by defining the first 2 indexes in the vector as topics more likely to be fake. If the sum of the values in the first two indexes is greater than 0.3, the content has twice as high a probability of being misinformation ($2 \cdot fake\_news\_percentage$). The threshold 0.3 was decided based on the assumption that if the normalized topic vector's first two indexes summarized up to almost a third of the vector, the content contains enough information on topics likely to be misinformation for the model to actually increase the probability of it being misinformation itself.

Each content item is also given an engagement variable that decays over time to simulate how news is more engaging when it is first released and declines the longer it has existed in the content pool's life cycle. As misinformation is more engaging than regular news [23], misinformation has an initial engagement value

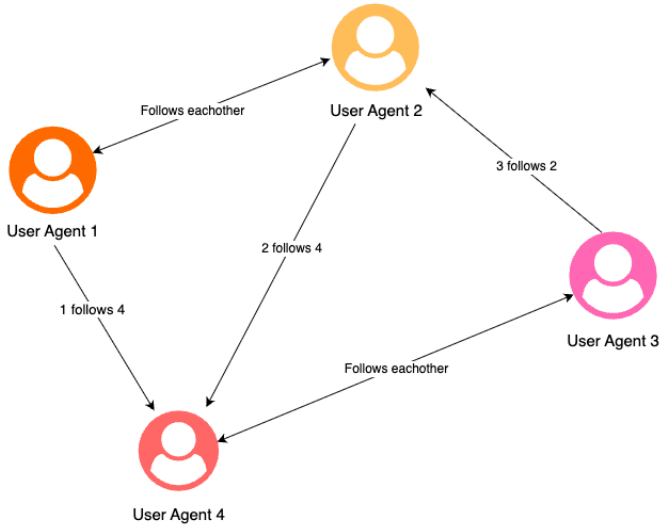

Figure 4.3: Visual representation of social network

of 1.5, while regular content has an initial value of 1.

4.2.4 Social Network

The social network is modeled as a directed graph using the NetworkX library. The nodes on the network correspond to user agents, and the directed edges represent that a user follows another user, as visualized in Figure 4.3. A directed graph is appropriate in this context because in real Online Social Networks (OSNs), one user can follow another without being followed back.

This approach creates a synthetic network that resembles real OSNs, with particular focus on preference-based connections and the distinct roles of regular users, influencers, and bots. The number of users, or nodes, to be added to the social network, as well as the average number of edges between regular users, are passed as the model parameters num_agents and m_links.

In real life, users are more likely to follow others who have similar interests [22]. Therefore, the implemented clustered network accounts for users' preferences. When the network is created, the initial step is to generate a list of preference vectors of the same length as the number of users. Then a similarity matrix is created that calculates the similarity between all pairs of preference vectors, which is used to calculate the probability of a potential edge between nodes. The more

similar the preferences between two users are, the greater the possibility is for an edge to be added to the network.

Observing real OSNs, some agents have notable more followers than an average user. These are the agents referred to as influencers, as they have the power to influence a significant number of people [17]. This study assumes that influencers are usually more strict with whom they follow; therefore, it is implemented that they follow half the number of accounts as regular users. Regular users are assumed to be four times more likely to follow influencers than to follow regular users. These implementations are supposed to simulate influencers' reach on OSNs. Follower counts are adjusted to follow a power-law distribution, where few users have many followers, and many users have few followers.

Bots are automated accounts that have a high posting rate and follow more people than average users [19]. Bots are assumed to follow approximately three times as many accounts as regular users. The model ensures that bots receive relatively few followers compared to regular users and influencers, reflecting their limited organic appeal. A strong asymmetry in connection probabilities is implemented:

- Influencers almost never follow bots ($P \approx 0.001$)
- Regular users rarely follow bots ($P \approx 0.01$)
- Bots aggressively follow both regular users and influencers

The different following and follower specifics of the agents are accounted for in the creation of the social network, increasing the realism and social dynamics in the model. The probability of an agent i following another agent j is determined by:

$$P(i \to j) = f(S_{ij}, T_i, T_j)$$

where S_{ij} is the similarity of the preference vector between agents, T_i is the type of agent (the type of agent with the outgoing connection), and T_j is the target type (the type of agent possibly being followed).

To be able to compare different RAs' performance in a social network, it is essential to reuse the same network for valid results. Therefore, at each initialization the social network is stored in a local file, and if the model parameter *use_stored_network* is set to True, the model will fetch and use the last stored network, instead of generating a new one.

4.2.5 Model Execution

When the model is first initialized, it performs several setup steps to further initialize the other components, as it works as the main class deciding how and

when the other objects interact. The model uses the parameter values when initializing other objects, such as the number of agents or the RA. The model also has a step function, which outlines what functionality should occur at each timestep, and this is therefore the most central function in the simulation. Both the initialize function and the step function are described in pseudocode in Algorithm 1.

Function `Initialize`(*parameters*)**:**
  Validate parameters (*parameters*);
  Set model parameters and initialize tracking variables;
  Generate preference vectors $P = \{p_1, p_2, ..., p_N\}$ for len *num_agents*;
  Initialize SocialMediaPlatform, Social Network and Recommender;
  for $i \leftarrow 0$ **to** *Num_Agents* **do**
    Create agent based on type distribution (User/Bot/Influencer);
    Place agent in network position i;
  end
  Generate initial news-content pool;
  Distribute initial news to agents;
  Initialize datacollector for metrics tracking;
Function `Step()`**:**
  Extend news content with 50 new items;
  for *each content* $\in$ *news_content* **do**
    content.engagement $\leftarrow$ updated based on current step;
  end
  Filter *news_content* $\leftarrow$ $\{$*content* $\in$ *news_content* $|$ *content.engagement* $> 0.2\}$;
  for *agent* $\in$ *Agents* **do**
    Update recommendations using recommender;
  end
  Shuffle agents and execute *agent.step*() for each agent;
  Collect data metrics for current timestep;

Algorithm 1: Model class initialization and step function

User Agents

During the execution of the model's step function, all agents' step functions are also run. This function outlines all actions performed by the agents and how they interact with the environment. The step function is where users browse their feeds, containing recommendations and shared content, and evaluate all items. The step function is defined in the pseudocode in Algorithm 2.

```
Function Step():
    if state = I and timeSinceInfection ≥ 40 then
        Change state I → S
    end
    Determine activity state based on time patterns;
    if agent is active then
        for content ∈ feed + recommendations do
            if content is fake then
                Change state S → E
            end
            evaluate content(content);
            if evaluatedContentScore > 0.6 then
                Interact with content, but not share;
                if content is fake then
                    Change state E → I
                end
            end
            if evaluatedContentScore > 0.8 then
                share content to all followers;
            end
        end
        Potentially create and post new content;
        Clear processed content;
    end
```

Algorithm 2: User agent step function

State Transitions

Agents' state transitioning is modeled based on the epidemiological model SEIS, explained in Section 2.4.1, but ODEs are replaced with agent-level transition rules to fit the ABM approach.

Although LTM and ICM are valuable conceptual tools and can be well supported by ABM's, the SEIS model is central to the simulation framework of this study. This is because of its ability to simulate how misinformation spreads by capturing repeated exposure to misinformation and the possibility that agents can return to a Susceptible state. ICM, LTM, SIR and SEIR assume that once agents enter an Infected state, they will remain in that state, which is an unrealistic representation of how misinformation propagation works. Using SEIS with ABM also allows inclusion of heterogeneity in the network, where some agents are more susceptible than others, and some agents are more active on social media overall. This provides greater flexibility to make the simulation as realistic as possible.

The process of states starts when the agent evaluates each content in the feed, which in this case includes recommended content. If any of the content in the feed is classified as fake, then the state will change from *Susceptible* to *Exposed*. In addition, the agent will evaluate the content based on how similar the topic is to the user's preferences, the engagement factor of the content, and the naivety level of the agent. If the evaluated score is above a certain threshold (0.6 for liking and 0.8 for sharing), the agent interacts with the content. The thresholds are chosen based on assumptions, which are inspired by a 5-star rating system. If a user likes an item, they at least would give it 3 out of 5 stars (based on assumptions), which can be quantified to 60%. Taking this further, if a user shares an item, the threshold is even higher than with liking. It is plausible that users rating an item 4 out of 5 stars would also share it to their followers, which can be quantified to 80%.

If the content with which the agent interacts is misinformation, the agent's transition from state *Exposed* to $Infected$. To further maintain realism in the model, the $Infected$ state is avoided as final. Therefore, if an agent has not interacted with misinformation for 40 timesteps, they return to the *Susceptible* state. Choosing 40 timesteps as the recovery time was an assumption made to ensure that agents spend enough time in the $Infected$ state before returning to the *Susceptiblestate* to avoid chaotic results that are hard to decipher. A complete overview of the state transition model is presented in Figure 4.4.

4.2.6 Time Dynamics

A central advantage of conducting experiments using ABM simulations is the ability to analyze how the results change over time. The model includes two central aspects of social media time dynamics.

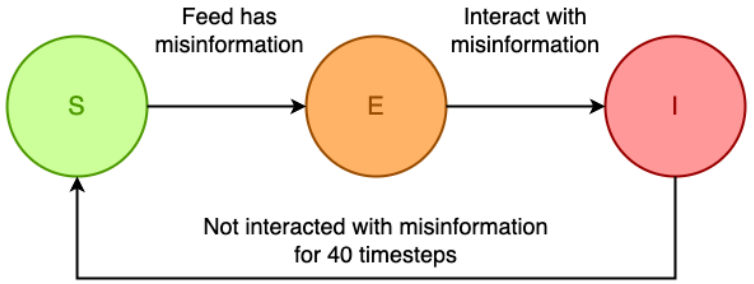

Figure 4.4: The implemented SEIS model for state transitions

Activity Pattern

The simulation works in discrete timesteps, where at each timestep, the group of active agents is updated. Agents' execution of the timestep is coordinated by using Mesa's built-in scheduler, ensuring random execution to avoid order bias in the results. To reflect agents' variations in active intervals due to time zones or general usage patterns, user activity is implemented as a time-dynamic aspect. All users are given a peak activity pattern upon initialization. A 24-hour activity pattern is conducted and divided into 8 time slots, as each timestep is defined as 3 hours. Every agent has an activity pattern where regular users have 1-3 peak activity times, meaning that they have a probability of being active in all time slots, and 1-3 of them have an extra high probability. This is based on the assumption that regular users have between 1 and 3 regular peak times to be active on social media, due to either a time schedule or habit.

Bots have continuously high levels of activity on social media at all times, without distinct peaks [20]. Influencers are more strategic about their social media activity and carefully manage their content to make a greater impact [18]. At each timestep, agents update their activity state based on their activity pattern and their last active step, reflecting the increased probability of becoming active after a period of inactivity. The content accumulates in the agent's feed and recommendations even if the agent is not active for several steps, so the more active the agent, the less content it will have to evaluate each time. Only agents that are set to active after updating their state will proceed to evaluate their content feed.

Content Engagement Decay

Inspired by [64], the model simulates the natural decline in the discourse of news as its relevance decreases over time. This is done by giving all news items an

engagement variable that decays at each timestep. All news items update their engagement variable using the exponential decay function, which is formulated:

$$\mathrm{E(t)} = E_0 \cdot e^{-\lambda \times t} \tag{4.1}$$

Where E_0 is the initial engagement, and λ is the decay rate set to 0.1. Using this exact decay rate means the half-life of the engagement is approximately:

$$t_{1/2} = \frac{\ln(2)}{0.1} \approx 6.93 \text{ time units} \tag{4.2}$$

corresponding to a drop in engagement by 50% after 7 units of time, which is equal to 21 hours in this model. This is assumed to be a reasonable timescale for news items and social media posts, which most often lose their engagement factor within a few days due to the frequent publishing of novel and more interesting content. In every timestep, the model filters through the content pool and removes content having a lower engagement than 0.2 to ensure that users receive only relevant and timely content.

4.2.7 Recommender Systems

The analysis includes a variety of classical RAs. To simulate the different algorithms, LensKit has been used for the CF methods, while the other algorithms have been manually implemented. The chosen algorithms are described below, organized under NP, CF, and CBF approaches:

NP Pop: Ranks content according to recent high engagement and virality.

Rnd: Serves a random selection of content to agents (excluding items already seen), used for cold-start scenarios or as a baseline.

CF Recommends items based on how other users with similar preferences have engaged with content. Two different CF algorithms are used:

IB-CF: Recommends content similar to items the user previously engaged with, based on interaction patterns across the user base.
UB-CF: Suggests content based on the preferences of users with similar interaction histories. Identifies user neighborhoods to predict relevant content.

CBF Uses cosine similarity between the vectors of the content topic and the agent's preference vector to rank items.

A more in-depth explanation of the different recommendation strategies is presented in Section 2.2.

User-Item Matrix

The RAs usually base recommendations on user preferences or previous interactions. To track agents' interactions over time, a global user-interaction matrix is implemented. Interactions are recorded whenever an agent shares or likes content, and the entry consists of:

- `user_id`: agent position in the grid
- `item_id`: content identifier
- `rating`: the computed content evaluation score

Table 4.5 shows an abstract representation of the interaction matrix. Each cell (i, j) represents the score evaluated (if the user interacted with the content) from the user u_i to item v_j.

Table 4.5: User-item interaction matrix

User / Item	**News 1**	**News 2**	**News 3**	...
User A	0.85	—	0.62	...
User B	—	0.79	—	...
User C	0.65	—	0.58	...
⋮	⋮	⋮	⋮	⋱

Most users interact with only a small subset of available content, making the matrix inherently sparse.

Implementation details

The CF algorithms are implemented using the LensKit library. Specifically, `ItemKNNScorer` with an `ItemKNNConfig` is used for IB-CF, and `UserKNNScorer` with an `UserKNNConfig` is used for UB-CF [12]. The following parameter values are chosen for the configuration of the algorithms:

- max_nbrs = 20: This defines the maximum number of nearest neighbors considered for computing a prediction. A value of 20 provides a good trade-off between locality and generalization, especially in sparse interaction environments.

- min_nbrs = 1: Setting the minimum number of neighbors to 1 ensures that predictions can still be made even when limited historical data is available for a user or item. This is particularly important in the early stages of the simulation or for new agents.
- min_sim = 0.1: This threshold ensures that only neighbors with a non-trivial level of similarity are considered. A cutoff of 0.1 filters out weak correlations and can avoid noise from neighbors who are not as relevant without being too strict.
- feedback = 'explicit': Since the simulation stores a numeric evaluation score for the content at each interaction, the 'explicit' feedback setting is chosen to more accurately model these interactions.

LensKit is used to manage the internal computation of neighborhood-based scores and generate top-N recommendations from the user-item interaction matrix. The training process constructs a similarity matrix $S \in \mathbb{R}^{n \times n}$ for items (or $S \in \mathbb{R}^{m \times m}$ for users in UB-CF) and identifies the k neighbors most similar for each item/user. Training is triggered conditionally every time the user-interaction matrix is updated since the last training step. Especially in the early stages of the simulation, due to the cold start problem, there is little interaction on which to base the recommendations. Therefore, a minimum number of interactions threshold is set before continuing to use the CF algorithms. This minimum interaction threshold is given by:

$$\max(150, \frac{|A|}{2}) \tag{4.3}$$

to ensure that the algorithms are trained on sufficient interactions before providing recommendations. The limit is set to at least 150 interactions, but to make the threshold scale with the model, it will switch to using half of the number of agents ($\frac{|A|}{2}$) when the number of agents exceeds 300. The threshold values are based on assumptions, which are estimated to ensure a good foundation of interactions before training the models. Until the user-item matrix includes a number of interactions that pass the threshold, the Rnd algorithm is used to generate recommendations, ensuring news content's availability to user agents in the early stages.

