ELECTRONICS AND COMPUTER SCIENCE

FACULTY OF PHYSICAL SCIENCES AND ENGINEERING

UNIVERSITY OF SOUTHAMPTON

Author
DANNY MARTINEZ HIBBERT

April 28, 2024

Final Report

Simulating social networks with user content bias to improve the analysis of inauthentic user behaviour

Primary project Supervisor
Dr BOOJOONG KANG

Secondary project Supervisor
Dr JIAN SHI

A final report submitted for the award of BSc Computer Science

Abstract

This report presents a study aimed at advancing and improving OSN (online social networks) simulation methods to help in the analysis of botnet behaviour. The focus of the simulation method is through the use of a topic bias distribution to create targeted inauthentic node behaviour. The software project presented delves into the design and implementation of the base simulation model and its topic bias addition to shape user engagement within the network, similarly to a targeted spamming campaign.

Utilizing the SimSoM model as a foundational framework, this study introduces key modifications to incorporate topic bias, a novel feature that significantly enhances the model's realism in simulating information diffusion and its applicability in detecting inauthentic nodes.

The evaluation of the simulation tool demonstrates its effectiveness in replicating the complex interplay between authentic and inauthentic users within social networks, showcasing the impact of topic biases on information diffusion and user engagement. The findings highlight the tool's potential as a resource for researchers seeking to generate tailored datasets for studying the nuances of social media dynamics and bot influence.

<u>Statement of Originality</u>

- I have read and understood the ECS Academic Integrity information and the University's Academic Integrity Guidance for Students.
- I am aware that failure to act in accordance with the Regulations Governing Academic Integrity may lead to the imposition of penalties which, for the most serious cases, may include termination of programme.
- I consent to the University copying and distributing any or all of my work in any form and using third parties (who may be based outside the EU/EEA) to verify whether my work contains plagiarised material, and for quality assurance purposes.

You must <u>change the statements in the boxes</u> if you do not agree with them.

We expect you to acknowledge all sources of information (e.g. ideas, algorithms, data) using citations. You must also put quotation marks around any sections of text that you have copied without paraphrasing. If any figures or tables have been taken or modified from another source, you must explain this in the caption <u>and</u> cite the original source.

I have acknowledged all sources, and identified any content taken from elsewhere.

If you have used any code (e.g. open-source code), reference designs, or similar resources that have been produced by anyone else, you must list them in the box below. In the report, you must explain what was used and how it relates to the work you have done.

I have not used any resources produced by anyone else.

You can consult with module teaching staff/demonstrators, but you should not show anyone else your work (this includes uploading your work to publicly-accessible repositories e.g. Github, unless expressly permitted by the module leader), or help them to do theirs. For individual assignments, we expect you to work on your own. For group assignments, we expect that you work only with your allocated group. You must get permission in writing from the module teaching staff before you seek outside assistance, e.g. a proofreading service, and declare it here.

I did all the work myself, or with my allocated group, and have not helped anyone else.

We expect that you have not fabricated, modified or distorted any data, evidence, references, experimental results, or other material used or presented in the report. You must clearly describe your experiments and how the results were obtained, and include all data, source code and/or designs (either in the report, or submitted as a separate file) so that your results could be reproduced.

The material in the report is genuine, and I have included all my data/code/designs.

We expect that you have not previously submitted any part of this work for another assessment. You must get permission in writing from the module teaching staff before re-using any of your previously submitted work for this assessment.

I have not submitted any part of this work for another assessment.

If your work involved research/studies (including surveys) on human participants, their cells or data, or on animals, you must have been granted ethical approval before the work was carried out, and any experiments must have followed these requirements. You must give details of this in the report, and list the ethical approval reference number(s) in the box below.

My work did not involve human participants, their cells or data, or animals.

ECS Statement of Originality Template, updated August 2018, Alex Weddell aiofficer@ecs.soton.ac.uk

Acknowledgements

Thanks to *Bao Tran Truong, Xiaodan Lou, Alessandro Flammini and Filippo Menczer* from the *Observatory on Social Media* for the great OSN model this project is based on[29] which otherwise would not have been possible.

Thanks to *Haokai Lu, James Caverlee, Wei Niu* for helping inspire the implementation of the algorithm for topic bias through their 'BiasWatch'[20] approach.

Contents

1 Introduction

1.1 Problem

Online social media platforms are increasingly challenging to manage due to their expanding use and influence. The prevalence of fake profiles has escalated, allowing private interest groups to exploit these platforms [31]. Although there are some theoretical frameworks for simulating botnets within social networks, they fall short of addressing the complex interactions and dynamics among users.

There's a pressing need for researchers to delve into and understand the multifaceted aspects of social networks, particularly how their structures are manipulated by malicious actors using fake accounts. Yet, there is a noticeable absence of publicly accessible simulations of social platforms that examine the impact of content topics on user influence.

In addressing the challenge of bot detection, machine learning emerges as a cutting-edge solution with great results. Machine learning techniques analyse network and agent behaviour to detect inauthentic users and edge cases [30]. However, this approach is hampered by a dearth of datasets and significant issues with API access. A dedicated social network simulation tool could fill this gap by generating tailored datasets that meet specific research needs, thereby advancing our understanding of and ability to combat malicious activities on social media platforms.

1.1.1 API access restrictions and lack of social network simulators

Over the recent years the most popular providers of OSN platforms such as Meta and twitter have severely restricted access to important data in regards to social trends on their platforms, instead favouring data that correlates to commercial purposes such as advertising reach [16].

Twitter is a very active area of research, their new API accessibility policies have changed, forcing many researchers to abandon their projects[1]. This decision was done do help the platforms sustainability[2]. Most researches spend a lot of time developing novel data extraction tools to circumvent the lack of API accessibility[7]. A simulation model could alleviate a lot of these concerns or contribute to moving away from API access.

1.2 Goal

The project presented in this thesis aims to alleviate the described problem by designing a tool that simulates the effects of bots in a social network and describes the way in which user engagement is shaped. The main purpose of the tool is to provide a more complex simulation that captures the underlying aspects of message re-sharing through the use of user topic interest to help detect inauthentic user behaviour.

The broader use case is to help researchers in this field in the future by providing an

[1] `https://tinyurl.com/5a5spcyj`

[2] `https://tinyurl.com/4x3my39t`

accurate simulation and avoid relying on existing private API's as well as avoid the time and financial bottleneck of sourcing data[1, 4, 8].

1.3 Scope

The key point of the presented simulation model is to research the effects of inauthentic posts on user engagement through a simulation of topic bias, where each user in the network is biased towards engaging with certain topics.

The simulation is scaled realistically for the hardware available to avoid any dependencies on fast hardware, scalable solutions in OSN (online social networks) is a different problem [14]. A more in depth analysis of efficiency in the implemented algorithms, such as message propagation or topic comparison is out of the scope of this project. The model will primarily use two key parameters to socialize the nodes with each other, engagement and quality of message/post.

There are primarily two types of social media networks, directional and bidirectional, this project focuses on directional networks, such as twitter based models. For example - when a user communicates with another user, it does not necessarily imply the followed user will communicate back [15]. This unilateral nature of connections allows for a wide range of network structures, from highly reciprocal connections between users who follow each other to broad, non-reciprocal connections, like those between celebrities and their followers [24].

2 Literature review

The presented research will focus on the current implementations of OSN simulation models, how they are constructed as well as their common aspects and elements.

2.1 The gravity model

Xiaochao Wei et al. in "Product diffusion in dynamic online social networks: A multi-agent simulation based on gravity theory"[32] proposes a 'gravity' method for simulating interactions between users in a OSN. This approach is effective as analysing the weighted edges within clusters of users in the network. The model focuses on susceptibility to advertisements on the OSN platform.

2.2 The SMSim model

The field of OSNs is lacking in depth for the study of information diffusion, mostly consisting of high level analysis and do no focus on explaining individual user behaviour[5]. Simulation based approaches tend to have a specific use case for the analysis of general sentiment analysis - not spam detection. The following are some examples.

Maira A de C Gatti, et al. in "A simulation-based approach to analyze the information diffusion in microblogging online social network"[5] the model focuses on predicting user behaviour bu analysing the emergent behaviour from posting messages in a ’Twitter like’ social network in reference to a particular topic/narrative. The approach by this model This model doesn’t capture the engagement of the messages topics, each message is attributed with a ’good’ or ’bad’ sentiment. Another problem is the model is based strictly on Twitter’s social structure which doesn’t represent another platform such as Instagram or Facebook. Ultimately the model fails to capture the most influential users in the network, useful for predictive behaviour modelling but not for detecting bots.

2.3 The SimSoM model

This project will use the model from Truong, B. T., et al (2023). "Vulnerabilities of the Online Public Square to Manipulation"[29]. The paper discusses the SimSoM model, an agent-based simulation designed to explore information diffusion in social networks infiltrated by malicious actors.
The SimSoM model is the most sophisticated agent-based simulation designed to analyze the dynamics of information diffusion within a social network, with a particular focus on the influence of inauthentic accounts. This model is perfect for the needs of this project due to its focus on agent behaviour through the application of parameters that change its behaviour for flexibility and control.

2.3.1 SimSoM’s key parameters

SimSoM has a few key parameters that are used to manipulate the output of the simulation by altering parts of its behaviour such as agent behaviour and social structure. These parameters will be fully adopted into our model and the topic distribution bias will be added on top.

Time step(t): Similarly to the SMSim model [5] SimSoM has a sequential time series.

Information Load (μ): Represents the frequency at which new content is introduced into the system versus re shared content.