An abstracted overview of the UB-CF and IB-CF implementation is outlined in Algorithms 3 and 4.

TheRnd, CBF, and Pop algorithms are implemented without the use of libraries from external recommendation systems. Abstract algorithmic descriptions of Rnd is described in Algorithm 5, CBF is shown in Algorithm 6, and Pop is presented in Algorithm 7.

Function `IB-CF(`*agent*`)`**:**
  if *not enough user interactions* **then**
    return `Rnd(agent)`;
  end
  Train item-KNN model on interaction data;
  candidateItems ← content not in agent's feed;
  recommendations ← use item-KNN algorithm to generate top k items;
  return *recommendations*;

Algorithm 3: Item-based collaborative filtering

Function `UB-CF(`*agent*`)`**:**
  if *not enough user interactions* **then**
    return `Rnd(agent)`;
  end
  Train user-KNN model on interaction data;
  candidateItems ← content not in agent's feed;
  recommendations ← use user-KNN algorithm to generate top k items;
  return *recommendations*;

Algorithm 4: User-based collaborative filtering

Function `Rnd(`*agent*`)`**:**
  availableContent ← content not in agent's feed;
  recommendations ← randomly select k items from *availableContent*;
  return *recommendations*;

Algorithm 5: Random recommendation

Function `CBF(`*agent*`)`**:**
  availableContent ← content not in agent's feed;
  for *item* ∈ *availableContent* **do**
    $similarity[item] \leftarrow \cos(agent.preferenceVector, item.topicVector)$;
  end
  recommendations ← top k items sorted by *similarity*;
  return *recommendations*;

Algorithm 6: Content-based recommendation

Function `Pop(`*agent*`)`**:**
  for $item \in availableContent$ **do**
    $popularityScore \leftarrow$ interaction count for item;
    $recencyFactor \leftarrow \exp(-\lambda \cdot \text{age})$;
    $preferenceSim \leftarrow \cos(\text{agent.preferenceVector}, item.topicVector)$;
    $noveltyBoost \leftarrow 1 + 0.5 \cdot \exp(-0.1 \cdot \text{views})$;
    $exploration \leftarrow$ small random factor;
    $finalScore[item] \leftarrow$ weighted sum of all factors;
  end
  $recommendations \leftarrow$ top k items by $finalScore$;
  return $recommendations$;

Algorithm 7: Popularity-based recommendation

4.2.8 Simulation Dashboard

Mesa has built-in functionality for a browser-based front-end visualization, which creates an interactive dashboard of the ABM [11]. The dashboard is a helpful tool for model development, analysis, and presentation and can be used to test and visualize the ABM. Model parameters are displayed on the sidebar and can be configured by a user to test different combinations of parameters. The parameter configurations are presented in Appendix A.1.

The actual dashboard can be configured by using Mesa's default plotting functions for metrics, or users can create custom plotting functions or visualizations. The dashboard created for this study incorporates a project description, visualization of the social network, and timeline plots of different metrics collected by the DataCollector. An overview of the dashboard is visualized in Appendix B.1.

4.2.9 Optimization and Efficiency Strategies

RAs, especially the ones that use the user-item matrix, turn out to be computationally intensive, which affects the run-time of the simulation. To ensure that the simulation remains efficient and scalable, certain performance optimization tactics are implemented. The most significant performance enhancement is caching the user-item matrix and only regenerating it when the number of user interactions has increased since the last training step. This avoids unnecessary retraining and preserves computation time. Additionally, the content pool and associated directories (e.g., ID-to-content mappings) per simulation step are cached, ensuring repeated access to the same content data structures remains efficient. To eliminate unnecessary calculations, each recommendation process starts by removing all

content already present in the users' feeds and recommendation lists from the candidate items.

For vectorized computations, operations from Python libraries such as NumPy and Scikit-learn are used for batch similarity computation, which improves runtime performance. In addition, due to the cold start problem, the system defaults to the Rnd algorithm if there is insufficient data available instead of attempting complex recommendations when the foundational training data is limited.

4.3 Increasing Diversity

To improve the diversity of recommendations, a modified version of the MMR algorithm, described in Section 2.2.4, is implemented. MMR is originally proposed by [35] as a method to reduce redundancy while maintaining relevance in information retrieval systems. The technique is adapted specifically for news recommendation to ensure that users receive diverse content while preserving relevance to their interests.

MMR, a post-processing approach, is chosen for several factors. Pre-processing approaches are mainly about balancing the training data before training the RAs, and since the model is simulated, there is no access to training data before the simulation is running. Several of the implemented RAs, such as Pop, Rnd, and CBF, are not trained on data, which makes pre-processing an unfitting approach in this case. In-processing methods involve making adjustments to the algorithms themselves, and it is therefore difficult to implement a common diversity increase function that can be applied across all algorithms. To choose the post-processing method, both MMR and DPP, described in Section 2.2.4, were considered. Although DPP initially seemed like a suitable choice due to its low computational cost, it lacks a tuning parameter to examine different levels of diversity, which is desirable in these experiments.

When re-ranking recommendations to increase diversity, a larger subset than normal is needed when choosing k recommendations. Therefore, it is implemented that if the $diversity\_level$ parameter is > 0, the RAs will choose top-$(N * 3)$ recommendations before sending those to the re-ranking MMR function, which then outputs a list of N recommendations.

Diversity increase is applied to all RAs except Rnd. This decision was based on Rnd being the baseline for comparison, and increasing diversity would contaminate this baseline by introducing algorithmic bias, as in experimental design isolated variables are desirable. Rnd is also inherently diverse, and adding diversity re-ranking would just introduce higher computational complexity.

Constrained MMR for Guaranteed Diversity Improvement

Although the traditional MMR approach balances relevance and diversity, it does not guarantee that the final recommendation set will be more diverse than a purely relevance-based selection. Since consistently increasing diversity is essential in this study, these limitations are addressed using a restricted version of the MMR that ensures that the diversity of recommendations meets or exceeds a baseline threshold.

The algorithm first calculates the diversity of the top-k recommendations that would be selected using relevance alone. This establishes a baseline diversity score D_{original}. Then, a two-phase selection process is implemented:

1. **Initial selection phase:** The algorithm starts by selecting the most relevant item and then applies the standard MMR for the first few items to establish a balanced starting set.

2. **Constrained selection phase:** For subsequent selections, each candidate item is evaluated not only by its MMR score but also by whether its inclusion would maintain or improve overall diversity compared to D_{original}.

Mathematically, the diversity of a set of items S with topic vectors $\{v_1, v_2, \ldots, v_n\}$ is defined as:

$$D(S) = 1 - \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \text{cosine\_similarity}(v_i, v_j) \tag{4.4}$$

This measure represents the average dissimilarity between all pairs of items in the set. Higher values indicate greater diversity.

The constrained selection criterion can be formalized as:

$$\text{Select } d_i \text{ if } D(S \cup \{d_i\}) \geq D_{\text{original}} \tag{4.5}$$

If no candidate satisfies this constraint, the item that maximizes diversity is selected:

$$\text{Select } d_i = \arg\max_{d_i \in R \setminus S} D(S \cup \{d_i\}) \tag{4.6}$$

Diversity-Relevance Trade-off Parameter

The implementation includes a diversity level parameter (corresponding to $1 - \lambda$ in the MMR formula) that controls the balance between relevance and diversity. Higher values prioritize diversity, while lower values favor relevance to user preferences. This change is added to make it more user-friendly, as a higher

diversity level corresponds to a greater increase in the diversity score. Using this parameter enables testing different levels of diversity and analyzing its implications.

In cases where the constrained MMR approach fails to select a sufficient number of items (which may occur in extremely constrained scenarios), the algorithm falls back to a pure diversity maximization approach. This ensures that the system always returns the requested number of recommendations while maintaining the highest possible diversity.

4.3.1 Implementation Optimizations

RA are computationally intensive on their own, but adding the re-ranking functionality further increases the complexity of the recommender. To improve computational efficiency, several strategies are implemented. Numpy's vectorized operations are used to calculate the similarities and MMR scores for multiple items simultaneously. A caching mechanism for diversity scores is also implemented to avoid recalculating diversity for previously evaluated item combinations.

The candidates are evaluated in descending order of the MMR score, and the first candidate that satisfies the diversity constraints is accepted to reduce unnecessary computations. For the fallback diversity maximization approach, the remaining items in the batches are processed to balance memory usage and computational efficiency.

4.4 Metrics

To answer the research questions, several metrics are utilized to quantify the results of the experiments. These metrics are defined in the following sections.

4.4.1 Infection Rate

A central aspect of this study is misinformation propagation, and therefore the Infection Rate (IR) metric is used to evaluate how users interact with misinformation. This is calculated by detecting the number of agents in the state "Infected," which means that they have interacted with misinformation, as can be seen in Equation 4.7.

$$\mathrm{IR} = \left(\frac{A_{\text{infected}}}{|A|} \right) \times 100\% \tag{4.7}$$

Where A_{infected} is the number of infected agents, and $|A|$ is the total number of agents.

4.4.2 Misinformation Ratio Difference

To further evaluate the impact of RA on the exposure of misinformation, Misinformation Ratio Difference (MRD) is calculated, the difference between the misinformation rate in the recommendations and the total misinformation [8]. MRD measures the degree to which RAs amplify or reduce misinformation compared to their natural prevalence in the content pool.

MRD is defined mathematically as:

$$\text{MRD} = \frac{1}{|A|} \sum_{a \in A} (r_a - r_{\text{overall}})$$

Where:

- A is the set of agents that have at least one item in their recommendation list
- $r_a = \frac{\sum_{c \in R_a} \mathbf{1}_{c.\text{isFake}}}{|R_a|}$ is the ratio of fake news in agent a's recommendations
- $r_{\text{overall}} = \frac{\sum_{c \in C} \mathbf{1}_{c.\text{isFake}}}{|C|}$ is the overall ratio of fake news in the entire content pool C
- R_a is the set of recommended content items for agent a
- $\mathbf{1}_{c.\text{isFake}}$ is an indicator function that equals 1 if content item c is fake news, and 0 otherwise

The interpretation of MRD values is straightforward. If MRD > 0, the RA amplifies misinformation by exposing users to a higher proportion of misinformation than exists in the content pool. If MRD < 0, the RA reduces the spread of misinformation.

This metric is particularly valuable for evaluating whether recommendation systems exacerbate the misinformation problem by disproportionately promoting it or whether they help mitigate it by filtering out misinformation.

4.4.3 Misinformation Count

Another metric for analyzing how each RA affects user exposure to misinformation, is Misinformation Count (MC). MC is the count of misinformation items recommended to each user [8]. This metric provides a direct measure of the amount of misinformation that is distributed through the recommendation system. Whereas MRD is RAs' amplification of misinformation relative to the content pool, MC only cares about the actual number of misinformation items in the recommendation list.

Mathematically, MC, using an average approach, is defined as:

$$\text{MC} = \frac{\sum_{a \in A} \sum_{c \in R_a} \mathbf{1}_{c.\text{isFake}}}{|A|}$$

Where:

- A is the set of all agents in the model
- R_a is the set of recommended content items for agent a
- $\mathbf{1}_{c.\text{isFake}}$ is an indicator function that equals 1 if content item c is fake news, and 0 otherwise
- A is the set of agents that have at least one item in their recommendation list

The numerator counts the total number of fake news items in all recommendation lists, while the denominator normalizes by the number of agents. Higher values indicate that more misinformation is recommended to users in the system.

4.4.4 Echo Chamber Effect

To measure the Echo Chamber Effect (EC) within the simulated social media environment, a computational approach was developed that quantifies the similarity of content consumed within the detected communities. This method enables tracking the evolution of the echo chambers over time and evaluating the effectiveness of various intervention strategies.

Community Detection

The first step in the EC measurement involves identifying cohesive communities within the social network. A dual-approach community detection strategy is implemented:

$$\mathcal{C} = \text{CommunityDetection}(G) \tag{4.8}$$

where G represents the social network graph and $\mathcal{C}$ is the resulting community partition. The implementation prioritizes the Infomap algorithm, first published in [67], which detects communities by minimizing the expected description length of a random walker's trajectory:

$$L(M) = q_{\curvearrowright} H(\mathcal{Q}) + \sum_{i=1}^{m} p^i_{\circlearrowright} H(\mathcal{P}^i) \tag{4.9}$$

where $L(M)$ is the description length, $q_{\curvearrowright}$ is the probability of moving between modules, $H(\mathcal{Q})$ is the entropy of movement between modules, $p^i_{\circlearrowright}$ is the probability of movement within module i, and $H(\mathcal{P}^i)$ is the entropy of movement within module i [67].

As a fallback mechanism, the Louvain method, first published in [68], is implemented. This algorithm maximizes modularity through a hierarchical approach. To adapt this method for the directed social network, the network is transformed into a weighted undirected graph that preserves directional information:

$$w(u,v) = \begin{cases} 2.0 & \text{if } (u,v) \in E \text{ and } (v,u) \in E \\ 1.0 & \text{if } (u,v) \in E \text{ and } (v,u) \notin E \end{cases} \tag{4.10}$$

where $w(u,v)$ represents the weight of the edges between nodes u and v, and E is the set of edges in the original directed graph. This weighting scheme assigns greater importance to mutual connections, which typically indicate stronger social bonds. This algorithm was not chosen as the first priority approach due to its need for undirected networks, and the improvements for directional information might not be sufficient for optimized algorithm performance.

Content Similarity Within Communities

Once communities are identified, the similarity of the recent content that circulates within each community is measured. The recent content consists of the last 30 items in an agent's feed, which are either recommended or shared by the users the agent follows. For each community $c \in \mathcal{C}$, the topic vectors are collected of all recent content items by community members:

$$V_c = \{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n\} \tag{4.11}$$

where $\vec{v}_i$ represents the topic vector for a content item within the community c.

The average pairwise cosine similarity is then calculated between these content vectors:

$$S_c = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{\vec{v}_i \cdot \vec{v}_j}{|\vec{v}_i||\vec{v}_j|} \tag{4.12}$$

where S_c represents the similarity of content within the community for community c. Higher values of S_c indicate that members of the community are exposed to more similar content, suggesting the presence of an echo chamber effect.

Beyond the EC score, this implementation captures additional community-level metrics that provide deeper insights into the dynamics of misinformation

propagation. For each community, the system stores the size of the community, the echo chamber score, and the MR observed in the recent content of the community. MR is the amount of misinformation content in recent content divided by the total number of recent content.

Global Echo Chamber Score

To obtain a global measure of echo chamber strength across the entire network, the weighted average of within-community similarities is calculated, where each community's contribution is proportional to its size:

$$\mathrm{EC} = \min\left(\frac{\sum_{c\in\mathcal{C}} |c| \cdot S_c}{\sum_{c\in\mathcal{C}} |c|}, 1.0\right) \tag{4.13}$$

where $|c|$ represents the size of the community c and S_c is the similarity of the content within the community for that community. This weighted approach ensures that larger communities, which affect more users, have a proportionally greater influence on the global metric. The score is normalized to the range $[0, 1]$ for easier interpretation, with higher values indicating stronger echo chamber effects throughout the network.

This weighting methodology provides a more representative measure of the overall echo chamber effect experienced by users in the system, rather than treating each community as equally important regardless of its size. It also reduces the sensitivity of the metric to potential outliers in very small communities, where a few users sharing identical content could otherwise disproportionately affect the global score.

To optimize computational performance in the simulation, several efficiency measures are implemented. Echo chamber measurements are computationally intensive, so the score is calculated at each fifth simulation step rather than at each step. For intermediate steps, the value most recently calculated is reused. Also, since the social network is static, communities are detected once, and then the results are cached for subsequent calculations. This significantly reduces computational overhead while maintaining measurement accuracy.

4.4.5 Diversity score

The Diversity Score (DS) metric quantifies the dissimilarity among content items in a recommendation list, measuring topic variation to evaluate whether a recommendation system provides users with diverse content or creates a narrow information bubble. Mathematically, DS can be defined as:

$$\mathrm{DS} = 1 - \frac{\sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \mathrm{sim}(v_i, v_j)}{n(n-1)} \tag{4.14}$$

Where:

- n is the number of content items in the recommendation list
- v_i and v_j are the topic vectors of content items i and j respectively
- $\text{sim}(v_i, v_j)$ is the cosine similarity between the topic vectors of items i and j
- $n(n-1)$ represents the total number of pairwise comparisons between different items

The implementation first calculates a similarity matrix using cosine similarity between all pairs of topic vectors. The diagonal elements (self-similarity) are excluded since they are always 1 and do not contribute to measuring diversity. The average similarity between different items is then calculated, and the diversity score is defined as 1 minus this average similarity. A higher diversity score indicates greater dissimilarity among the recommended items:

- DS ≈ 1: Very diverse recommendations with little topical overlap
- DS ≈ 0: Very similar recommendations with high topical overlap

This metric is particularly useful for evaluating how RAs balance between relevance (recommending items similar to what users already like) and diversity (exposing users to a broader range of content).

Diversity Increase

The Diversity Increase (DI) metric is calculated as a percentage increase from the original diversity score to the new diversity score. The mathematical formula is:

$$\text{DI} = \frac{D_{\text{new}} - D_{\text{original}}}{D_{\text{original}}} \times 100\% \tag{4.15}$$

Here D_{new} is the DS after re-ranking, and D_{original} is the DS before re-ranking. This percentage represents the increase in diversity in recommendations after applying diversity re-ranking algorithms and can be used to evaluate the effectiveness of the re-ranking.

Chapter 5

Experiments

This chapter introduces the experiments conducted in this study to answer the research questions, which are presented in Section 1.3. The experimental setup is explained in detail in Section 5.1. The first experiment is outlined in Section 5.2, and encompasses exploring how the different RAs affect the propagation of misinformation in a social network and is linked to RQ1. The second experiment in Section 5.3, involves increasing diversity in recommendations to observe how this impacts the propagation of misinformation and is linked to RQ2. The final research question, RQ3, will be answered by using results from both experiments.

5.1 Experimental Setup

To evaluate the research questions, an ABM is used to simulate a social network and various RAs. A Python script is created to run the experiments, where Mesa's BatchRunner API is utilized for the parallel execution of experiments [11]. 5 iterations of the experiment are run because of implicit randomness in the model to ensure that the results are less prone to outlier data and edge cases. The BatchRunner is optimized to run several iterations of the model and, when provided with multiple parameter values, to run the simulation with all possible combinations of those given parameters. If a parameter list of 5 different RAs is provided and the number of iterations chosen to run is 5, the resulting number of simulation runs is 25. An iteration is therefore defined as one run of the whole experiment, which again has several simulation runs for each combination of the possible parameters.

Each iteration first initializes the ABM and stores the social network to be used in the sequential runs with different parameters. The network is reused across all runs with the different parameters to ensure that differences between

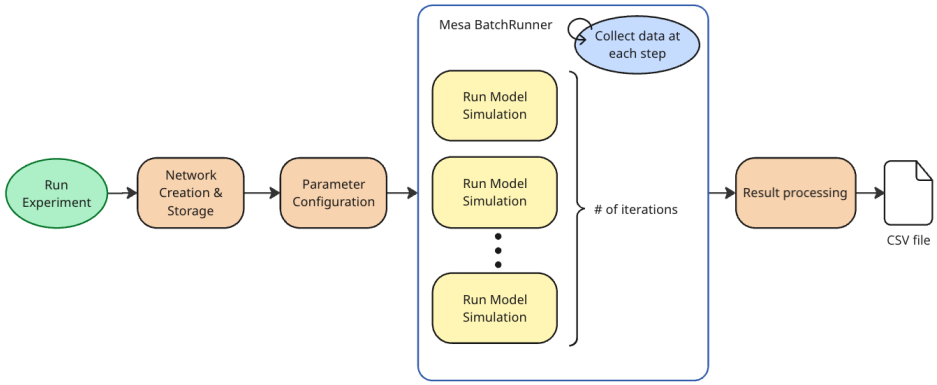

Figure 5.1: Experimental protocol

connections in the network do not affect the results. The model is then run with the BatchRunner API, with the parameters chosen for the experiment. When the experiment is finished, the data collected during the simulation is stored in a dataframe, which is then saved as a CSV file in the repository for further analysis. The whole experimental protocol is presented in Figure 5.1.

Data Collection

One of the main advantages of running a simulation to test the experiments is access to how data changes over time in the model. To gain insight into the progress of the model, the Mesa framework provides a DataCollector class that captures agent- and model-level data at each timestep [11]. At the end of the experimental run of all iterations, the data is stored in a CSV file for further analysis and plotting. The DataCollector not only calculates aggregated results at the end of the simulation but also tracks value changes over time, which can be observed in the simulation. The data collected at each step of the simulation are described in Table 5.1.

In addition to collecting model- and agent-level data, community-level data is also stored to enable analysis of whether there are any big differences between communities in the social network. The echo chamber scores and misinformation ratios of the communities are described in Section 4.4.4, and the complete overview of the data stored at the community level is presented in Table 5.2.

Table 5.1: Collected data per simulation step

Metric	Level	Description
Num Infected	Model	Number of agents in state I
Num Susceptible	Model	Number of agents in state S
Num Exposed	Model	Number of agents in state E
Avg DS	Model	Average diversity score in recommendations
Avg MC	Model	Average misinformation items per agent
Avg MRD	Model	Difference between misinformation ratio in feed and content pool
IR	Model	Percentage of agents in state I
EC	Model	Echo chamber intensity across the network
State	Agent	Infection state: S, E, or I
Followers	Agent	Number of followers (in-degree)
Following	Agent	Number of followed accounts (out-degree)
DS	Agent	Diversity score of recommendations for the agent
MC	Agent	Number of misinformation items in the agent's feed

Table 5.2: Community data per simulation step

Metric	Description
Communities	Mapping of agents to communities.
Sizes	Size of each community.
EC	Similarity of recent content within each community.
MR	Ratio of misinformation in recent content per community.

5.2 Experiment 1: RAs' Impact on Misinformation Propagation

The first experiment is related to RQ1 and RQ3, which are defined in Section 1.3, and explores how different RAs affect misinformation spread in a social network, including the echo chamber effect. In further detail, the experiments involve researching these subquestions:

1. How each RA affect misinformation propagation
2. How much misinformation is recommended by each algorithm, and whether the algorithm amplifies misinformation
3. To what extent does the RAs contribute to echo chambers in the social network

Certain assumptions are made for the values of fixed parameters. There exists no correct way to set up a social network, as they can be very different. The amount of bots and influencers in a social network also varies a lot, so the numbers are chosen based on general observations of social media platforms. The parameters chosen for Experiment 1 are presented in Table 5.3.

The number of users, average followers, initial news amount, and the percentage of bots and influencers are chosen to represent a scaled-down version of a realistic social network. The probability of misinformation is a baseline set to ensure that there is enough misinformation in the content pool to have an impact, but not enough to dominate the content propagation. At each timestep, 10 recommendations per user agent are generated to ensure sufficiently large content feeds, thus increasing the probability of user interaction. This enables earlier observation of changes in the network and helps reduce the total number of timesteps needed in the simulation.

Table 5.3: Parameters used in experiment 1

Parameter	**Value**
Timesteps	700
Number of users	200
Average followers	8
Initial news amount	400
Misinformation likelihood	10%
Bot percentage	7%
Influencer percentage	3%
Recommendations per step	10
Recommendation algorithm	{Rnd, CBF, UB-CF, IB-CF, Pop}
Diversity Level	0

5.3 Experiment 2: Increasing Diversity

The next experiment addresses RQ2 and RQ3 by testing whether increased diversity can affect the propagation of misinformation and the formation of echo chambers in the social network. To increase diversity in recommendations, a post-processing approach is used with MMR, as described in Section 4.3. The increased diversity method accounts for different levels of increased diversity for a more complex analysis. In more detail, this experiment investigates these subquestions:

1. Does increasing diversity limit the RA's impact on misinformation propagation
2. Does increasing diversity help break up echo chambers formations
3. Does different levels of increased diversity affect the results differently

To ensure a valid comparison with the results from Experiment 1, the same parameters are chosen except the diversity level. To analyze whether varying levels of increase in diversity affect the results differently, two distinct levels are explored: [0.75 and 1], where level 0 implies no increase in diversity, and level 1 implies a complete increase in diversity. Due to the nature of MMR, where the λ (lambda) parameter decides whether to prioritize user engagement or diversity, level 0.5 implies a balance between these competing priorities but does not indicate a significant increase in diversity. Therefore, the two diversity levels are chosen as higher than 0.5 to ensure an increase in diversity, but at different rates. The complete overview of the parameters chosen in this experiment is given in Table 5.4.

Table 5.4: Parameters used in experiment 2

Parameter	**Value**
Timesteps	700
Number of users	200
Average followers	8
Initial news amount	400
Misinformation likelihood	10%
Bot percentage	7%
Influencer percentage	3%
Recommendations per step	10
Recommendation algorithm	{Rnd, CBF, UB-CF, IB-CF, Pop}
Diversity Level	{0.75, 1}

Chapter 6

Results

This chapter presents the results of the experiments conducted. First, the results of Experiment 1 are presented, described in Section 5.2, followed by the results of the second experiment, described in Section 5.3. The results are presented using the evaluation metrics described in Section 4.4. For Experiment 2, λ is used to represent the diversity level.

6.1 Experiment 1: Misinformation Propagation

The results of Experiment 1 show how the different RAs affect misinformation spread and exposure, and how they perform over time during the simulation. Table 6.1 compares the overall average for the IR, MRD, MC, EC, and DS metrics for each RA, with each cell shaded from green (best) to red (worst), showing a classification from best to worst performance. Three figures are presented to illustrate how each RA performs over time regarding misinformation propagation and exposure, using the metrics IR (Figure 6.1), MRD (Figure 6.2), and MC (Figure 6.3). Figure 6.4 shows how each RA contributes to the echo chamber effect over time. In these figures, all 4 exhibit a pronounced transient during the first 20 to 50 steps before settling into a steady pattern. Therefore, the initial period is treated as a warm-up phase and is excluded from the average metrics calculations, as seen in Table 6.1.

6.1.1 Infection Rate

Looking closer at the metric IR in Table 6.1, CBF shows the lowest average Infection Rate (IR) of 0.276 and, therefore, is the RA which has the least impact on misinformation propagation. With a value of 0.505, almost twice as high as

Table 6.1: RA rankings by metric ($\lambda = 0$)

Rec Type	IR	MRD	MC	EC	DS
CBF	#1 (0.276)	#2 (-0.001)	#2 (11.237)	#5 (0.813)	#5 (0.076)
IB-CF	#2 (0.291)	#1 (-0.056)	#1 (8.350)	#3 (0.777)	#4 (0.223)
Pop	#5 (0.505)	#5 (0.077)	#5 (14.791)	#4 (0.795)	#1 (0.240)
Rnd	#4 (0.441)	#3 (0.001)	#3 (11.582)	#1 (0.766)	#2 (0.240)
UB-CF	#3 (0.402)	#4 (0.009)	#4 (11.745)	#2 (0.769)	#3 (0.235)

CBF, Pop acts as the algorithm with the worst impact on the IR. Pop is the only RA that performs worse than the baseline Rnd, which has a score of 0.441, but UB-CF comes close with a score of 0.402. IB-CF is similar to the first place, with an average IR of 0.291, which is only 0.015 higher, making it the second best RA in this metric.

Figure 6.1 shows the proportion of agents "infected" by misinformation in each simulation step. The figure exhibits significant fluctuations, which is mostly due to the implementation that after 40 timesteps, if they have not interacted with misinformation, the agent returns to the Susceptible state. The timeline figure does not show any significant new findings, as all RAs fluctuate around a level value. This makes it easy to observe the same performance ranking as presented in Table 6.1, with the exception of IB-CF and CBF, whose curves overlap more closely. The variation, indicated by the lighter shading surrounding the curve, is quite high in all algorithms.

6.1.2 Misinformation Ratio Difference

For the Misinformation Ratio Difference (MRD) metric in Table 6.1, IB-CF has the lowest score, with an average of -0.056, indicating a filtering effect against misinformation. The only other algorithm that results in an average negative value is CBF, but the filtering effect is close to insignificant with only a value of -0.001. The RA amplifying the most misinformation is Pop, again ending up with the lowest ranking with an average value of 0.077. UB-CF also shows a slight amplification with a value of 0.009, and Rnd shows almost no impact with a low value of 0.001, which is expected from the baseline algorithm.