Agent Attention (α): This models the limited attention span of social media users by defining the size of a user’s news feed.

Engagement of Messages (e_m): Influences the likelihood of the message being re shared.

Quality of Messages (q_m): Reflecting the objective desirable properties of content, such as originality or truthfulness.

Prevalence of Bots (β): Parameters control the ratio of bots to authentic accounts.

Infiltration of Bots (γ): To what degree have inauthentic accounts over taken the network in regards to connections to other nodes.

Deception Parameter (φ): This parameter models the likelihood of bot-generated content being irresistibly engaging, regardless of its quality.

Flooding (θ): This parameter represents the extent to which bots can spam the network with messages. Equivalent to a spam attack.

2.4 Rogers' Diffusion of Innovations Theory

Most OSNs base their network diffusion on Roger's diffusion theory. For instance, the adoption of features like Stories, Reels, or network formation algorithms can be analyzed through the lens of knowledge, persuasion, decision, implementation, and confirmation stages[3]. "Diffusion is the process by which an innovation communicated through certain channels over time among the members of a social system. Diffusion is special type of communication in which the massages are about a new idea" [17, 5].

2.5 Network generation and diffusion models

The basis of any OSN simulation model needs an appropriate user distribution to reflect a realistic network. Real social media networks with the usual expected functionality (where a user in the network can follow anyone else in the network) will inter-connect users following a power-law distribution[6]. In fact the most popular social media networks such as Facebook and Twitter feature an almost exclusive power law distribution for their entire network [2].

Social platform simulation tools are severely limited in regards to commercially or open source software. This area of research mainly involves research applicants from statistical and psychology backgrounds - this is where this project derives a lot of its background information in regards to existing implementations.

Most social media networks tend to follow a power law distribution, especially in terms of connections (followers, friends) and content distribution (likes, shares). A power law distribution in this context means that a small number of users have a disproportionately large number of connections or interactions, while the vast majority of users have relatively few.

This phenomenon is often referred to as the "80/20 rule" or Pareto principle[4], where roughly 20% of the users hold 80% of the connections or influence.

2.5.1 Preferential attachment

This is the most common explanation for power law distributions in social media networks. "The mechanism of preferential attachment assumes that a vertex's probability of receiving a new edge is proportional to the number of edges it already has."[26]. This is an interesting algorithm that has each new node in the network establish an edge with nodes that have the most edges already through a probability bias.

[3] https://files.eric.ed.gov/fulltext/ED501453.pdf

[4] `https://en.wikipedia.org/wiki/Pareto_principle`

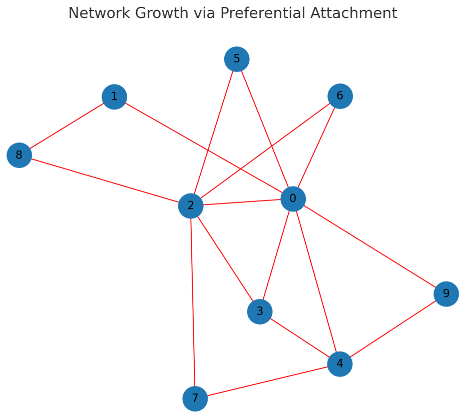


Figure 1: This graph was generated using the Barabási–Albert model, starting with a small number of interconnected nodes and then adding new nodes that preferentially attach to the more connected ones. Nodes 2 and 0 have the most vertices.

2.5.2 Triadic closure

This method of social network formation is also very popular for formulating simulated networks. It relies on the principle of association. Newly added nodes to the network are likely to establish edges with nodes if they have other nodes(friends) in common with each other[26, 3].

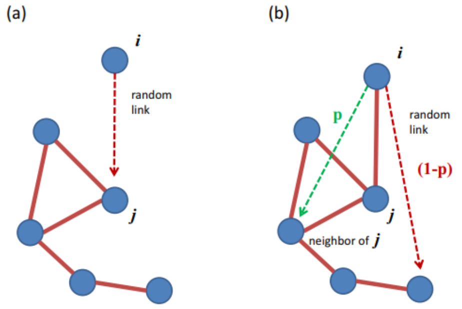


Figure 2: "Basic model. One link associated with a new node i is attached to a randomly chosen node j, the other links are attached to neighbors of j with probability p, closing triangles, or to other randomly chosen nodes with probability 1 - p" [3]

.

2.5.3 Homophily

In this case, new nodes added to the network are more likely to form edges with nodes that share similar features. This is likely to cause clusters, for example reddit asks new users to join groups upon joining the network, creating clusters is what they are looking do to as the network operates on ’subreddit’ clusters or groups[26, 21].

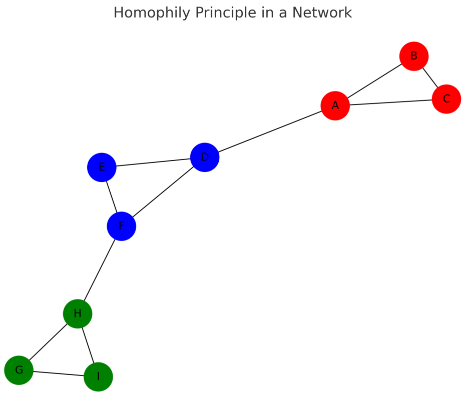


Figure 3: In this graph, nodes are more likely to connect with others within the same group, which is denoted by their color.

2.5.4 Reciprocity

Another phenomena that might explain the power law distribution in social networks is the theory of users reciprocating an edge formation in directed networks. If node A follows B, then B is more likely to be following A[26, 11].

In terms of popularity and current relevance, Preferential Attachment and Homophily are particularly prominent in explaining the dynamics of most large-scale social media networks, due to their focus on network growth and content sharing. Triadic Closure is also a significant model, especially in networks with a strong emphasis on personal relationships. Reciprocity is more specific to networks where mutual connections are a core feature.

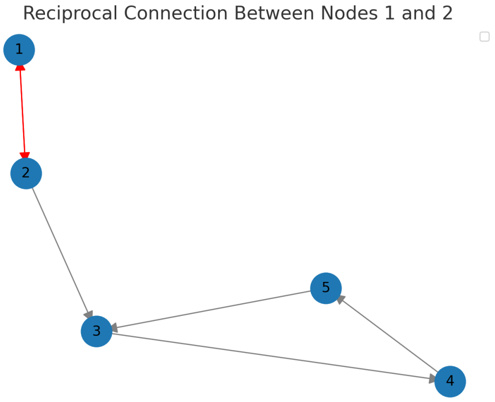


Figure 4: The red connection shows a reciprocal connection, node 1 is friends with node 2 through mutual interest, hence the bidirectional edge.

2.6 Agent generation

An agent in the network is either an authentic or inauthentic user. User agents within the OSN can vary greatly depending on the application of the simulation and type of data required. User interactions can be very specific to certain platforms that offer unique features as part of their service. Agents usually should be generic in nature in order to reflect users in different social media networks such as Facebook, twitter, tiktok, etc. Some of the agent behaviours of interest are based on a mix of active, rest, and sleep states. Another key area of agent behaviour is generation of novel messages and the re-sharing of existing ones [9].

2.6.1 Authentic agents

A general assumption to make in regards to authentic users within the network is that they naturally join and interact based on their interests, forming connections and sharing content that reflects their real-life experiences[22]. This type of behaviour is fundamentally different from a user that is trying to subvert the network. Real users in the network engaging with the community and contributing to it naturally, no malicious activity or harm is intended[27]. The output of these nodes is generally considered to be of high quality with varying degrees of engagement.

2.6.2 Inauthentic users

These types of agents are strategically created and deployed to manipulate social media platforms. They often operate in networks, using both real and fake accounts to amplify specific narratives or silence opposition. Their behavior includes the coordination of content sharing, leveraging platform algorithms for greater visibility, and engaging in harassment campaigns against individuals or groups[22].

2.7 Topic exploitation for increase information diffusion

More sophisticated botnets within an OSN have used topic bias to increase the influence of their campaigns by tailoring produced content to complement current trends. These types of agents are defined by the Department of Social Sciences, University of Naples Federico II, Naples, Italy as CIB's (Coordinated Inauthentic Behavior)[23]. They strategically spread misinformation and manipulate public opinion by amplifying divisive narratives, in this example in regards to COVID-19 vaccines. This exploitation of topic bias enhances information diffusion by engaging users more likely to share content that aligns with their pre-existing beliefs, thereby increasing the spread of inauthentic content across networks[23].

It has been observed that there's a higher activity of bot behaviour in OSN when scanning popular platforms such as Facebook and twitter [33] showing a tendency for bots to target topic bias. This is more visible on platforms such as Twitter that have trending topics through the hashtag system. Research presented in the Journal of Global Security Studies by Oxford Academic focused on the strategic use of social media networks by states for disinformation campaigns. It specifically examined how entities like RT and Sputnik, linked to the Kremlin, targeted networks of social media users by mimicking cultural cues to camouflage their intentions and attract sympathizers [19].

2.8 An early measure of media bias

Tim Groseclose and Jeffrey Milyo utilised an empirical method to quantify media bias in relation to the ideological positions of media outlets[13]. The research is significant as it was one of the first quantitative analysis of topic bias. The researchers analyzed citations of think tanks and policy groups in media outlets in order to compare them to citations by members of American Congress in their speeches.

2.9 Sentiment analysis

Sentiment analysis is a dependable and relatively early approach to analysing user sentiment within in social media networks. Tan et al. [28] propose a form of sentiment analysis that uses complimentary information about the users friendship connections - helping establish a link between social engagement and topic bias. This approach as per so many in sentiment analysis requires large amounts of labelled data.