Figure 6.2 shows the MRD values of the RAs on average during each simulation step. Pop frequently shows positive MRD spikes, revealing that such algorithms tend to over-recommend misinformation during these spikes. Initially, a large

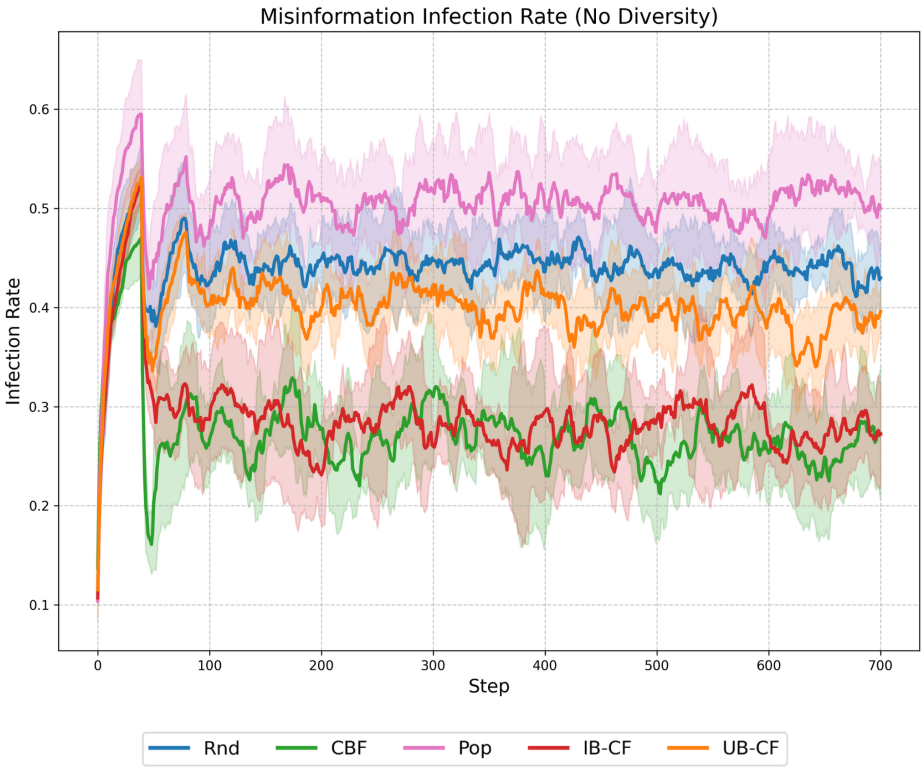

Figure 6.1: IR over time

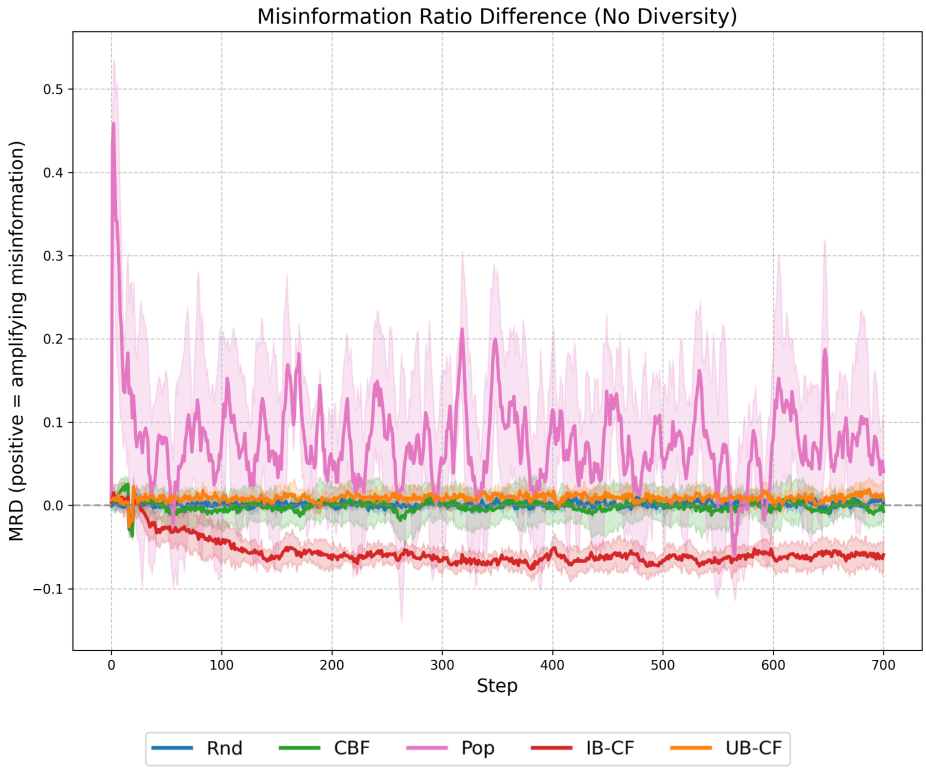

Figure 6.2: MRD over time

spike is observed by Pop, which over time stabilizes into lower spikes. Rnd and UB-CF fluctuate around zero, with UB-CF often being slightly positive. A small decreasing slope can be observed in the IB-CF algorithm before it stabilizes, which is an interesting observation. Another notable finding is how the variations of the algorithms differentiate. Pop shows high variation, while others are more stable. Although the average curves of CBF, UB-CF, and Rnd overlap largely, CBF shows much higher variance than UB-CF and Rnd. IB-CF has some variation but appears relatively stable.

6.1.3 Misinformation Count

Misinformation Count (MC) is often closely aligned with MRD, and as observed in Table 6.1, the order of the ranking of the algorithms in MC is the same as in MRD. IB-CF also ranks highest in MC, with an average score of 8.35. CBF has the second-lowest MC of 11.237, but Rnd and UB-CF are not far behind with scores of 11.582 and 11.745. Pop has the worst score and ends up with a significantly higher average MC of 14.791.

Figure 6.3 shows the amount of misinformation in the recommended content in each simulation step. Similarly to the pattern observed in Figure 6.2, which displays MRD over time, Pop also presents significant spikes of high values in MC. In the timeline, it is observed that UB-CF, Rnd, and Pop overlap, which aligns with their similar average values. IB-CF can be distinguished as the best performer, and a distinctive trend also appears in this figure, where the IB-CF curve has a slight downward slope. In terms of variance, Pop stands out with the highest levels, whereas the other RAs have relatively smaller, similar variance.

6.1.4 Echo Chamber Effect

While the different metrics have shown similar ranking positions so far, a closer look at the Echo Chamber Effect (EC) metric in Table 6.1, shows a different ordering. CBF, which previously only exhibited good rankings, has the highest average EC score of 0.813. Pop maintains its poor performance by ranking second worst, with a value of 0.795. IB-CF returns an average EC score of 0.777, the middle ground between the best and worst performers. Rnd ranks highest with a value of 0.766, but UB-CF is a close second place with 0.769, only 0.003 higher than the first place.

Figure 6.4 shows the EC values over time for each RA. These curves seem less continuous than the other figures because EC is calculated only every fifth timestep. CBF is consistent in providing the highest EC values throughout the simulation, having stronger community homogeneity in the recommendations. An observation from the figure is that CBF and Pop have a significantly higher variance than the other RAs. Both of them also show more unstable behavior

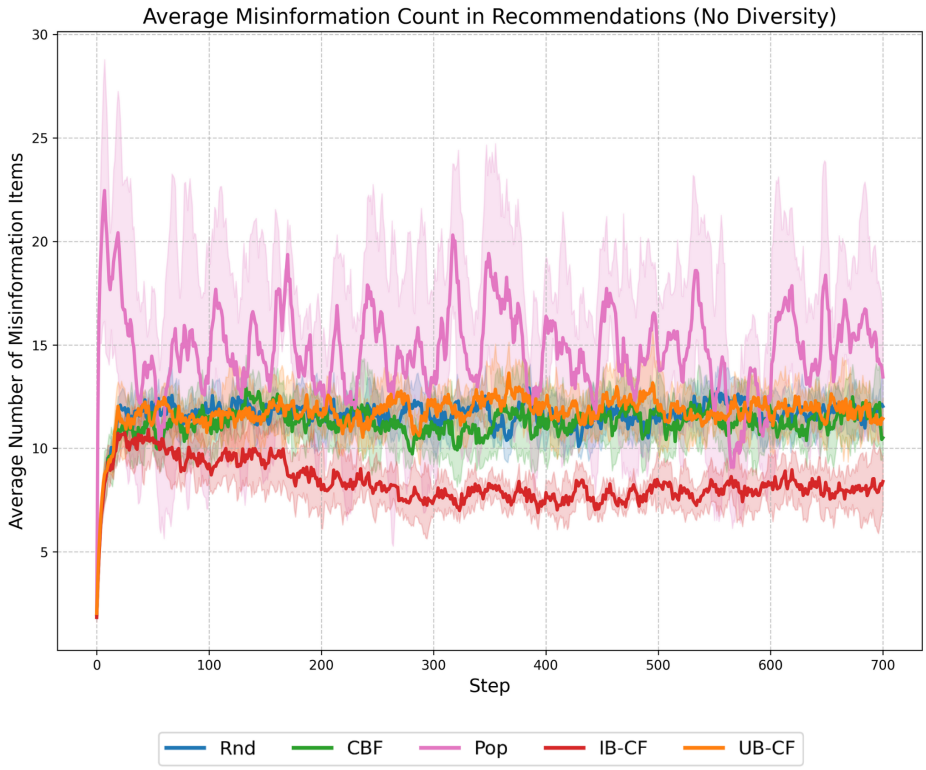

Figure 6.3: MC over time

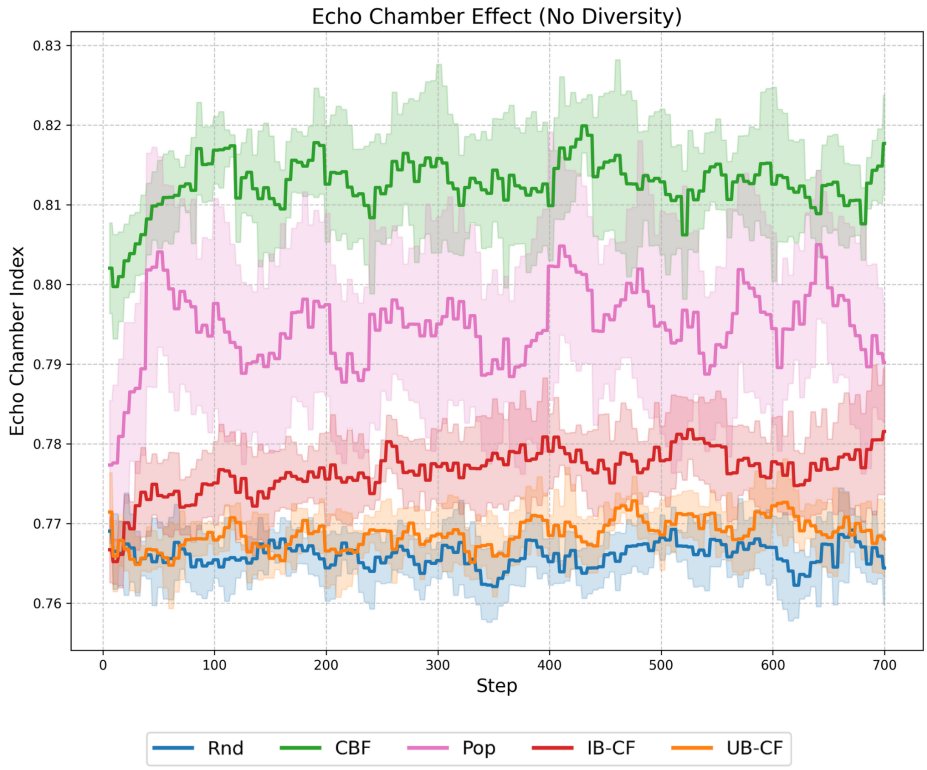

Figure 6.4: EC over time

recognized by several peaks of high values, while IB-CF, UB-CF and Rnd show more stable and linear patterns. Another notable insight is how the IB-CF curve has a slightly rising curve, where it seems to get higher EC values over time.

6.2 Experiment 2: Increasing Diversity

The results of Experiment 2 show how increased diversity affects misinformation spread and echo chamber formation in a social network. Table 6.2 presents the performance of the RAs on the evaluation metrics at $\lambda = 1$, including the Diversity Increase (DI) metric mentioned in Section 4.4. Table 6.3 shows 4 sub-tables of the IR, MRD, MC, and EC metrics, with each RA as a row and the different levels of diversity (λ) as columns to make it easy to compare how the diversity levels affect the metrics.

Table 6.2: RA rankings by metric ($\lambda = 1$)

Rec Type	IR	MRD	MC	EC	DS	DI
CBF	#1 (0.265)	#3 (0.005)	#4 (11.772)	#5 (0.803)	#5 (0.127)	#1 (67.502)
IB-CF	#2 (0.284)	#1 (-0.029)	#1 (9.718)	#2 (0.762)	#2 (0.268)	#2 (55.854)
Pop	#5 (0.503)	#5 (0.060)	#5 (14.432)	#4 (0.766)	#1 (0.316)	#4 (36.785)
Rnd	#4 (0.424)	#2 (0.002)	#3 (11.410)	#3 (0.764)	#4 (0.241)	#5 (0.000)
UB-CF	#3 (0.410)	#4 (0.005)	#2 (11.405)	#1 (0.761)	#3 (0.250)	#3 (42.008)

Pop shows the highest absolute diversity score with or without an increase in diversity, as evidenced by comparing the results of Table 6.1 and Table 6.2. However, its diversity increase is the lowest among the algorithms where diversity is increased (Rnd does not increase diversity), meaning that the proportional improvement is modest even if the baseline is high. In contrast to this finding, the CBF shows the lowest diversity score but the highest increase in diversity.

6.2.1 Infection Rate

After increasing diversity, Pop remains the worst algorithm in relation to IR, as shown in Table 6.2. When comparing IR before and after increasing diversity in Table 6.1 and Table 6.2, the ranking of all algorithms remains the same; Pop still induces the highest IR and CBF the lowest.

Comparing the effect of different values of λ in Table 6.3 (a), each algorithm shows a decrease in IR from $\lambda = 0$ to $\lambda = 1$, excluding UB-CF which has a slight increase. The best performer, CBF, changes from a score of 0.276 at $\lambda = 0$ to

Table 6.3: Impact of diversity level on various metrics

(a) IR

RA	0.0	0.75	1.0
CBF	0.276	0.289	0.265
IB-CF	0.291	0.298	0.284
Pop	0.505	0.513	0.503
UB-CF	0.402	0.376	0.410

(b) MRD

RA	0.0	0.75	1.0
CBF	-0.001	-0.011	0.005
IB-CF	-0.056	-0.016	-0.029
Pop	0.077	0.070	0.060
UB-CF	0.009	0.006	0.005

(c) MC

RA	0.0	0.75	1.0
CBF	11.237	10.995	11.772
IB-CF	8.350	10.232	9.718
Pop	14.791	15.172	14.432
UB-CF	11.745	11.557	11.405

(d) EC

RA	0.0	0.75	1.0
CBF	0.813	0.801	0.803
IB-CF	0.777	0.765	0.762
Pop	0.795	0.773	0.766
UB-CF	0.769	0.764	0.761

0.265 at $\lambda = 1$, which is not a significant decrease. This level of small differences can be observed across all RAs, showing that the impact of different values λ on IR is minor.

An interesting observation is how CBF, IB-CF and Pop all get increased IR values at $\lambda = 0.75$, which then are all decreased below the original value, at $\lambda = 1$. In contrast, UB-CF has the opposite effect, where the value decreases at $\lambda = 0.75$, but then increases to a higher level than the original.

The IR values of the RAs are illustrated over time in Figure 6.5. A noticeable difference can be observed on the UB-CF curve, where IR levels decrease when diversity increases to $\lambda = 0.75$, but rise again at $\lambda = 1$. Another feature of the UB-CF curve is that it has a slight downward slope at all levels of λ, where the other algorithms fluctuate around certain levels. Another finding is how the Pop algorithm's variation is lower when diversity is increased. The same seems to happen for IB-CF and CBF, although it is more difficult to observe. Analyzing the curves of IB-CF and CBF at $\lambda = 1$, they exhibit a notably similar behavior, particularly from timestep 300 onward.