2.10 Matrix factorization

A modern approach by Preethi L. et al.[18] uses a similar mathematical technique to recommended systems by utilizing ml learning by combining 2 factorized matrices representing the social network structure and the propagated content within.

2.11 BiasWatch

'BiasWatch' models a bias propagation methods over a set of topics to estimate the amount of bias in a social media post[20]. BiasWatch's modelling method is relevant to the purposes of the simulation method in this thesis. Each user in the network is associated with a unique timeline with their own tweets - each tweet is associated with a unique topic. Each user is assigned a probability of susceptibility to a particular topic ranging from (-1,1). A user similarity network is built and measures the similarity between any two users. This approach is great as it doesn't require labelled data and instead use hashtags as seeds. This approach is also more relevant as it focuses on bias in regards to trends instead of sentiments. "sentiment analysis centers around users attitude or emotional state, usually reflected by the use of emotional words."[25, 20].

3 Planning and design

The main goal of this project is to implement the SimSoM[29] model and improve upon it by adding a layer of abstraction for topic bias to capture more realistic metrics for analysis purposes, such as detecting inauthentic users or producing a dataset which could be used for machine learning purposes. Therefore this paper will implement the following key aspects:

- **1.** A network generation method
- **2.** An agent generation for authentic and inauthentic nodes
- **3.** An inauthentic agent infiltration method
- **4.** Network manipulation through parameters

- **5.** Agent behaviour manipulation through parameters
- **6.** Message generation and re-sharing method
- **7.** Topic bias algorithm

The backend of the project is implemented using python3. The reason for choosing python is to take advantage of the rich graph and visual libraries - allowing the project to progress faster and preventing the need to write common algorithms from scratch. Another main reason for using python is to leverage its support for web interface integration. The frontend is used to display network information for the user to easily see any changes and developments in the network.

3.1 Github

All code and utilities are kept in a github repository, this allows the code to be version controlled and managed appropriately. It also allows other developers to join the project and review code changes.

3.2 Lucid chart

The Lucid chart service is used to provide a detailed UML diagram to visualise sections of the application - detailing the different constituent parts and their sub elements.

3.3 The code structure

Python is a dynamic interpreted language that allows the use of object oriented programming as well as procedural programming, the project features both paradigms where appropriate. The project is split into separate modules for increased flexibility.

Files

- **App.py** The entry point for the application, contains all globally defined parameters
- **data_extraction.py** Returns dataset of entire network with its feature set
- **interface.py** Frontend code for displaying all visual representations
- **messaging.py** All logic to do with the way nodes send messages to eachother within the network
- **network.py** The logic behind the formulation of the network structure
- **plotter_*.py** All logic for generating plotted graphs

- **topic_distribution.py** Logic for generating topic biases for each node in the network

Libraries

- **networkx** For generating complex graph data[5]

- **dash + visdcc** For providing the frontend interface[6]

- **numpy** For general math applications[7]

- **plotly** For plotting data[8]

3.4 Module structure

The project is split into modules for better separation of concerns, their dependencies are shown as arrows. The **network** module is built first as its the base for all other data. The **messages** module is next as its dependent on the **network** module by generating messages for each timeline. The **plotter_*** library is very important for making sense of all the data in the network, from individual nodes to distributed messages. Finally **data_extraction** and **interface** can be finalised independently. The very last module is the **topic_distribution** which can be added to the system at the very end as an extra abstraction layer.

3.5 Code structure

The software is written in python3, therefore each part has been implemented according to the best practices[9]. This approach allows the written code to be easy to understand and consistent throughout.

Software modules

[5] https://networkx.org/

[6] https://dash.plotly.com/

[7] https://numpy.org/

[8] https://plotly.com/

[9] https://peps.python.org/pep-0008/

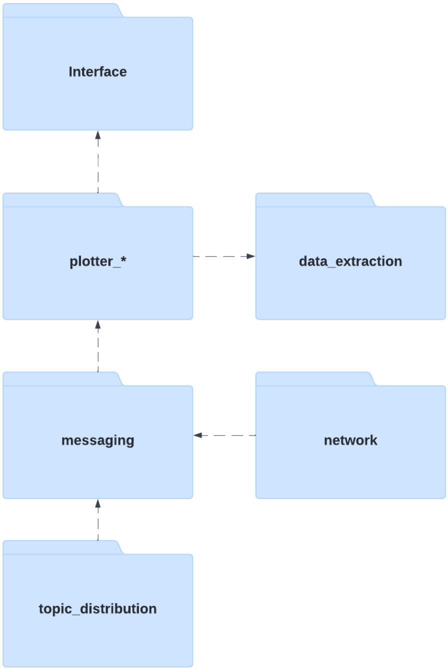


Figure 5: Software modules and their relationship

3.6 Topic bias design

The topic system is separated into two parts. The first part of the system is broken deals with the formation of each topic by providing a genre and an associated probability of its presence within the network. During the population of the network each node is attributed a topic. The second part attributes each newly generated message a topic which targets a node with a matching node - this is performed in reference to the probability distribution when the message is generated per real-time step.

4 Implementation

4.1 Network

In social networks, this means a small number of nodes (hubs) have many connections, while most nodes have few. This is characteristic of scale-free networks.

Random-walk growth models such as preferential attachment as outlined earlier in the report leads to a network structure that exhibits properties of networks following a power-law distribution, this approach is used in this report as follows.

The preferential attachment mechanism, which is key to the emergence of scale-free networks, implies that the probability $\boldsymbol{P(k)}$ that a node in the network has $\boldsymbol{k}$ connections follows a power-law distribution:

Degree distribution:

$$\mathbf{P(k) \sim k^{-\gamma}}$$

Where γ is a parameter typically in the range $2 < \gamma < 3$ for many real-world networks.

The growth model, through its iterative process such that new nodes may attach to nodes based on existing connections (with $\mathit{probability}\ p$) or link to any node in the network (with $\mathit{probability}\ 1 - p$), encourages a network structure with hubs and clustering. These features are indicative of the scale-free and small-world properties, respectively. Both properties can be described by power-law or similar heavy-tailed distributions in their degree distributions:

Formally:

$$p_i = \frac{k_i}{\sum_j k_j}$$

$$\text{For each edge } e_i : \begin{cases} \text{Select a friend of } j \text{ with probability } p, \\ \text{Select any node from the network with probability } 1 - p. \end{cases}$$

Here, j is a node already in the network that has been randomly selected as a starting point for the new edge. This process, while not explicitly preferential attachment, leads to a similar emergent behavior where the network develops characteristics of scale-free networks. This slightly modified preferential attachment approach more closely mimics are real social network formation by capturing both scale-free and small-world properties where not all connection are made purely based on existing node degrees.

4.2 Agents

As mentioned previously the user agents within the network will be composed of certain mathematical properties to help define their message generation and resharing behaviour. The following are the agents properties and their relationships.

Bot Generation and Infiltration Parameters:

n: Total number of authentic nodes.

β: Ratio of bots to authentic accounts in the network.

γ: Probability that an authentic account follows a bot, modeling bot infiltration.

Q_t: The total quality of the node, represented by the sum of the quality of all messages in the timeline.

$\boldsymbol{E_t}$: The total engagement of the node, represented by the sum of the engagement of all messages in the timeline.

$\boldsymbol{m}$: Number of edges from inauthentic to authentic nodes.

$$\boldsymbol{Number\ of\ inauthentic\ nodes} = \frac{n * \beta}{m}$$

Engagement and Quality of Content for each agent:

For authentic accounts, engagement equals quality:

$$\boldsymbol{q = e}$$

Meaning the value of the content is directly tied to how users interact with it

With high-quality information assumed to be rare. Quality and engagement are drawn from an engagement distribution where,

$$\boldsymbol{P(e) = 2(1 - e)}$$

Meaning lower engagement values are more common, as the probability increases as e approaches 0.

For bots, all content is of low quality:

$$\boldsymbol{q_m = 0}$$

but can have deceptively high engagement. This is modeled by the deception parameter $\boldsymbol{\theta}$, where:

$$\boldsymbol{e_m = min(q_m + \theta, 1)}$$

with probability $\boldsymbol{\theta}$, making some bot-generated content highly engaging.

Parameter $\boldsymbol{\theta}$ indicates how much more content a bot generates compared to an authentic account, representing the bot's spamming behavior.

4.3 Messaging

Message Selection and Sharing:

Probability of a message $\boldsymbol{m}$ being selected from a user's feed for sharing is proportional to its engagement:

$$\boldsymbol{P(m)} = \frac{e_m}{\sum_{j \in M_i} e_j}$$

where $\boldsymbol{M_i}$ is the set of messages in user $\boldsymbol{i}$'s feed. The probability of selecting a message for sharing is proportional to its engagement.