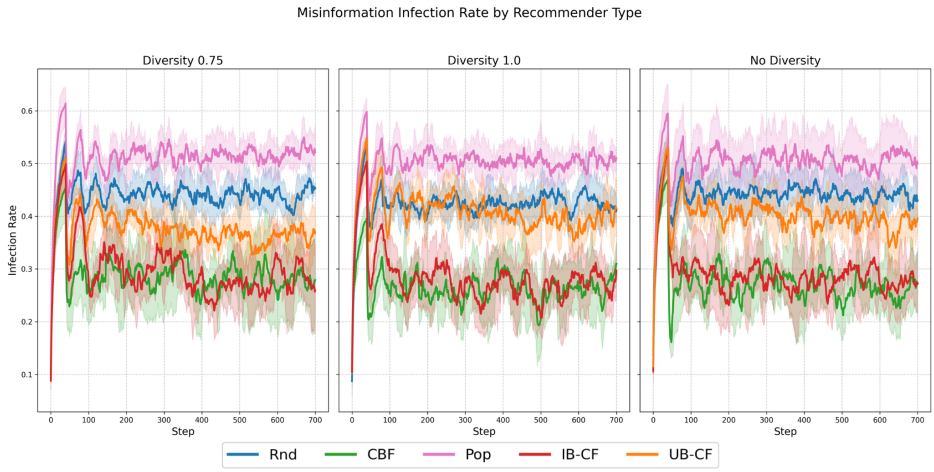

Figure 6.5: IR comparison

6.2.2 Misinformation Ratio Difference

Regarding the MRD metric, the various levels of diversity affected the performance of the RAs differently. Table 6.3 (b), shows that the CBF changes from counteracting to slightly amplifying misinformation as diversity increases, changing from an average MRD of -0.001 to 0.005. IB-CF continues to be the algorithm with the highest ranking but experiences a slight increase in the MRD value, losing some of its ability to filter out misinformation. Interestingly, IB-CF has the highest value at $\lambda = 0.75$, which is further slightly decreased at $\lambda = 1$. UB-CF shows an improvement in MRD, indicating that it becomes better at filtering misinformation with increased diversity. Pop is the overall weakest performer, being the most prone to amplifying misinformation, but shows a slight reduction from 0.077 to 0.06 in MRD with increased diversity.

Figure 6.6 shows various trends across the algorithms with respect to the MRD metric. An interesting observation is that IB-CF, which shows a slight downward slope without added diversity, is pushed toward 0 at $\lambda = 0.75$, however, at $\lambda = 1$ a slight slope reappears. Another finding is how the variation of IB-CF increases as the diversity increases. Figure 6.6 also shows how Pop's spikes of high MRD levels are toned down with higher diversity levels, aligning with the overall average decrease in MRD.

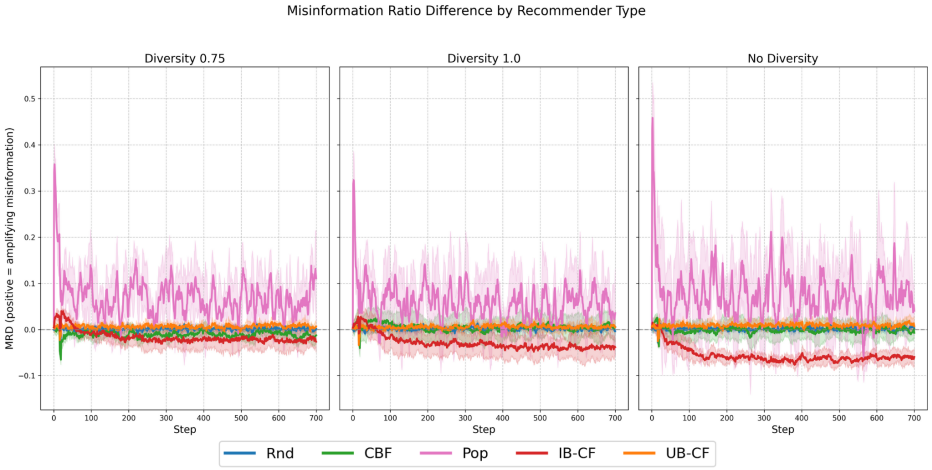

Figure 6.6: MRD comparison

6.2.3 Misinformation Count

Table 6.2 shows that Pop remains the worst performer with respect to the MC metric, reflecting that, even with increased diversity, it still recommends the highest number of misinformation items. Table 6.3 (c), shows that at $\lambda = 0.75$ the MC actually increases, but at $\lambda = 1$ the value decreases beyond the original value at $\lambda = 0$. The same phenomenon can be seen by IB-CF, where the MC value changes from 8.35 at $\lambda = 0$, to 10.232 at $\lambda = 0.75$, and then down to 9.718 at $\lambda = 1$. A contrasting pattern is observed for CBF; MC first decreases at $\lambda = 0.75$, before it increases at $\lambda = 1$, exceeding the value observed without an increase in diversity. UB-CF's MC values have no dramatic changes but show a slight decreasing trend as λ increases.

Figure 6.7 shows the variation of the MC values over time across different diversity levels. Many of the same trends observed in Figure 6.6 are also present for MRD. The downward slope of IB-CF at $\lambda = 0$ flattens to a linear trend at $\lambda = 0.75$, before again slightly decreasing at $\lambda = 1$. The variation of IB-CF can again be observed to increase when λ increases. Pop shows slightly lower peaks as λ increases. The other algorithms are mainly centered around zero, showing no significant changes in the timeline as λ changes.

6.2.4 Echo Chamber Effect

For the EC metric, the CBF maintains the highest score after diversity increase, as observed in Table 6.2. UB-CF, which ranks a close second after the baseline Rnd

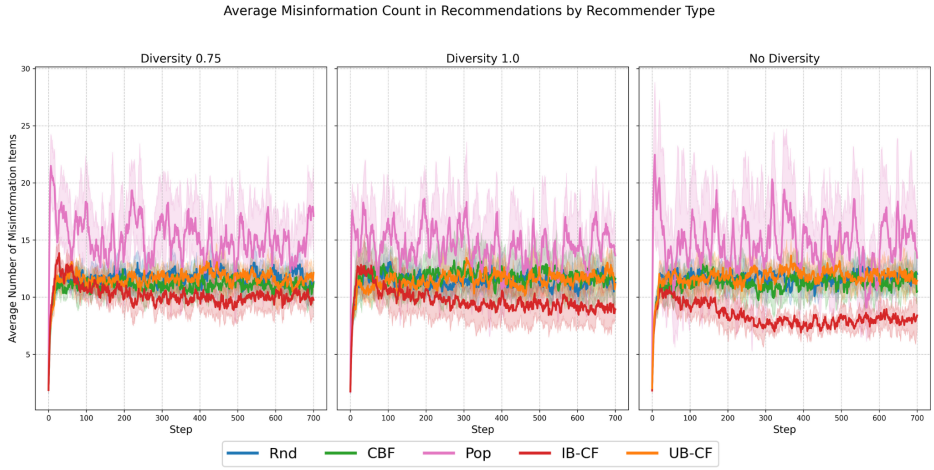

Figure 6.7: MC comparison

before increasing diversity, progresses to the first place when $\lambda = 1$. Comparing the average results of each RA at different levels of λ in Table 6.3, it is shown that all RAs have decreasing EC values as λ increases. Although most RAs show low differences across diversity levels, Pop shows the largest decrease, changing from 0.795 at $\lambda = 0$ to 0.766 at $\lambda = 1$.

Figure 6.8 also shows that EC decreases for all RAs as diversity increases. However, the magnitude and effectiveness of this reduction varies between RAs, again showing the differences in how they handle information diversity. Pop shows the largest absolute reduction in the EC score, although its average EC value remains relatively high even at $\lambda = 1$. Another interesting finding from Figure 6.8 is that IB-CF shows an upward trend when $\lambda = 0$, but as diversity increases, the curve stabilizes around a lower value. The same phenomenon can also be observed, although not as prominent, in UB-CF.

Figure 6.9 complements these findings by providing the mean EC values by community for each RA at all levels of diversity. CBF displays the highest EC through all communities and diversity levels. UB-CF, IB-CF and Rnd maintain relatively low EC values across all communities, with UB-CF being the lowest in most cases. Pop has higher EC than UB-CF and Rnd, but there is clearly a decreasing trend in most communities as diversity increases. When λ increases from 0 to 1, EC values in general converge between communities, especially for IB-CF, UB-CF and Rnd.

By analyzing theEC value within each community in Figure 6.9, Rnd and UB-CF exhibit minimal variation in the local EC in all communities. However,

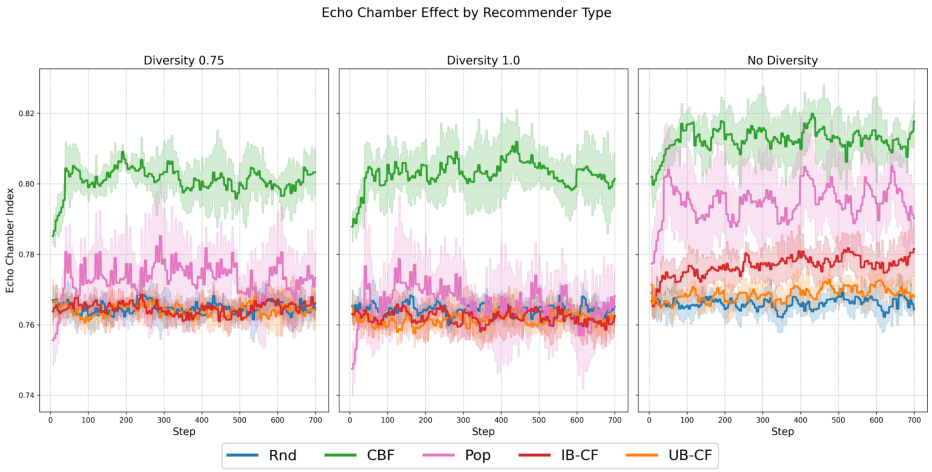

Figure 6.8: EC comparison

Pop, CBF, and IB-CF all show community-dependent swings with higher levels of variation. This variance is largely maintained across all diversity levels, but the values are overall lower. Figure 6.10 visualizes the Misinformation Ratio (MR) for each RA in communities and diversity levels. MR increases with diversity for all algorithms except Rnd, where values remain stable. The largest increases in MR are observed for Pop and CBF. IB-CF and UB-CF also show noticeable increases in communities as diversity increases.

When comparing the EC in each community in Figure 6.9, to their MR in Figure 6.10, several interesting observations can be made. First, at $\lambda = 0$ in community 6, the MR for CBF is quite small, even though the EC value appears to be the highest in all RAs and diversity levels. Among the communities, it is evident that in community 1 at $\lambda = 0$, although CBF has one of the lowest EC values, it produces the highest MR value in all communities.

Another interesting finding is that, although increasing λ reduces EC primarily in all communities, as shown in Figure 6.9, it appears to increase the level of MR in the same communities. This suggests that the content that circulates in the communities is more diverse but also includes more misinformation.

Figure 6.9: Mean EC value across communities and RAs

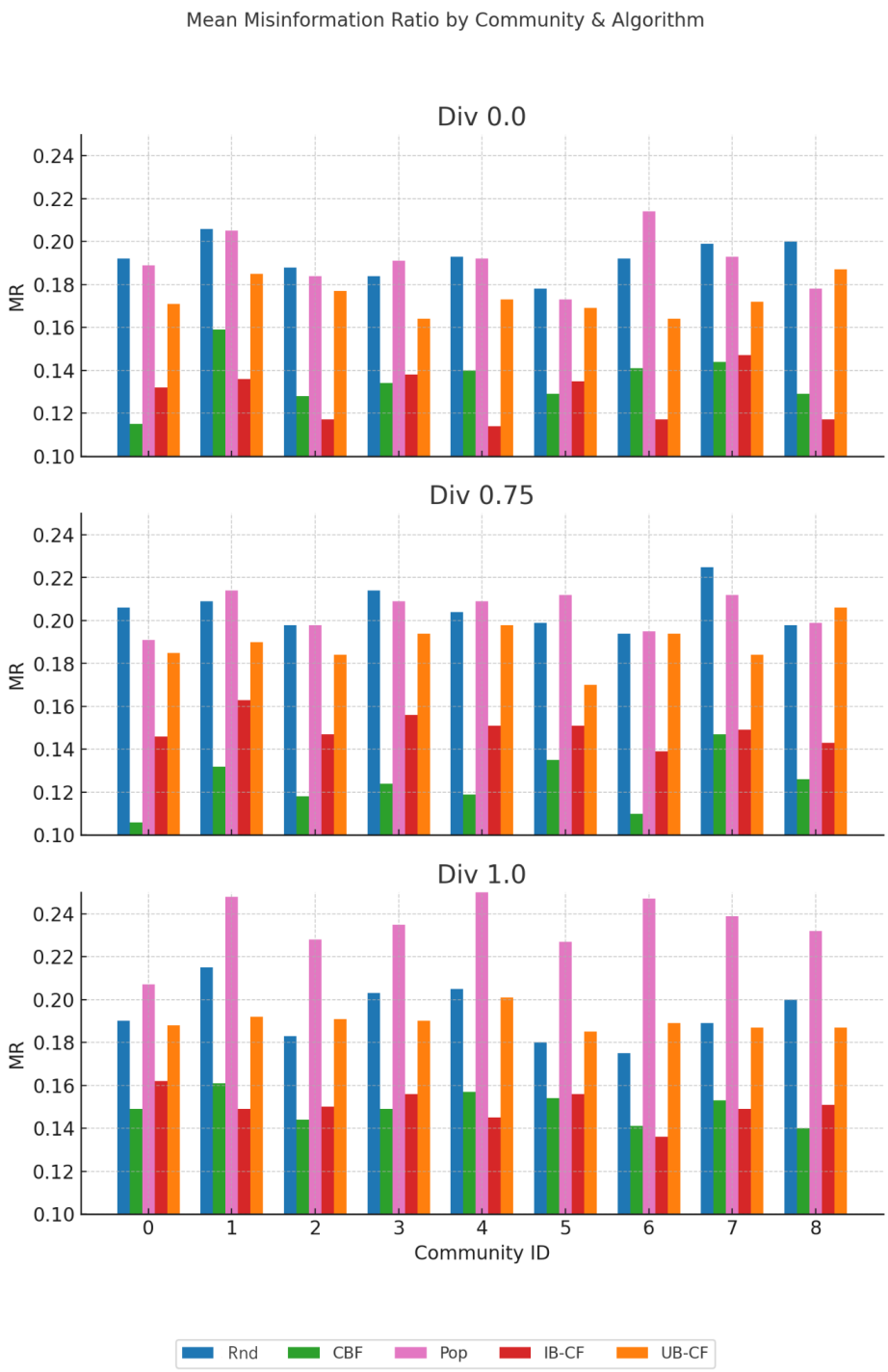

Figure 6.10: Mean MR across communities and RAs

Chapter 7

Discussion

This chapter discusses the experimental findings in relation to the research questions outlined in Section 1.3. RQ1, RQ2, and RQ3 are discussed under the respective Sections 7.1, 7.2 and 7.3. Section 7.4, includes a general discussion of the experiments, while Section 7.5, outlines the limitations of this study, including an analysis of its impact on sustainability.

7.1 RQ1: RAs' Effect on Misinformation Spread

How do different RAs affect the spread of misinformation?

Examining RAs' influence on misinformation propagation using ABM provides some interesting results, including Pop being the algorithm which leads to the highest amplification and exposure of misinformation spread. By incorporating feedback loops, this model mirrors real-world conditions, where user interactions with content influence subsequent recommendations. This dynamic aspect enhances the realism of the simulation, allowing the observation of how algorithmic biases may evolve over time. Unlike the primarily static evaluation frameworks used in the studies by Pathak et al. and Fernandez et al. [4, 8], the inclusion of feedback loops allows this model to have self-reinforcing behavior patterns, such as the dominance of misinformation in a user's feed.