Effect on Information Quality:

The overall quality of the system at time $\boldsymbol{t}$ is given by:

$$\boldsymbol{Q_t} = \frac{1}{\alpha N} \sum_{i=1}^{N} \sum_{m \in M_i} \boldsymbol{q}_m^t$$

where $\boldsymbol{q}_m^t$, item Is the quality of message $\boldsymbol{m}$ in user $\boldsymbol{i}$'s feed at time $\boldsymbol{t}$, $\boldsymbol{N}$ is the number of authentic accounts, and $\boldsymbol{\alpha}$ is the size of a user's feed. Therefore the overall quality of information Q_t is assessed as an average across all messages in all users' feeds, adjusted for the number of messages α seen by each user

Visual example of a node generating or sharing a message in the network:

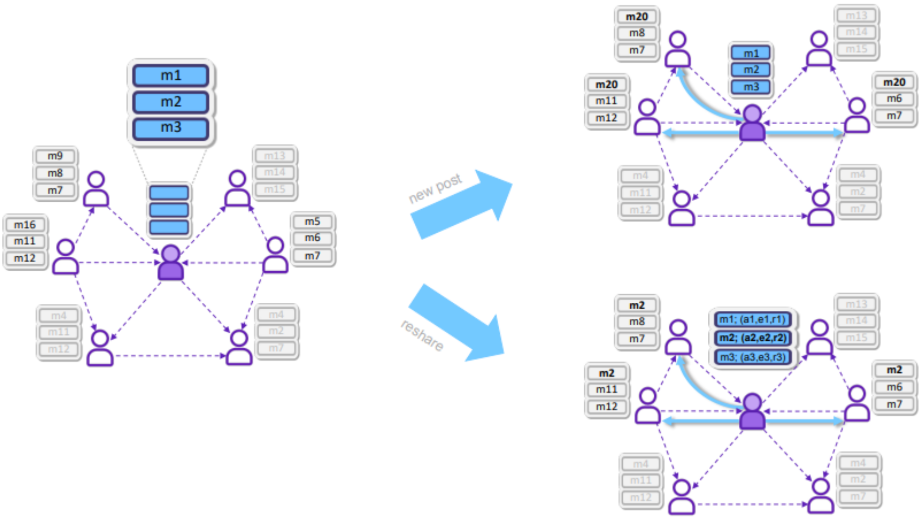


Figure 6: Each agent has a timeline with a limited amount of messages displayed, dotted arrows are friendship connections. Solid links show a sent message. Coloured nodes are the currently active node generating or re-sharing a message for the current time step. Graphic from Bao Tran et al. 'Quantifying the Vulnerabilities of the Online Public Square to Adversarial Manipulation Tactics'[29].

4.4 Topic bias system

The topic bias feature in the network simulation is implemented by assigning topics to messages and nodes, then adjusting the likelihood of message creation, sharing, or engagement based on the alignment between a node's topics and the message's topic. This involves weighting topics differently to reflect their popularity/relevance to the simulated community, influencing how likely nodes are to engage with or spread messages related to those topics.

4.4.1 Topic distribution

Extending the original SimSoM model, a topic layer has been added to capture finer grade engagement. The topic bias distribution is loosely based on the BiasWatch algorithm by Haokai Lu, PictureJames Caverlee and PictureWei Niu in their paper "A Lightweight System for Discovering and Tracking Topic-Sensitive Opinion Bias in Social Media" [20].

- Let $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$ represent the social network, where $\boldsymbol{V}$ is the set of nodes (users), and $\boldsymbol{E}$ is the set of edges (connections between users).

- Let $\boldsymbol{T} = \{\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_k\}$ be the set of topics, with each topic $\boldsymbol{t}_i$ having an associated weight $\boldsymbol{w}_i$ that represents its popularity. The weights satisfy the condition $\sum_{i=1}^{k} \boldsymbol{w}_i = \boldsymbol{1}$ to form a probability distribution.

- For each node $\boldsymbol{n} \in \boldsymbol{V}$, the process of assigning a topic is modeled as a random selection based on the topics in the topics list:

$$\boldsymbol{P}(\text{node } n \text{ is assigned topic } t_i) = \boldsymbol{w}_r$$

- This means the probability that a node $\boldsymbol{n}$ is assigned topic $\boldsymbol{t}_i$ is not equal to the weight $\boldsymbol{w}_i$, ensuring that the distribution of topics among nodes reflects a randomly populated network with diverse interests, therefore w_r is a random topic. This acts as a stochastic element beyond simply using the weight w_i, providing a diverse topic distribution across the network.

4.4.2 Topic similarity

Once the collection of topics have been distributed amongst nodes within the network, messages are assigned a topic based on the probably distribution. Ensuring messages relate to popular topics within the network.

$$\boldsymbol{P}(\text{message } m \text{ is assigned topic } t_i) = \boldsymbol{w}_i$$

- The similarity between two topic distributions $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$, with a small constant $\boldsymbol{\varepsilon} = \boldsymbol{1e - 10}$ to ensure numerical stability, is calculated as:

$$\boldsymbol{similarity} = \frac{\boldsymbol{d}_1 \cdot \boldsymbol{d}_2}{\|\boldsymbol{d}_1\| \times \|\boldsymbol{d}_2\| + \boldsymbol{\varepsilon}}$$

- where $\boldsymbol{d}_1 \cdot \boldsymbol{d}_2$ is the dot product of the two distributions, $\|\boldsymbol{d}_1\|$ and $\|\boldsymbol{d}_2\|$ are the norms of the distributions, respectively, and $\boldsymbol{\varepsilon}$ is a small constant to avoid division by zero.

- The dot product of the two topic distributions $\boldsymbol{d}_1 \cdot \boldsymbol{d}_2$, measures the extent to which the two distributions align or overlap in terms of topic interests.

- The Norms of the distributions $\|\boldsymbol{d}_1\|$ and $\|\boldsymbol{d}_2\|$, provide a measure of the 'strength' or 'concentration' of topics within each distribution.

The similarity equation is conceptually similar to the 'Signed Information Gain (SIG)' concept from BiasWatch as mentioned earlier [20]. The updated equation has been simplified to directly compare topics, it indicates how similar two messages are in terms of their topic distribution. The original equation derived a signed measure from a 'twitter' hashtag and a class which not relevant to this project.

Topic assignment and distribution algorithm:

Algorithm 1 Assign Topic Distributions

```
procedure ASSIGNTOPICDISTRIBUTIONS($G$: graph)
    for $n$ in $G.nodes()$ do
        $chosen\_topic \leftarrow$ random choice from $TOPICS$ with probabilities $TOPIC\_WEIGHTS$
        $G.nodes[n]['topic'] \leftarrow chosen\_topic$
    end for
end procedure
```

Each node will be assigned a topic randomly chosen from a set of predefined topics, with associate probabilities. The result of this algorithm is that each node in the network has a topic that influences subsequent interactions, such as message creation or sharing.

Algorithm 2 Topic Similarity

```
procedure TOPICSIMILARITY($distribution1, distribution2, \varepsilon = 1e-10$)
    $denom \leftarrow$ norm($distribution1$) $\times$ norm($distribution2$) $+\varepsilon$
    return dot($distribution1$, $distribution2$) / $denom$
end procedure
```

Calculates the similarity between two topic distributions and adds stability for edge cases via a constant. This constant is particularly important in cases where one or both distributions might be very sparse.

Algorithm 3 Reshare to Followers

```
procedure RESHARETOFOLLOWERS($G$, $node$, $message$, $finite\_attention$)
    for $follower$ in $G.successors(node)$ do
        if 'messages' not in $G.nodes[follower]$ then
            $G.nodes[follower]['messages'] \leftarrow []$
        end if
        if $message$ not in $G.nodes[follower]['messages']$ then
            if length($G.nodes[follower]['messages']$) $\geq finite\_attention$ then
                $G.nodes[follower]['messages']$.pop(0)
            end if
            $G.nodes[follower]['messages']$.append($message$)
        end if
    end for
end procedure
```

Re-share a message to all followers of a given node in the network. It ensures that duplicates of the message are avoided, and it adheres to a maximum timeline length by removing older messages.

Algorithm 4 Reshare Based on Topic Similarity

```
procedure ReshareBasedOnTopicSimilarity($G$, $finite\_attention$)
    for $n, data$ in $G.nodes(data = True)$ do
        $node\_topic \leftarrow$ topic of node $n$
        if 'messages' in $data$ then
            for $msg$ in $data['messages']$ do
                if topic of $msg == node\_topic$ then
                    ReshareToFollowers($G, n, msg, finite\_attention$)
                end if
            end for
        end if
    end for
end procedure
```

Re-shares messages to followers if the message topic matches the node's topic. It considers both the topic of the message and the interest of the nodes and helps simulate how information typically spreads in a network where users prefer content that aligns with their interests.

5 Evaluation and tests

The implementation of the base model in this project is successful and recreates the network power distribution, agent propagation and infiltration as well as message sharing and generation. The addition of a topic distribution layer is also successful and captures the case when botnets in a OSN focus on specific topics.

5.1 Network propagation

The network of interconnected nodes follows a power law distribution as expected through the implementation of the preferential attachment mechanism discussed earlier.