In line with previous findings [4,8], which demonstrated that the Pop algorithm propagates misinformation more aggressively, the results of this study further substantiate this statement. The results in Table 6.1 indicate that Pop is ranked as the worst RA in terms of IR, MRD, and MC, showing a significantly poor impact on misinformation propagation in the social network. A closer examination of the IR metric in Figure 6.1 shows that Pop is the only algorithm with a higher IR than

the baseline Rnd. Consequently, this means that IB-CF, UB-CF and CBF all lead to less interaction with misinformation than the baseline. However, this does not imply that IB-CF, UB-CF and CBF are "good" choices with respect to the limitation of spreading misinformation in real life, as the compared baseline Rnd is rarely implemented RA in social media platforms. A more in-depth analysis on the performance of the other RAs will be discussed later in this section. As the findings align with previous research, it can be concluded that in a time-sensitive analysis, the Pop algorithm still has the worst and highest impact on spreading misinformation on a social network.

Pop's behavior is consistent with the nature of the algorithm, which recommends content that receives high levels of engagement to most users. If misinformation content gains popularity, the Pop algorithm is likely to further amplify its reach, therefore causing large spikes in both MRD and MC, as seen in Figure 6.2 and 6.3. Misinformation is also more engaging than regular news content [23], further enhancing its easy access to virality.

A notable insight from Figure 6.2 and 6.3, is how Pop has a significant peak in the early stages of the simulation. This peak can be explained by the cold start problem, where, in the early stages, the user-item matrix has very few entries, and initial interactions are most likely provided by the most naive user agents. Consequently, the limited number of items that are interacted with are likely to be recommended to almost everyone, as they are the only ones deemed "popular". Since misinformation typically has a higher engagement factor in the early stages, it is highly likely that the first items interacted with on the network are misinformation, causing the huge MC and MRD spike.

Pop also exhibits the largest variance among the different algorithms in the timeline results. This is likely due to how it prioritizes popular news items; the timing of when misinformation gains high engagement is rather random and unpredictable.

The results diverge from previous work in one important aspect. In Pathak et al.'s study [4], CF methods, both UB-CF and IB-CF, tend to recommend a higher volume of misinformation. The findings of this study contradict this statement, as IB-CF demonstrates resistance to the propagation of misinformation, in addition to consistently outperforming other RAs in minimizing both exposure and infection. Rather than being susceptible to high engagement, the IB-CF model effectively limits misinformation through its focus on item similarity, suggesting that it may help filter out low-quality or misleading content clusters. These differences could either be attributed to how the simulation includes time dynamics and feedback loops or to the differences in implementation of the algorithms. However, IB-CF and UB-CF in this implementation are algorithms from the LensKit library, indicating that they are based on standard functionality. Therefore, the differences in the results are more likely due to the experiments using ABM simulations and,

consequently, to the inclusion of more complexity than the approach in [4]. In addition, all data in this project are synthetically generated, compared to [4] which uses labeled datasets and information diffusion models.

The downward slope of IB-CF in Figure 6.2 and Figure 6.3 may be explained by how, over time, more interactions are added to the user-item matrix, and the IB-CF algorithm is trained on more reliable data, thus becoming better at distinguishing the content users prefer. This creates a positive feedback loop that can gradually shift recommendations away from misinformation.

UB-CF is not the worst performer among RAs, but it is also not among the best. In the rankings of the different metrics, UB-CF is usually observed somewhere in the middle, and its results over time show that it maintains rather stable profiles. UB-CF also has quite similar behavior to the baseline Rnd, showing that social proximity is not strongly worse than without personalization or engagement-driven approaches.

In this thesis, CBF yields lower misinformation metrics, probably due to its reliance on semantic matching rather than user behavior patterns or viral trends. These findings align with those of Pathak et al. [4], who also found CBF to be effective in limiting misinformation. The nature behind CBF is to recommend content based on how similar the topic of the content is to the user's preference, and the only time this could amplify misinformation spread is when the user consistently prefers content that has a high probability of being misinformation. For example, during the 2016 presidential election in the USA, a user who consistently engaged with content that is positive toward Donald Trump would have been more likely to be exposed to misinformation about Hillary Clinton through algorithms like CBF. The same principle applies to other item-based methods, such as IB-CF. However, this can create an amplification for a subgroup of users. At the same time, a personalized relevance approach will protect most users from a severe spread of misinformation.

Running experiments that include time dynamics can be shown to align well with other research performed on static datasets when observing how RAs propagate misinformation. The most significant difference in this simulation study remains how the IB-CF method helps restrict the spread of misinformation.

7.2 RQ2: Diversity's Effect on Misinformation Spread

> *How does increasing diversity in recommendations affect the spread of misinformation?*

The results in Table 6.2 show that CBF and IB-CF increase their MC and MRD values while remaining relatively low compared to Pop and UB-CF. Such results

imply that content semantics might contribute to filtering against misinformation, because their core logic relies on content similarity. Increasing diversity could actually reduces this filtering effect.

Looking at Table 6.3, it is interesting how for both CBF and IB-CF, although MC and MRD increase from $\lambda = 0$ to $\lambda = 1$, IR actually decreases. The slight drop in IR may indicate that users encounter a wider variety of content, making it less likely to cause infection even if there is more misinformation present. The new content may also be less engaging in general, reducing the chance of infection. The study by Lunardi et al. [3], also found that IB-CF with MMR diversification significantly decreased interactions with misinformation, which the results of this study can further substantiate.

When $\lambda = 1$, UB-CF recommends slightly less misinformation on average, as reflected in its MRD and MC scores in Table 6.3. However, the difference is minor and the IR value actually shows a slight increase. Therefore, it is hard to conclude that increasing diversity has a significant effect on how UB-CF propagates misinformation. This could be because the UB-CF recommendations depend on social context and other users, and not item similarity. Because of that, UB-CF's recommendations are diverse by default, and increasing diversity might consequently not alter the results much. Given its previously observed similar behavior as Rnd, increasing diversity has limited leverage to disrupt misinformation bubbles.

Even with increasing diversity, Pop continues to have the highest MRD, IR, and MC, as seen in Table 6.1. This may be because, although the recommendations are more diverse, they are not curated for veracity and might still include misinformation. The nature of the Pop algorithm remains the same; misinformation that is among the "top" can reach a large number of users in the social network in a short amount of time. The persistent low ranking shows that Pop is fundamentally vulnerable to misinformation and that diversity alone cannot counteract these issues.

An interesting finding in Table 6.3, is that the changes in the metrics' values is not linear as diversity increases. Sometimes, as for CBF, IB-CF, and Pop regarding IR, the values increase at $\lambda = 0.75$ before decreasing even lower than the original value at level $\lambda = 1$. This is most likely due to the nature of the MMR method. At $\lambda = 0$, the RA selects items solely on the basis of relevance. At $\lambda = 0.75$ diversity influences the recommendation process but only accounts for 25% of the decision. The first selection typically remains the most relevant item chosen by the RA, while subsequent selections will only face a mild penalty for being similar. Thus, although some variety is injected, the general habit of the RA is not broken. In addition, since the recommendations remain based on relevance, the content that is presented to the user may be somewhat more diverse, yet still highly relevant to the user (particularly in the case of CBF and IB-CF). This

increases the probability that the user actually interacts with the content and becomes infected. In general, the fluctuations of opposite performances between the diversity levels can be attributed to the inner workings of MMR.

Another highlight of the results is how the increase in diversity affects the variance of the different RAs. Across Figure 6.5, 6.6 and 6.7, Pop's variance is reduced when diversity increases. This might be explained by how when diversity increases, the personalization weight of the Pop algorithm decreases, and following the individual variance decreases. Diversity might homogenize the final selection for users, recommending the same popular content to everyone. This leads users to converge towards similar levels, which in turn results in lower variance overall. Conversely, IB-CF shows an increase in variance in MRD and MC as the diversity increases, which can also be explained by the algorithm's nature. IB-CF relies on user interactions for their recommendations, and the users' interaction patterns vary significantly. When $\lambda = 0$ the algorithm recommends items based on the interactions of the users, which are largely based on their preferences. As a result, IB-CF will recommend quite similar items over time, causing the average count of misinformation to follow the same probability distribution, leaving minimal variance of MRD and MC. However, when λ increases, the algorithm could disrupt previous homogeneous recommendations of IB-CF, and the probability distribution changes to be more unpredictable. When the diversity levels increase, Pop's variance decreases and IB-CF's variance increases, showing the same phenomenon but from an opposite direction.

As λ increases from 0 to 1, the higher diversity and reduced misinformation recommended by the RA increase the MR in almost all communities, as illustrated in Figure 6.10. This may be because diversity exposes users to a wider range of content, which can include misinformation that would normally not reach them in a more homogeneous environment. Examining Pop specifically, it raises the MR for all communities when diversity is increased, while its MRD and MC values both slightly decrease. This observation shows a complex aspect of the simulation. Without increasing diversity, Pop tends to recommend the same viral misinformation content over and over again, and these few posts gain a huge reach. This leads to elevated MC MRD, and MR in the communities. However, when diversity is increased, MR further increases, as certain "safe" and true popular items are replaced with more random selections (which have a high likelihood of being misinformation). MRD and MC simultaneously go down because misinformation items are less likely to go super viral anymore.

Why increasing diversity causes minimal effects

One reason why increasing diversity does not appear to have a huge impact on misinformation propagation is that increasing diversity can still introduce more misinformation. MMR does not inject carefully curated diversity, and some of

those more diverse items might be misinformation. Another possible explanation is based on the implementation of how some news items are more likely to be misinformation than others. If misinformation exists across all topic areas, then diversifying the recommended topics will not lead to less misinformation exposure. The implementation used in this study might not be representative enough to provide the necessary dynamics that increasing diversity could disrupt. Incorporating a more complex distribution of topics and their varying possibility of being misinformation could have resulted in a more effective outcome in response to diversity increase.

One reason the diversity increase had limited effect could be the way we implemented MMR: before re-ranking, only a triple-sized candidate list was used, which may be too modest, potentially limiting the pool of candidate items to perform a proper re-ranking, further affecting the impact of the diversity increase.

Another aspect is that RAs, which reduces direct exposure to misinformation, may still be insufficient in a network where users themselves amplify the spread of misinformation through their interactions. In reality, users may be more likely to share misinformation more widely when exposed to new, engaging content, or they may disregard various recommendations and instead favor content shared by those they follow. The results indicate that even when the MR in the different communities increases, the IR decreases or stays stable, specifically for CBF, IB-CF, and Pop. Thus, exposure to misinformation does not equal infection. These findings highlight that algorithmic improvements alone are not sufficient and that user behavior and network structure also have a great impact on RAs' response to increased diversity. Moreover, the increased diversity may further weaken the persuasive power of misinformation, as encountering misinformation in a diverse content feed will have less impact on the user, although this dynamic is not accounted for in this model.

When diversity is increased to $\lambda = 1$, algorithms that initially have a positive MRD tend to decrease their values, and algorithms that have a negative MRD increase their values. This may be explained by the nature of the MRD metric that measures the deviation from the global baseline. Higher diversity makes individual users' recommendations more representative of the global content pool, pushing MRD toward zero regardless of whether this shift indicates improvement or decline in performance.

Analyzing the impact of increased diversity on the propagation of misinformation in different RAs reveals that it does not necessarily lead to a significant reduction in the spread of misinformation. The paper by Pathak and Spezzano [10] claims that a lower diversity in recommendations contributes to a stronger echo chamber effect, which propagates misinformation at a higher rate than it would have with higher levels of diversity. The paper also states that CBF and CF methods exhibit higher diversity than Pop. However, the results presented in

this study contradict that finding. The discrepancies could occur due to the way the DS is calculated or due to the fundamentals of the implementation. CBF in academic studies is often driven by genre and tag matching, while this implementation is purely similarity-driven. The same applies to Pop, where, in this simulation, exploration factors and minor user-preference factors are introduced to prevent the algorithm from recommending identical content to all users. This approach further amplifies their popularity and edges out all other content that tries to become viral. As a result, the implementation is essentially a hybrid recommender, which may explain the differences from the study by Pathak and Spezzano [10]. Although their study shows that high diversity levels insinuated lower misinformation spread, the differences in results between their study and this study imply that this statement depends on factors such as RA implementation and metrics.

The experimental results in this study show that not all algorithms benefit equally from increased diversity and that there is a trade-off between diversity and relevance. In practice, diversity needs to be balanced with other aspects, such as content quality, as diversity itself can be a limitation when used as a standalone solution.

7.3 RQ3: Diversity and Echo Chamber Effect

> *How does the echo chamber effect change in relation to RAs and increased diversity?*

Examining the impact echo chambers have on RAs' contribution to the spread of misinformation relative to diversity increase shows interesting results. In line with Tommasel and Menczer [9], the results indicate that increasing diversity can break up echo chambers, an outcome that is often beneficial, especially in communities where large amounts of misinformation are often spread freely. However, increased diversity also leads to some unintended consequences, as mentioned in the previous section.

With an increase in diversity, all RAs show a reduction in EC in Table 6.3, though the magnitude varies depending on the algorithm. CBF shows the highest EC, which can possibly be attributed to the way the social network is implemented by clustering users with similar preferences together. It has only a marginal reduction, indicating that the nature of CBF itself reinforces user-preference loops, which an increase in diversity struggles to limit. IB-CF also uses item similarity but ranks much better than CBF in the EC metric. One explanation is that IB-CF relies on user interaction patterns, not solely on the content itself. This allows it to recommend items that have been co-engaged by users with different preferences, especially when diversity is increased. As a

result, users are exposed to a wider range of content, which helps to break up echo chambers more effectively than CBF. Similarly, Lunardi et al. [3] state that IB-CF is the most effective in decreasing the effects of filter bubbles and reducing the interaction with misinformation when MMR is used for the diversification of the algorithm. This aligns with the findings of this study, in which IB-CF shows a decrease in both EC and IR when $\lambda = 1$.

UB-CF ends up with the least relative improvement for EC as diversity increases, but it also ranks as the best algorithm in Table 6.2. This is likely due to UB-CF already recommending diverse content to users, based on the algorithm's fundamental nature. Pop shows the greatest reduction in EC when diversity is increased, indicating that Pop recommends more variations of popular content to users, thus reducing content similarity within communities.

In general, the increase in diversity helps break up echo chambers, but whether it has resulted in improved or decreased performance has differed between the RAs. For example, it is presented that reducing the echo chambers in CBF and IB-CF leads to an increase in MRD and MC while simultaneously leading to a reduction in IR. The increased diversity therefore inflicts more misinformation in recommendations, but at the same time the echo chambers and misinformation propagation are reduced, which seems paradoxical. The apparent contradiction reflects the complex nature of the spread of misinformation. Although increased diversity leads to higher exposure to misinformation in metrics such as MRD and MC, it simultaneously breaks down echo chambers that facilitate viral propagation, as indicated by decreased IR. This observation highlights the distinction between exposure and propagation and suggests that the breaking of the echo chambers may be interconnected with mitigating the spread of misinformation, also supported by [3]. Conversely, breaking echo chambers may be less effective in reducing misinformation exposure. For example, Pop and UB-CF exhibit decreased MRD and MC values when echo chambers decrease. However, their IR remains stable or, in the case of the UB-CF, increases slightly, further substantiating the distinctiveness between exposure and propagation effects.

Based on the results, it can be seen that increased diversity in recommendations leads to reduced EC. However, the degree to which this aligns with improved performance in terms of the spread of misinformation and exposure varies between RAs.