Figure 7: total nodes=25

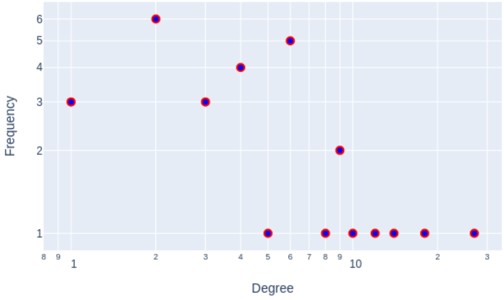


Figure 8: total nodes=70

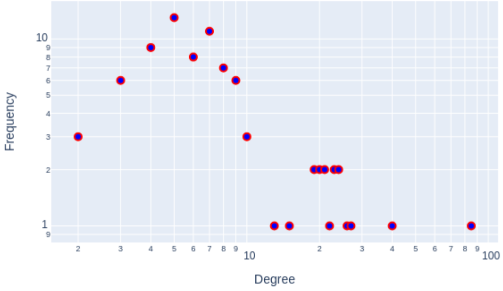


Figure 9: total nodes = 100

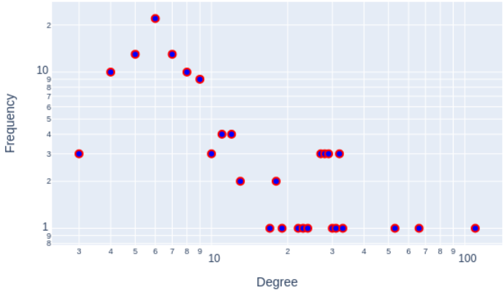
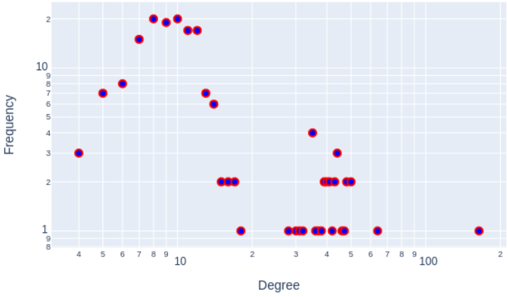


Figure 10: total nodes = 150



5.2 Inauthentic user infiltration

The inauthentic nodes infiltration method is accurately represented by the model as per the infiltration mechanic presented earlier in the implementation section.

Figure 11: n=35, m=1

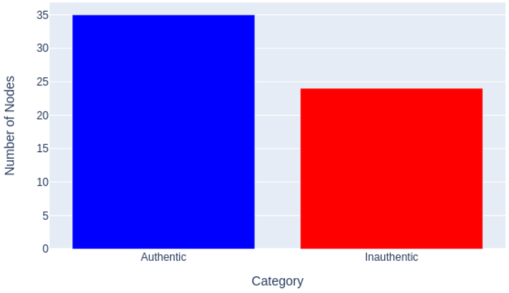


Figure 12: n=35, m=2

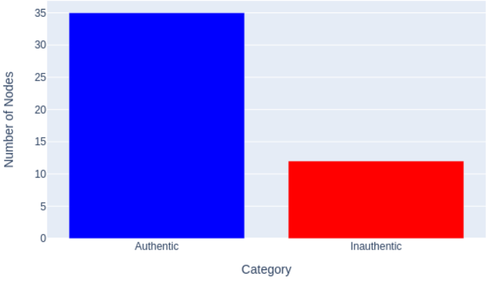


Figure 13: n=35, m=3

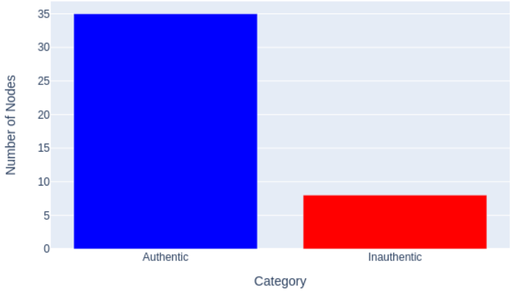
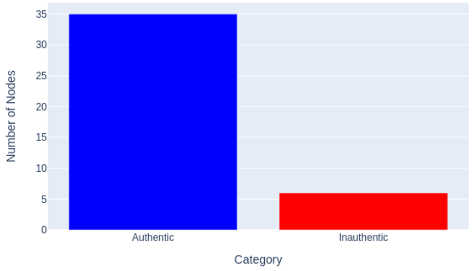


Figure 14: n=35, m=4



5.3 Message propagation

The implemented model confirms the expected behaviour of authentic and inauthentic nodes. The base model produces a series of messages and over time the engagement of content increases as more spam populates the network and overruns the overall message quality of the network.

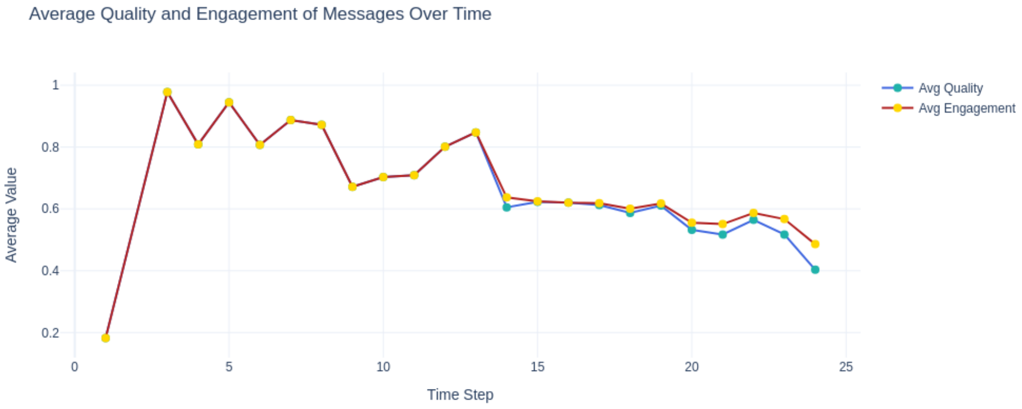


Figure 15: The average quality and engagement of all messages in the network

5.4 Agent propagation

The agent propagation implementation is also successful in the model. The distribution of nodes follow a power law distribution and the ratio of authentic to inauthentic nodes is controlled by beta and the amount of connections by gamma.

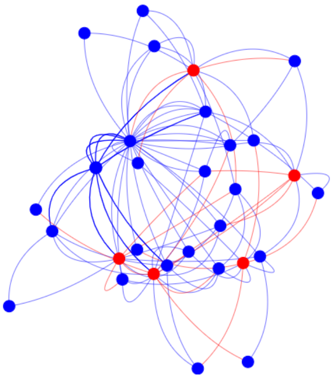


Figure 16: n = 25, beta = 0.7, gamma = 0.25, flood factor = 1

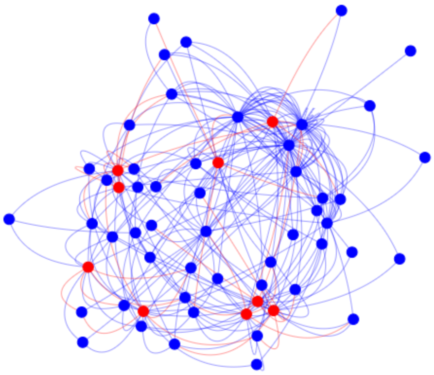


Figure 17: n=50, beta = 0.7, gamma = 0.25, flood factor = 1

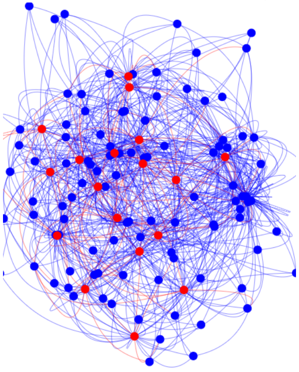


Figure 18: n=100, beta = 0.7, gamma = 0.25, flood factor = 1

5.5 Original base model without topic bias

We can see the base model simulation doesn't produce messages based on any topic distribution. The following scatter plot shows a random distribution of topic focus from the total messages produced in the network.

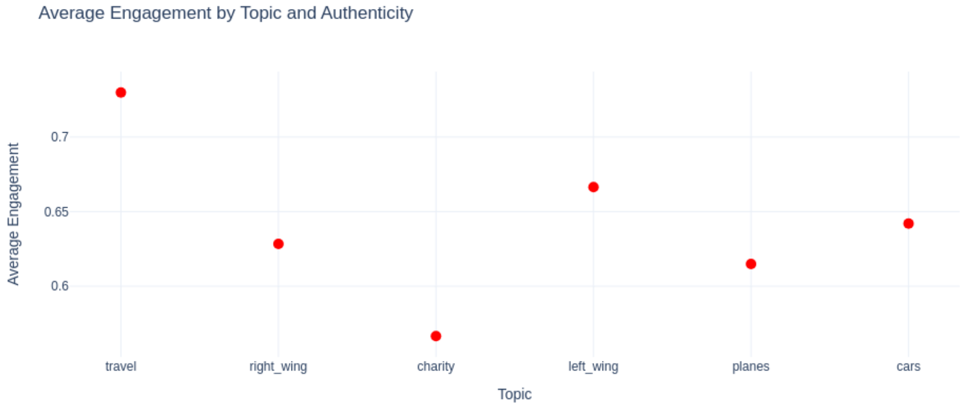


Figure 19: Average engagement by topic with no bias - topic focus is randomly assigned

The 3 presented examples show the models adaptation to the newly introduced topics, creating a focused flooding strategy. The addition of the topic bias creates a more realistic.

5.6 Base model with topic bias - Example 1

In this case for the purpose of modeling, a distribution of example topics have been propagated throughout the network for each node and each newly generated message. The messages are generated and re-shared based on the distribution, topics featured are "tech", "planes", "right wing", "left wing" and "charity".