7.4 General Discussion

The relationship between diversity, echo chambers, and misinformation can be complex and sometimes counterintuitive. Breaking echo chambers is viewed as a strategy to mitigate misinformation. By increasing diversity, echo chambers are disrupted, although the results show that the spread of misinformation does

not decrease in all cases. For some algorithms, they end up recommending less misinformation, while the misinformation circulating in the communities is still increased. This shows that there is a complex interplay between algorithmic performance and user interaction, and the increase in diversity must be carefully evaluated to prevent unintended consequences.

Evaluating a single metric can be misleading and does not capture the whole picture. For example, even if the RAs recommend less misinformation on average, there may still be more circulating in individual communities. The choice of metrics are carefully evaluated to, as much as possible, present the results in a way that captures the whole picture. For example, EC will not always lead to a lower IR, and a lower MRD does not always translate to a lower MR in communities. A comprehensive view is necessary to truly understand the dynamics of misinformation.

The simulation and results show that users are not passive recipients, but their choices and sharing behavior shape outcomes as much as algorithms do. If the RA recommends less misinformation, users can still engage with and amplify the spread and further affect the performance of the RA. This shows that user behavior is important to evaluate in addition to the algorithmic outcome.

Platform designers should consider the trade-offs between diversity, relevance, and safety in recommendation systems. It is not enough to only increase diversity or only break echo chambers. There must be a holistic view with an understanding of user behavior and with control of the quality of the content. It is also important to account for the feedback loops between the algorithm and user behavior to achieve effective mitigation of misinformation and echo chambers.

7.5 Limitations

Although the simulation captures key findings on the spread of misinformation on social media, there are also certain limitations to the approach. First, agents in the simulation follow fixed rules with predefined personality types. This simplification removes the behavioral diversity and adaptive learning that are observed in real social media users. This limits the model's ability to reflect nuanced opinion dynamics. In addition, user agents are characterized only by user preferences, without taking into account demographic or sociocultural factors such as age, sex, marital status, or occupation. By omitting these user attributes, the model may miss important connections between demographics and online behavior.

Users interact with content based on similarity with their preference, engagement, and naivety level. Content, in turn, is characterized solely by a topic vector, an engagement level, and a binary indicator of whether it is misinformation. It does not encapsulate semantics such as whether content supports or opposes a given topic, which are important factors when dealing with domains such as

political news. In reality, this process is more complex, and the implemented model's abstractions may fail to react to phenomena like confirmation bias.

All user preferences, topic vectors, and network structures are generated synthetically and remain static throughout the simulation. As a result, the model does not capture the temporal evolution in user interests or the emergence of topic trends, such as breaking news. Content creation is tightly linked with user preferences, ignoring the variability and creativity of real human-generated content, as not every human posts content that aligns completely with their interests. Engagement with content is based solely on topic similarity thresholds, overlooking other important factors such as emotional appeal, presentation quality, or source credibility.

A major limitation of the implemented model is the distribution of probabilities of misinformation on topics. The implemented functionality for segregating a sub-section of topic vectors as more likely to be misinformation appears insufficient for capturing misinformation dynamics. Implementing a complex and functional mapping between topics and their probability distribution in terms of veracity could produce more desirable results.

Furthermore, the social network remains static throughout the simulation, missing important dynamics such as network growth, unfollowing behavior, or shifting community structures. Platform-specific features such as trending topics and algorithmic feed boosts are also not modeled, which could significantly influence misinformation diffusion.

The implementation of RAs is also a substantial feature that affects the dynamics of the simulation. In this study, UB-CF and IB-CF are based on LensKit algorithms, so it is reasonable to assume that their performance reflects how they work in reality. However, Pop and CBF are both abstractly implemented, which could leave a gap between how they perform in this study and in other academic papers, making it difficult to compare the results. Additionally, the algorithms tested represent a small subset of many possible recommendation strategies; thus, implementing more algorithms could have added further value to the analysis.

The choice of the model parameters also significantly affects the outcome of the experiments. Initially, it was desirable to employ a large social network with at least 1000 agents to enhance the realism of the experiments. However, as the model became more complex, the computational intensity increased rapidly. Due to limitations in time and computer power, the experiment was run with 100 agents. This amount may be insufficient to allow meaningful social dynamics and echo chambers to occur.

Finally, while the implemented RAs reflects real-world systems, they are simplified abstractions created manually. In reality, social media platforms incorporate hybrid recommendation systems that combine elements of the CBF, CF, and Pop

Figure 7.1: United Nations Sustainable Development goals

approaches. Evaluating these algorithms in isolation may, therefore, not fully capture the complexity in actual social network environments.

7.6 Sustainability

This thesis focuses on exploring how RAs affect the spread of misinformation on social media and if diversification of recommendations can limit the spread. The simulation that is created, along with the results of the experiments, can have several different impacts on sustainability. Therefore, a thorough analysis of how the study affects sustainability is performed, with reference to the 17 sustainability goals of the United Nations [69]. The overview of the different goals are presented in Figure 7.1, which is adapted from [70].

United Nations Sustainable Development Goals

The SDG's has been developed by the United Nations over decades and provide a shared plan for peace and sustainability for people and the planet, now and into the future. It consists of 17 different main goals, along with several more specific sub-goals. This study especially aligns with four of the main goals:

Nr 4 Quality Education

Nr 9 Industry, Innovation and Infrastructure

Nr 16 Peace, Justice and Strong Institutions

Nr 17 Partnership for the goals

Quality Education

SDG nr 4 includes ensuring inclusive, just, and high-quality education for all. Sub-goal nr 4.7 aligns with this study and is defined as *"Ensuring learners acquire knowledge to promote sustainable development"* [69]. Understanding how misinformation spreads on social media, especially when impacted by recommendation algorithms, is central to education about digital media literacy and critical thinking. The implemented ABM, which is open-source and available to all, can be applied in educational settings to help learners and institutions understand how echo chambers and algorithms affect news distribution. Access to visualizations of systematic effects, such as those provided by the dashboard, can help to enhance educational purposes and understanding.

The experiments and results in this thesis can also be used for educational purposes in software engineering by highlighting the types of algorithms that have the worst impact on the propagation of misinformation. Showing that the results confirm that recommendation algorithms can significantly amplify misinformation spread may help promote the development of ethical and socially sustainable platforms for future system or platform developers.

Industry, Innovation and Infrastructure

SDG 9.5 is defined as *"Enhance scientific research and upgrade technological capabilities"* [69]. This study uses ABM and a realistic simulation approach, which demonstrates an innovative approach to social system modeling. This contributes to academic and technological innovation using an innovative and realistic approach to research compared to the baseline method of using curated datasets. Research conducted on the role of recommendation systems in the spread of misinformation also informs future system developers about the need to make more ethical and nuanced choices.

Peace, Justice and Strong Institutions

SDG 16.10 encompasses *"Ensuring public access to information and protecting fundamental freedoms"* [69]. The study directly challenges the threat of misinformation, which undermines democratic discourse, institutional trust, and public

safety online, all of which are central aspects connected to SDG 16. In this thesis, it is examined whether increasing diversity in recommendation can work as a mitigation strategy against misinformation, and the findings can be used to further examine how to decrease the threat of misinformation on social media.

Partnerships for the goals

A central focus of the SDG's is to ensure good partnerships across the globe to reach the goals. SDG 17.6 is described as *"Enhancing knowledge sharing on mutually agreed terms"* [69]. The implemented model can be open-source and, therefore, shared with researchers, policymakers, and platforms for collaborative testing. This openness promotes interdisciplinary collaboration between the technology, social science, and policy communities and thereby aligns with SDG 17.6. Furthermore, the model is implemented to be modular, allowing researchers to easily extend and adjust the model to their preferences and research.

Chapter 8

Conclusion and Future Work

The conclusion of this study, presented in Section 8.1, revisits the research questions and the answers drawn from the results, while Section 8.2 outlines future work that has not been investigated in this study.

8.1 Conclusion

This study investigated the impact of various classical RAs on the spread of misinformation on social media. The experiments searched for answers on algorithmic performance with respect to misinformation propagation and compared the results with previous related work. This thesis explored whether the impact of recommendation systems on misinformation spread could be mitigated. The experimental approach tested encompassed increasing diversity in recommendations. Consequently, the study examined whether increased diversity could enhance the filtering effect of RAs against misinformation, reduce echo chambers, and identify which algorithms responded most and least efficiently to this approach.

RQ1: How do different RAs affect the spread of misinformation? The results that address RQ1 confirm expectations, showing that Pop is the most prone to recommending misinformation and leads to the highest propagation of misinformation within the simulated social network. This finding is consistent with previous studies. Another result substantiating other research is the strong performance of CBF regarding the spread of misinformation, likely due to its focus on content similarity and preference dynamics, as well as its lack of linking naive users together in clusters. In this study, the performance of the IB-CF algorithm is the result that differs the most from other evaluated studies. Alongside CBF, IB-CF is the most effective in limiting the spread of misinformation. UB-CF perform in line with previous studies, showing a tendency to propagate misinformation, but

without significantly amplifying it.

RQ2: How does increasing diversity in recommendations affect the spread of misinformation? The findings indicate that the propagation of misinformation is not only a function of the amount of recommended misinformation. The results show that although certain RAs, such as CBF and IB-CF, recommend more misinformation when diversity increases, they still cause a decrease in misinformation propagation. This suggests that despite the fact that the algorithms recommend misinformation, the responsibility for sharing and circulating it lies with the users. User behavior and network structure are the driving forces in spreading misinformation, not the algorithm. Moreover, the results show that some algorithms benefit more from the diversity increase than others. The Pop algorithm, in contrast, remains the worst performer in all conditions but recommends less misinformation as diversity increases. UB-CF is minimally affected by the diversity increase but also slightly recommends less misinformation. Ultimately, the impact of diversity depends on the underlying recommendation strategy.

RQ3 - How does the echo chamber effect change in relation to RA and increased diversity? The results of RQ3 show that all RAs exhibit a decrease in echo chambers as diversity increases. Pop show the highest reaction, with the highest relative decrease. However, comparing the decrease of the RAs' with their performance on misinformation exposure and spread provides mixed results. IB-CF and CBF results in a lower propagation of misinformation when the echo chambers decrease, while Pop and UB-CF remain stable or slightly worse. IB-CF also paradoxically show higher levels of misinformation in their recommendations when the echo chambers decrease. Thus, the results show that diversity can break up echo chambers; however, this effect could allow misinformation to reach new communities, further increasing the exposure to misinformation.

Compared to similar studies, the simulation model from this thesis adds a dynamic dimension to other static frameworks. With behavioral feedback, social network influence, and changing exposure patterns, it is possible to observe self-reinforcement and unintended side effects, characteristics that are often absent in traditional evaluation setups.

The results obtained from the study indicate that the use of an agent-based simulation approach is a good alternative to gain realistic insight into how recommendation systems affect the propagation of misinformation. In addition, it provides valuable information on how algorithms influence the spread over time. However, the implemented ABM has limitations, but with some updates to the functionality and complexity, a highly realistic social dynamic can be captured and used for more academic experimentation.

8.2 Future Work

This study opens several promising opportunities for future research, including the implementation of improvements to the model presented in this thesis. Enhancing behavioral realism for user agents by making them memory-driven and influenced by their behavior in the past could better reflect evolving beliefs and user behavior over time.

Furthermore, integrating real-world datasets as the foundation for the generation of news content would help simulate a more realistic topic distribution, as well as a more accurate representation of the relationship between topics and their likelihood of being misinformation. Future work could also explore more complex network dynamics, including temporal evolution, platform-specific rules, and more advanced RAs that better represent real-life applications.

In addition to improving the current functionality, the model is built to be modular and extensible, so researchers can integrate new aspects for exploration. Other mitigation tactics to limit misinformation could be applied to the simulation to further examine how to eliminate the threat of misinformation to society.

The implemented ABM highlights the potential of simulation-based studies to uncover recommenders' role in propagating information in a social network. Although the approach is limited by abstractions, the model provides a flexible approach to test mitigation and intervention strategies and explore complex dynamics in the social network.

Bibliography

[1] A. Hermida, F. Fletcher, D. Korell, and D. Logan, "Share, like, recommend: Decoding the social media news consumer," *Journalism Studies*, vol. 13, no. 5-6, pp. 815–824, 2012.

[2] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *ACM SIGKDD Explor. Newsl.*, vol. 19, 08 2017.

[3] G. M. Lunardi, G. M. Machado, V. Maran, and J. P. M. de Oliveira, "A metric for filter bubble measurement in recommender algorithms considering the news domain," *Applied Soft Computing*, vol. 97, p. 106771, 2020.

[4] R. Pathak, F. Spezzano, and M. S. Pera, "Understanding the contribution of recommendation algorithms on misinformation dissemination on social networks," *ACM Trans. Web*, vol. 17, Oct. 2023.

[5] A. Gausen, W. Luk, and C. Guo, "Using agent-based modelling to evaluate the impact of algorithmic curation on social media," *Journal of Data and Information Quality*, vol. 15, December 2022.

[6] E. Coppolillo, G. Manco, and A. Gionis, "Relevance meets diversity: A user-centric framework for knowledge exploration through recommendations," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, (New York, NY, USA), pp. 490–501, Association for Computing Machinery, 2024.

[7] C. Avin, H. Daltrophe, and Z. Lotker, "On the impossibility of breaking the echo chamber effect in social media using regulation," *Scientific Reports*, vol. 14, no. 1, p. 1107, 2024.

[8] M. Fernandez, A. Bellogín, and I. Cantador, "Analysing the effect of recommendation algorithms on the spread of misinformation," in *Proceedings of*

the 16th ACM Web Science Conference, WebSci '24, (New York, NY, USA), pp. 159–169, Association for Computing Machinery, 2024.

[9] A. Tommasel and F. Menczer, "Do recommender systems make social media more susceptible to misinformation spreaders?," in *Proceedings of the 16th ACM Conference on Recommender Systems*, RecSys '22, (New York, NY, USA), pp. 550–555, Association for Computing Machinery, 2022.

[10] R. Pathak and F. Spezzano, "Analyzing the interplay between diversity of news recommendations and misinformation spread in social media," in *Adjunct Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization (UMAP Adjunct '24)*, (New York, NY, USA), pp. 80–85, Association for Computing Machinery, 2024.

[11] J. Kazil, D. Masad, and A. Crooks, "Utilizing python for agent-based modeling: The mesa framework," in *Social, Cultural, and Behavioral Modeling* (R. Thomson, H. Bisgin, C. Dancy, A. Hyder, and M. Hussain, eds.), (Cham), pp. 308–317, Springer International Publishing, 2020.

[12] M. D. Ekstrand, "Lenskit for python: Next-generation software for recommender systems experiments," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, (New York, NY, USA), pp. 2999–3006, Association for Computing Machinery, 2020.

[13] A. Sandu, I. Ioanăș, C. Delcea, L.-M. Geantă, and L.-A. Cotfas, "Mapping the landscape of misinformation detection: A bibliometric approach," *Information (Switzerland)*, vol. 15, no. 1, 2024.

[14] X. Zhou and R. Zafarani, "A survey of fake news: Fundamental theories, detection methods, and opportunities," *ACM Computing Surveys*, vol. 53, 09 2020.

[15] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," *Journal of Economic Perspectives*, vol. 31, no. 2, pp. 211–236, 2017.

[16] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *Science*, vol. 359, pp. 1146–1151, Mar. 2018.

[17] T. Niitsuma, M. Yoshida, H. Tamori, and Y. Nakawake, "Prestige bias drives the viral spread of content reposted by influencers in online communities," *Scientific Reports*, vol. 15, no. 1, p. 15282, 2025.