Topic distribution = {left_wing: **0.4**, right_wing: **0.3**, cars: **0.05**, charity: **0.05**, planes: **0.1**, travel: **0.05**, tech: **0.05**}

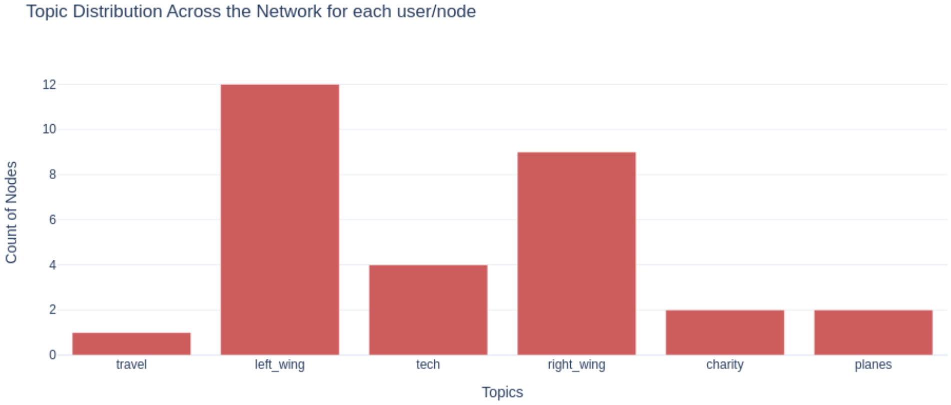


Figure 20: topic distribution for nodes in the network

The added topic bias mechanism generates multiple topics and assigns each one a probability of its appearance in the network for each node. Every node in the network is assigned a topic bias, which was not present previously.

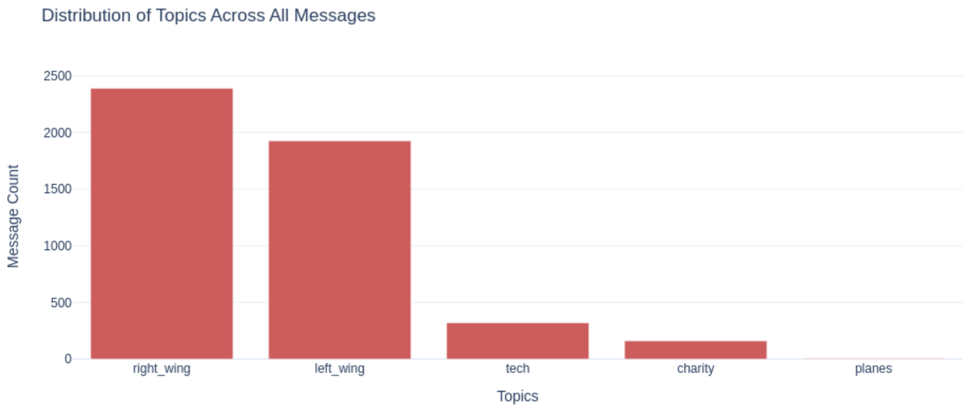


Figure 21: Topic distribution across all messages in the network

Across all messages generated, we can see a topic majority following the defined distribution. In this case the the most popular topics are ’left wing’ and ’right wing’. The least popular topics in this case ’travel’ and ’planes’ aren’t discussed are prominently and do not receive a focus when generating messages.

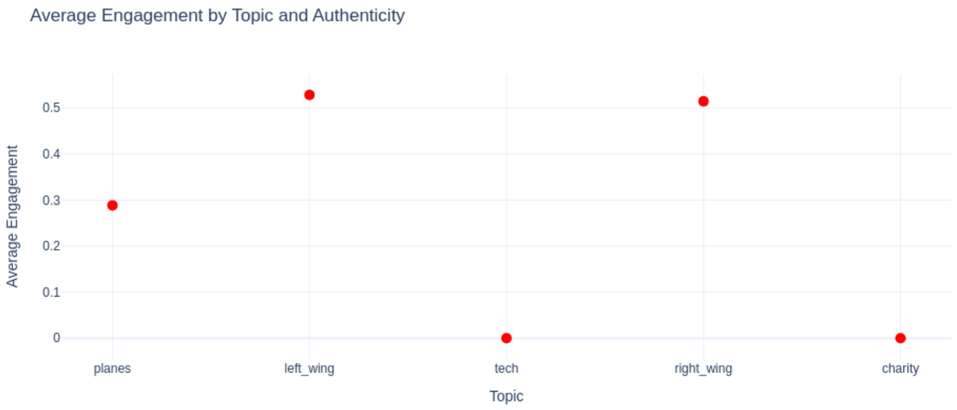


Figure 22: Average engagement by topic - with bias

With this system, the inauthentic nodes are targeting engagement based on certain topics. Nodes assigned the most popular topics are targeted the most - receiving messages with higher engagement after analysing topic similarity.

5.7 Base model with topic bias - Example 2

Another example of inauthentic nodes targeting produced messages towards authentic accounts, this time there’s a larger topic distribution.

Topic distribution = { tv_shows: **0.06**, architecture: **0.08**, jewelry: **0.012**, weather: **0.10**, politics: **0.22**, education: **0.012**, hacking: **0.16**, property: **0.08**, dating: **0.06** }

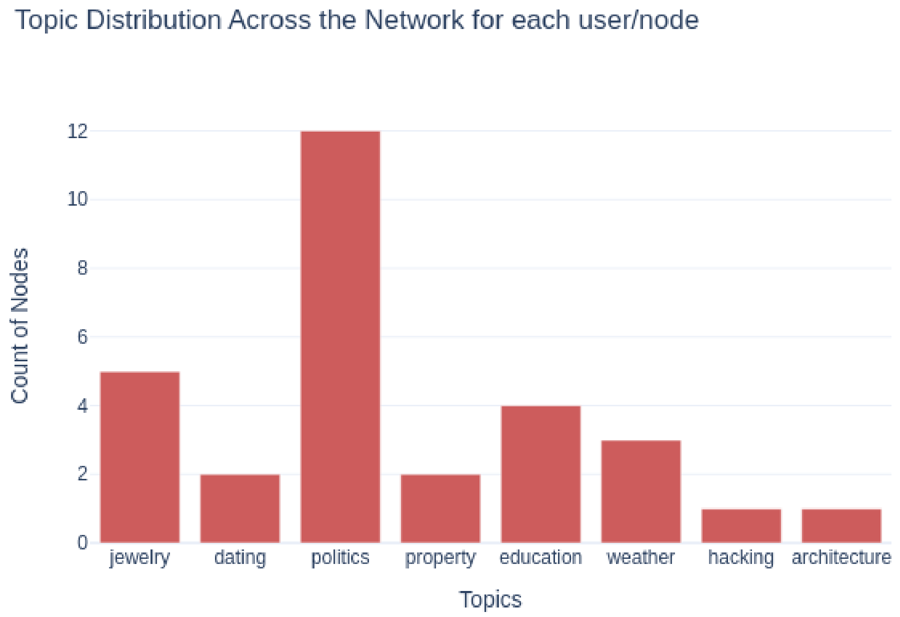


Figure 23: Topic distribution for nodes in the network - example 2

Figure 23 shows all the topics and their distribution across nodes in the network.

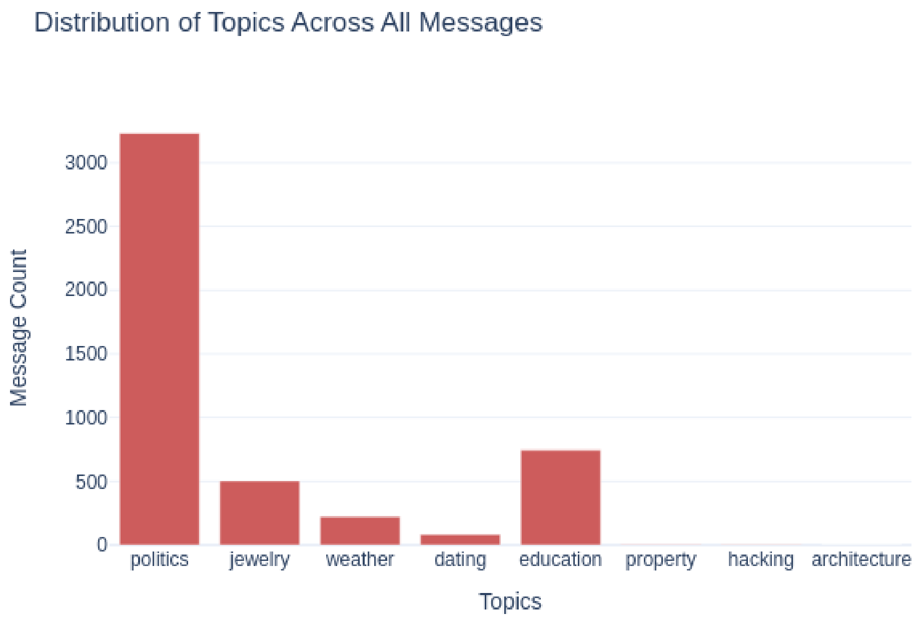


Figure 24: Topic distribution across all messages in the network - example 2

Figure 24 shows messages produced by inauthentic nodes targeting the most popular topics within the network. Due to the probability distribution, the most uncommon topics are dropped entirely - this behaviour is also a by product of simulating a smaller network (less users in the network)

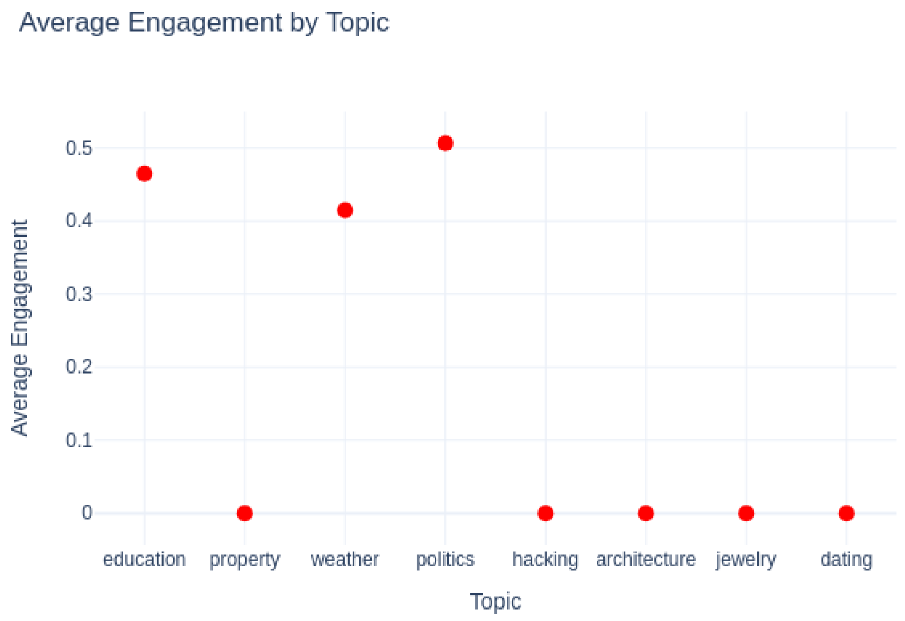


Figure 25: Topic distribution across all messages in the network - example 2

The final figure 25 shows the engagement increase for the most popular topics in accordance to the targeted topics. Inauthentic nodes are drumming up engagement through the flooding of high engagement content by exploiting the user's mainly interest.

5.8 Base model with topic bias - Example 3

Figure 26 shows the topic distribution for each node in the network, this time with more topics. The figure shows a primary interest in jewellery and politics.

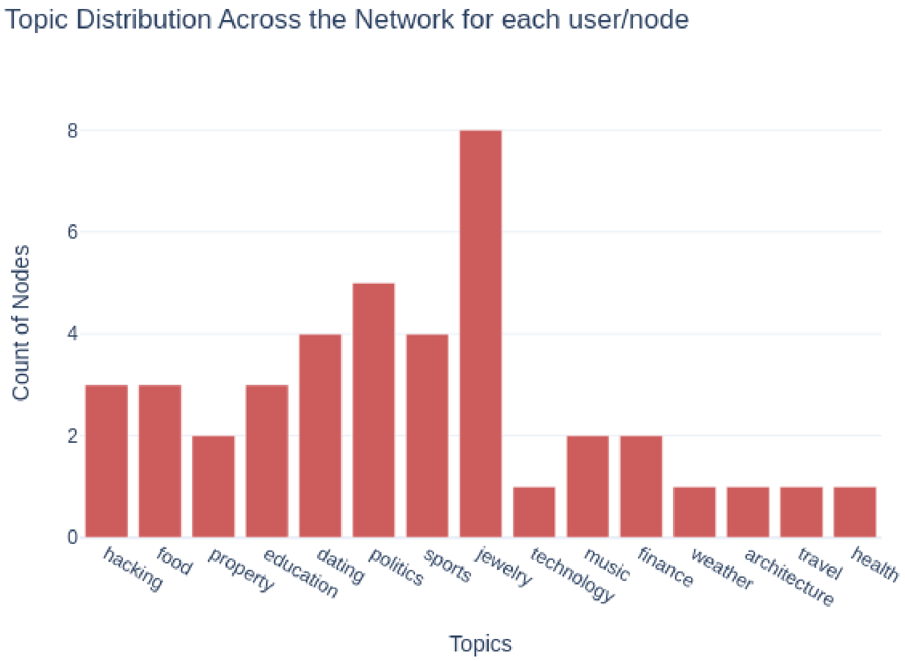


Figure 26: Jewelry, politics, sports and dating are the most popular topics - example 3

The following Figure 27 below shows the topic bias of all messages in the network. Messages are produced with a focus on the most popular topic within the network.

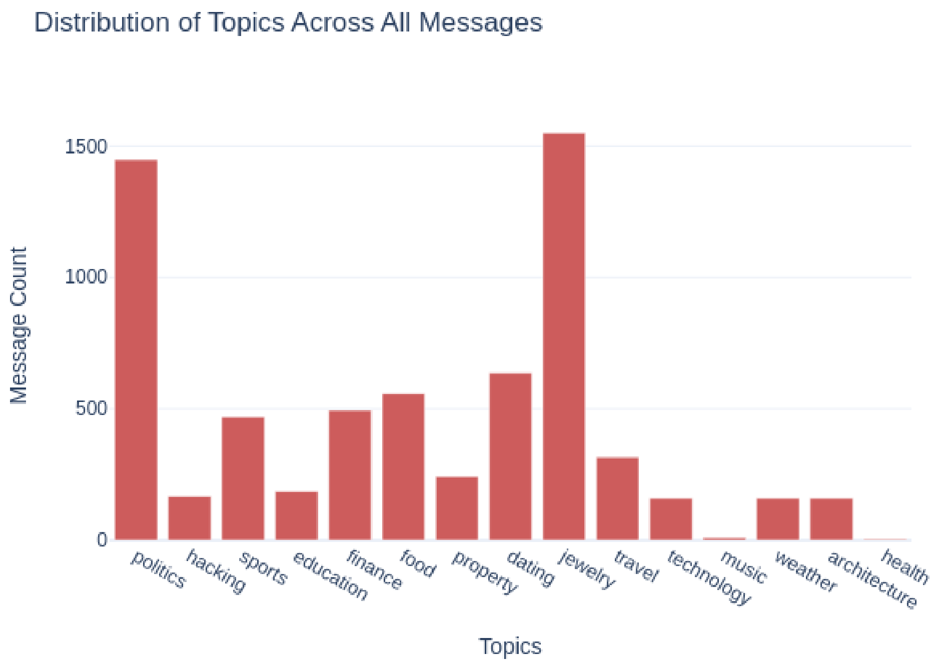


Figure 27: Messages about jewelry and politics are the most popular as a result of being the most popular topic - example 3

Inauthentic nodes have once again produced their messages based on exploiting the most popular interests in the network, in this case jewelry and politics.

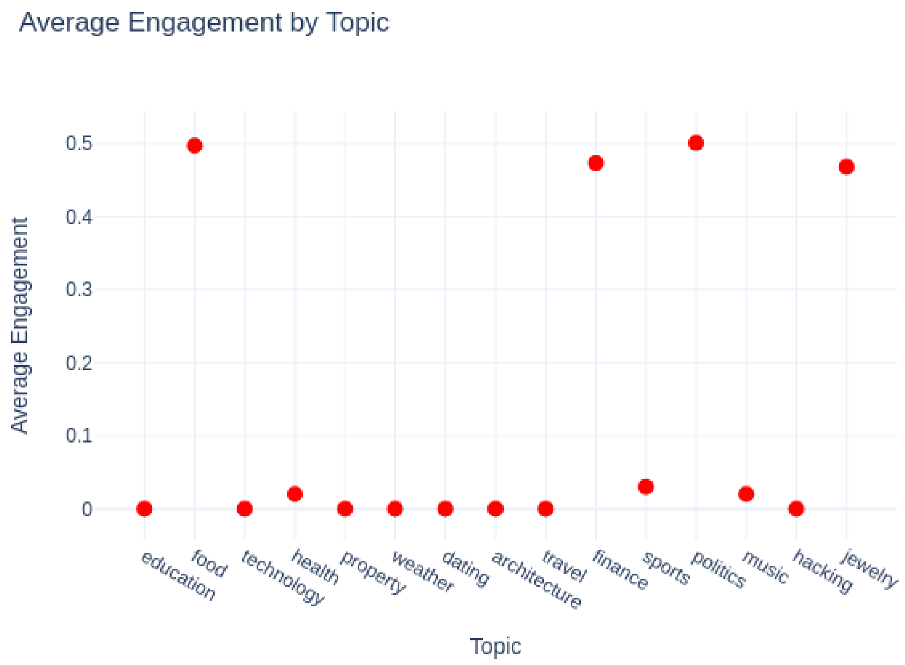


Figure 28: Topic distribution across all messages in the network - example 3

Figure 28 shows increased engagement across the most popular topics, resulting from the targeted messaging campaign.

6 Future work

More work could be done in regards to different aspects of this project. This thesis focuses on the addition of a layer of complexity to an existing base model, in regards to potential areas from improvement the current base model can still be improved, also improvements can be made to the new layer of complexity or engineer another layer of complexity to capture different metrics.

6.1 Improve existing model's metrics

The base model features a good baseline for simulating inauthentic agent infiltration through pooling. Specific behaviours can be simulated to explore different attack vectors and techniques such as discussed by Gan. C et. al [10].

6.2 Improve topic bias layer

The topic bias layer added in this thesis can be expanded upon through a more complex implementation of the user agents but creating different tiers of related personal interests as presented in [34]. This way each agent may capture the behaviour of human interest more accurately.

6.3 Implement new complexity layer

Once again the base model (SimSoM) provides a good basis for message sharing behaviour within the network, but a layer of complexity can be included for a more dynamic approach as currently basic model assumes all nodes will communicate with a distribution of any other connected node. The following paper by Gatti M et al. in 2014[12] discusses in great detail complex methods of simulating user behaviour.

7 Reflection

On reflection the project presented in this final thesis captures all the aspects of its intended implementation, however there areas to be expanded. The topic distribution method can be expanded to include a dynamic allocation of topic subject and topic distribution, allowing the topic trends to change depending on engagement with each time step.