[18] F. F. Leung, F. F. Gu, Y. Li, J. Z. Zhang, and R. W. Palmatier, "Influencer marketing effectiveness," *Journal of Marketing*, vol. 86, no. 6, pp. 93–115, 2022.

[19] L. Hagen, S. Neely, T. E. Keller, R. Scharf, and F. E. Vasquez, "Rise of the machines? examining the influence of social bots on a political discussion network," *Social Science Computer Review*, vol. 40, no. 2, pp. 264–287, 2022.

[20] O. Varol, E. Ferrara, C. Davis, F. Menczer, and A. Flammini, "Online human-bot interactions: Detection, estimation, and characterization," *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 11, 03 2017.

[21] M. R. DeVerna, R. Aiyappa, D. Pacheco, J. Bryden, and F. Menczer, "Identifying and characterizing superspreaders of low-credibility content on twitter," *PLOS ONE*, vol. 19, no. 5, p. e0302201, 2024.

[22] L. M. Aiello, A. Barrat, R. Schifanella, C. Cattuto, B. Markines, and F. Menczer, "Friendship prediction and homophily in social media," *ACM Trans. Web*, vol. 6, June 2012.

[23] S. Munusamy, K. Syasyila, A. Shaari, M. Pitchan, M. R. Kamaluddin, and R. Jatnika, "Psychological factors contributing to the creation and dissemination of fake news among social media users: a systematic review," *BMC Psychology*, vol. 12, 11 2024.

[24] F. Ricci, L. Rokach, and B. Shapira, *Introduction to Recommender Systems Handbook*, pp. 1–35. Boston, MA: Springer US, 2011.

[25] P. Brusilovsky, A. Kobsa, and W. Nejdl, *The Adaptive Web - Methods and Strategies of Web Personalization*. Springer Berlin/Heidelberg, 01 2007.

[26] Q. Zhao, F. M. Harper, G. Adomavicius, and J. A. Konstan, "Explicit or implicit feedback? engagement or satisfaction? a field experiment on machine-learning-based recommender systems," in *Proceedings of the 2018 ACM Symposium on Applied Computing (SAC)*, pp. 1331–1340, ACM, 2018.

[27] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender Systems: An Introduction*. New York, NY, USA: Cambridge University Press, 2011.

[28] C. Desrosiers and G. Karypis, *A Comprehensive Survey of Neighborhood-based Recommendation Methods*, pp. 107–144. Boston, MA: Springer US, 2011.

[29] H. Ko, S. Lee, Y. Park, and A. Choi, "A survey of recommendation systems: Recommendation models, techniques, and application fields," *Electronics*, vol. 11, 01 2022.

[30] M. Kunaver and T. Požrl, "Diversity in recommender systems – a survey," *Knowledge-Based Systems*, vol. 123, pp. 154–162, 2017.

[31] D. Kotkov, J. Veijalainen, and S. Wang, "How does serendipity affect diversity in recommender systems? a serendipity-oriented greedy algorithm," *Computing*, vol. 102, no. 2, pp. 393–411, 2020.

[32] S. Caton and C. Haas, "Fairness in machine learning: A survey," *ACM Comput. Surv.*, vol. 56, Apr. 2024.

[33] Y. Zhao, Y. Wang, Y. Liu, X. Cheng, C. Aggarwal, and T. Derr, "Fairness and diversity in recommender systems: A survey," *ACM Transactions on Intelligent Systems and Technology*, vol. 16, 05 2024.

[34] A. Sacilotti, R. F. d. Souza, and M. G. Manzato, "Counteracting popularity-bias and improving diversity through calibrated recommendations," in *International Conference on Enterprise Information Systems - ICEIS*, SciTePress, 2023.

[35] J. Carbonell and J. Goldstein, "The use of mmr, diversity-based reranking for reordering documents and producing summaries," in *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, (New York, NY, USA), pp. 335–336, Association for Computing Machinery, 1998.

[36] C. Wu, F. Wu, T. Qi, and Y. Huang, "End-to-end learnable diversity-aware news recommendation," 2022.

[37] L. Chen, G. Zhang, and H. Zhou, "Fast greedy map inference for determinantal point process to improve recommendation diversity," in *Advances in Neural Information Processing Systems (NeurIPS 2018)*, pp. 5627–5638, 2018.

[38] E. Bonabeau, "Agent-based modeling: Methods and techniques for simulating human systems," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99 Suppl 3, pp. 7280–7287, 06 2002.

[39] S. Govindankutty and S. Gopalan, "Epidemic modeling for misinformation spread in digital networks through a social intelligence approach," *Scientific Reports*, vol. 14, p. 19100, 2024.

[40] V. Thomopoulos and K. Tsichlas, "An agent-based model for disease epidemics in greece," *Information*, vol. 15, 03 2024.

[41] H. Wang, F. Wang, and K. Xu, *Ordinary Differential Equation Models on Social Networks*, pp. 3–13. Cham: Springer International Publishing, 2020.

[42] R.-S. Wang, *Ordinary Differential Equation (ODE), Model*, pp. 1606–1608. New York, NY: Springer New York, 2013.

[43] J.-H. Niemann, S. Uram, S. Wolf, N. Djurdjevac Conrad, and M. Weiser, "Multilevel optimization for policy design with agent-based epidemic models," *Journal of Computational Science*, vol. 77, p. 102242, 2024.

[44] K. Heng and C. L. Althaus, "The approximately universal shapes of epidemic curves in the susceptible-exposed-infectious-recovered (seir) model," *Scientific Reports*, vol. 10, p. 19365, 2020.

[45] M. Fan, M. Y. Li, and K. Wang, "Global stability of an seis epidemic model with recruitment and a varying total population size," *Mathematical Biosciences*, vol. 170, no. 2, pp. 199–208, 2001.

[46] P. Kumar and A. Sinha, "Information diffusion modeling and analysis for socially interacting networks," *Social Network Analysis and Mining*, vol. 11, December 2021.

[47] D. J. Watts, "A simple model of global cascades on random networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 9, pp. 5766–5771, 2002.

[48] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, (New York, NY, USA), pp. 137–146, Association for Computing Machinery, 2003.

[49] B. Hornig, M. S. Pera, and J. Scholtes, "Misinformation in video recommendations: An exploration of top-n recommendation algorithms," in *Proceedings of the 4th Workshop on Reducing Online Misinformation through Credible Information Retrieval (ROMCIR 2024), co-located with the 46th European Conference on Information Retrieval (ECIR 2024)*, vol. 3677, (Glasgow, UK), pp. 54–67, CEUR Workshop Proceedings, March 2024.

[50] M. Kaya, S. Conley, and A. Varol, "Visualization of the social bot's fingerprints," in *Evaluating recommender systems from the user's perspective: survey of the state of the art.*, p. 161 – 166, 2016. Cited by: 4.

[51] L. Valentine, S. D'Alfonso, and R. Lederman, "Recommender systems for mental health apps: advantages and ethical challenges," *AI and Society*, vol. 38, no. 4, pp. 1627–1638, 2023.

[52] E. Magrani and P. G. F. da Silva, "The ethical and legal challenges of recommender systems driven by artificial intelligence," *Law, Governance and Technology Series*, vol. 58, pp. 141–168, 2024.

[53] D. Paraschakis, "Towards an ethical recommendation framework," in *Proceedings of the 11th International Conference on Research Challenges in Information Science (RCIS)*, pp. 211–220, 2017.

[54] B. D. Horne, M. Gruppi, and S. Adalı, "Trustworthy misinformation mitigation with soft information nudging," in *Proceedings of the 2021 Workshop on Trustworthy AI for Multimedia Computing (TAIMC)*, (New York, NY, USA), pp. 1–8, ACM, 2021.

[55] N. Vo and K. Lee, "The rise of guardians: Fact-checking url recommendation to combat fake news," in *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '18)*, (New York, NY, USA), pp. 275–284, ACM, 2018.

[56] S. Wang, X. Xu, X. Zhang, Y. Wang, and W. Song, "Veracity aware and event-driven personalized news recommendation for fake news mitigation," in *Proceedings of the ACM Web Conference 2022 (WWW '22)*, pp. 3673–3682, Association for Computing Machinery, 2022.

[57] D. Sallami, R. B. Salem, and E. Aïmeur, "Trust-based recommender system for fake news mitigation," in *Adjunct Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization (UMAP '23 Adjunct)*, pp. 104–109, Association for Computing Machinery, 2023.

[58] M. D. Ekstrand, M. Tian, I. M. Azpiazu, J. D. Ekstrand, O. Anuyah, D. McNeill, and M. S. Pera, "All the cool kids, how do they fit in?: Popularity and demographic biases in recommender evaluation and effectiveness," in *Proceedings of the Conference on Fairness, Accountability and Transparency (FAT* '18)*, vol. 81 of *Proceedings of Machine Learning Research*, pp. 172–186, PMLR, 2018.

[59] B. Rastegarpanah, K. P. Gummadi, and M. Crovella, "Fighting fire with fire: Using antidote data to improve polarization and fairness of recommender systems," in *Proceedings of the 12th ACM International Conference on Web Search and Data Mining (WSDM '19)*, (New York, NY, USA), pp. 231–239, ACM, 2019.

[60] M. Jiang, Q. Gao, and J. Zhuang, "Reciprocal spreading and debunking processes of online misinformation: A new rumor spreading–debunking model with a case study," *Physica A: Statistical Mechanics and its Applications*, vol. 565, p. 125572, 2021.

[61] A. Guzman Rincon, S. Barragan Moreno, B. Rodriguez-Canovas, *et al.*, "Social networks, disinformation and diplomacy: a dynamic model for a

current problem," *Humanities and Social Sciences Communications*, vol. 10, p. 505, 2023.

[62] A. Ghoshal, N. Das, S. Das, *et al.*, "Minimizing spread of misinformation in social networks: a network topology based approach," *Social Network Analysis and Mining*, vol. 15, no. 15, 2025.

[63] F. C. Erd, A. L. Vignatti, and M. V. G. da Silva, "Blocking the spread of misinformation in a network under distinct cost models," in *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 232–236, 2020.

[64] Q. F. Lotito, D. Zanella, and P. Casari, "Realistic aspects of simulation models for fake news epidemics over social networks," *Future Internet*, vol. 13, no. 3, p. 76, 2021.

[65] J. Brainard, P. R. Hunter, and I. R. Hall, "An agent-based model about the effects of fake news on a norovirus outbreak," *Revue d'Épidémiologie et de Santé Publique*, vol. 68, pp. 99–107, 2020. Available online 6 February 2020.

[66] G. Ahmed, "Analysis and simulation of misinformation spread in social networks: A hybrid stochastic-deterministic approach with neab model," *Communications on Applied Nonlinear Analysis*, vol. 32, pp. 550–560, 09 2024.

[67] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences*, vol. 105, pp. 1118–1123, Jan. 2008.

[68] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008.

[69] D. o. E. United Nations and S. A. S. Development, "Transforming our world: the 2030 agenda for sustainable development," 2015.

[70] United Nations, "Sustainable development goals (sdgs)," n.d. Accessed: 2025-05-27.

Appendix A

Parameter Configuration

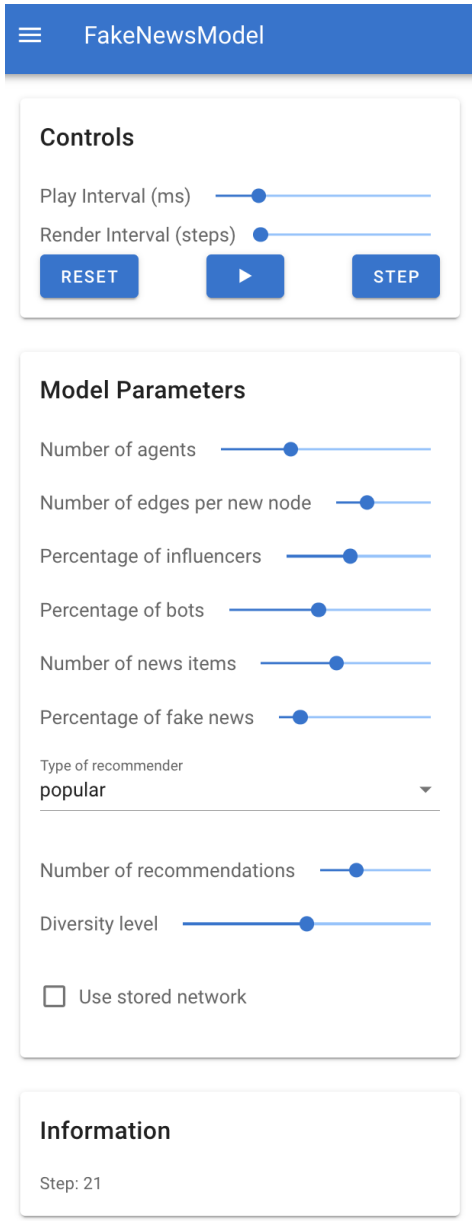

Figure A.1: Parameter configuration for the Dashboard

Appendix B

Dashboard

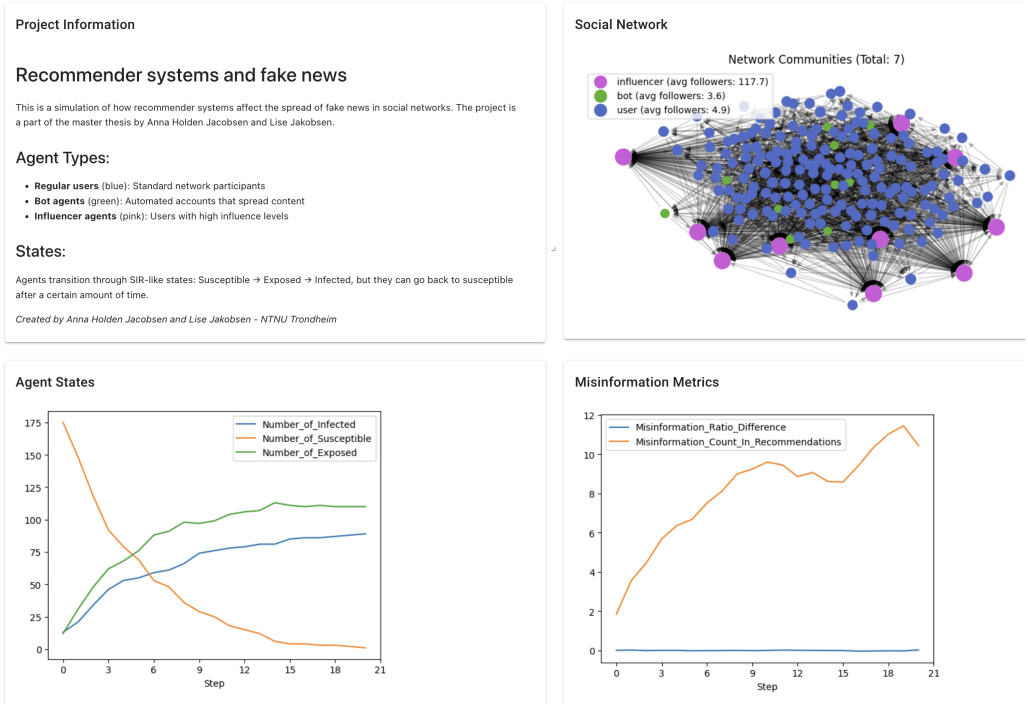

Figure B.1: Overview of the dashboard