8 Project management

8.1 Implementation schedule report

Task	Difficulty Expected	Actual Difficulty	Reason
Network generation	Medium	Medium	implementing the initial network generation was easier than expected
Agent generation	Large	Medium	Due to available libraries and frameworks, agent generation complexity was reduced
Agent behaviour	Large	Large	The complexity of modeling realistic agent behavior met expectations
Message generation	Medium	Large	Unanticipated challenges arose in generating diverse and realistic messages
Topic bias	Large	Medium	Initial concerns over modeling topic bias were mitigated with refined algorithms
UI	Small	Small	UI development proceeded as planned without significant issues

8.2 Software structure

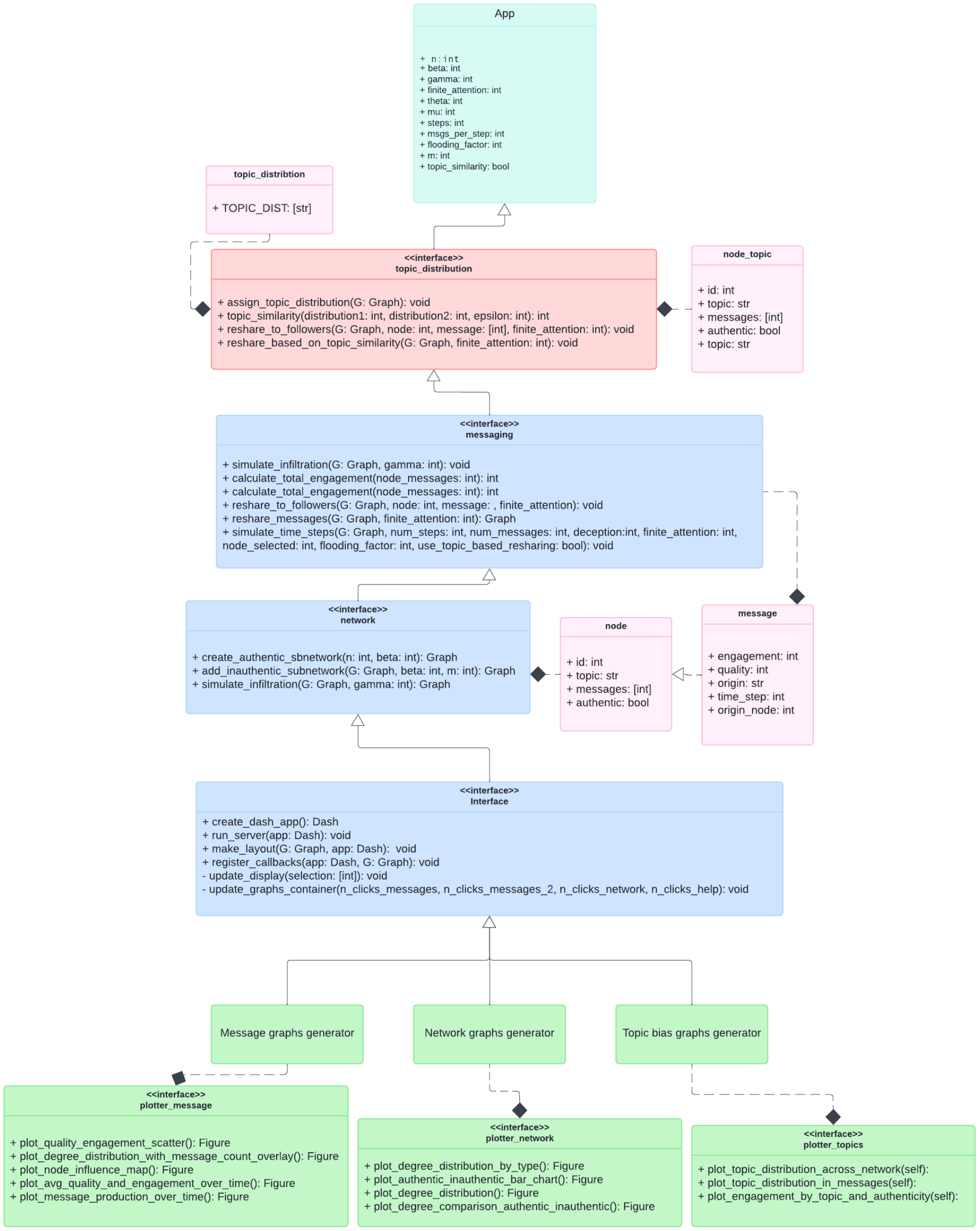

App
+ n:int
+ beta: int
+ gamma: int
+ finite_attention: int
+ theta: int
+ mu: int
+ steps: int
+ msgs_per_step: int
+ flooding_factor: int
+ m: int
+ topic_similarity: bool
topic_distribtion
+ TOPIC_DIST: [str]
<<interface>>
topic_distribution
+ assign_topic_distribution(G: Graph): void
+ topic_similarity(distribution1: int, distribution2: int, epsilon: int): int
+ reshare_to_followers(G: Graph, node: int, message: [int], finite_attention: int): void
+ reshare_based_on_topic_similarity(G: Graph, finite_attention: int): void
node_topic
+ id: int
+ topic: str
+ messages: [int]
+ authentic: bool
+ topic: str
<<interface>>
messaging
+ simulate_infiltration(G: Graph, gamma: int): void
+ calculate_total_engagement(node_messages: int): int
+ calculate_total_engagement(node_messages: int): int
+ reshare_to_followers(G: Graph, node: int, message: , finite_attention): void
+ reshare_messages(G: Graph, finite_attention: int): Graph
+ simulate_time_steps(G: Graph, num_steps: int, num_messages: int, deception:int, finite_attention: int, node_selected: int, flooding_factor: int, use_topic_based_resharing: bool): void
<<interface>>
network
+ create_authentic_sbnetwork(n: int, beta: int): Graph
+ add_inauthentic_subnetwork(G: Graph, beta: int, m: int): Graph
+ simulate_infiltration(G: Graph, gamma: int): Graph
node
+ id: int
+ topic: str
+ messages: [int]
+ authentic: bool
message
+ engagement: int
+ quality: int
+ origin: str
+ time_step: int
+ origin_node: int
<<interface>>
Interface
+ create_dash_app(): Dash
+ run_server(app: Dash): void
+ make_layout(G: Graph, app: Dash): void
+ register_callbacks(app: Dash, G: Graph): void
- update_display(selection: [int]): void
- update_graphs_container(n_clicks_messages, n_clicks_messages_2, n_clicks_network, n_clicks_help): void
Message graphs generator
Network graphs generator
Topic bias graphs generator
<<interface>>
plotter_message
+ plot_quality_engagement_scatter(): Figure
+ plot_degree_distribution_with_message_count_overlay(): Figure
+ plot_node_influence_map(): Figure
+ plot_avg_quality_and_engagement_over_time(): Figure
+ plot_message_production_over_time(): Figure
<<interface>>
plotter_network
+ plot_degree_distribution_by_type(): Figure
+ plot_authentic_inauthentic_bar_chart(): Figure
+ plot_degree_distribution(): Figure
+ plot_degree_comparison_authentic_inauthentic(): Figure
<<interface>>
plotter_topics
+ plot_topic_distribution_across_network(self):
+ plot_topic_distribution_in_messages(self):
+ plot_engagement_by_topic_and_authenticity(self):

Figure 29: UML diagram of application

8.3 Gantt chart

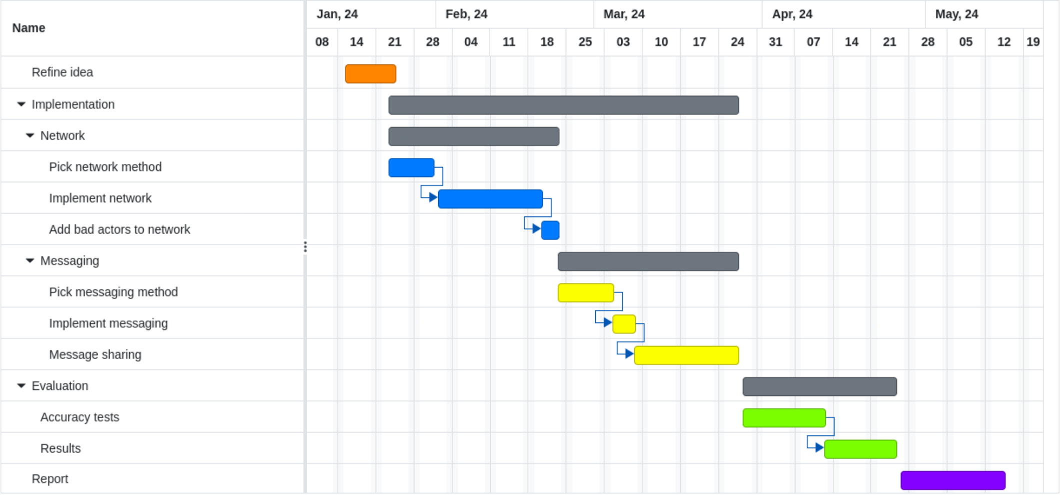


Figure 30: Gantt chart

8.4 risk assessment

Risk	Probability	Severity	Risk exposure	Mitigation
Underestimating task	3	4	12	Set a clear and detailed outline.
Implementation complexity	4	4	16	Take time to understand the design and focus on critical infrastructure.
Poor health	2	5	10	Implement critical sections first to mitigate impact of lost time.
Scope creep	3	3	9	Finish implementations before moving onto other tasks.
Resource availability	2	2	4	Contact ECS for extra machines.
Requirements change	1	5	5	Ensure initial requirements meet the problem criteria.
Data loss	1	6	6	Back up data onto cloud services or physical drives.

9 Bibliography

References

[1] Awrad Mohammed Ali et al. “Synthetic generators for cloning social network data”. In: *Proceedings of SocInfo* (2014).

[2] Sumit Kumar Banshal et al. “Power Laws in altmetrics: An empirical analysis”. In: *Journal of Informetrics* 16.3 (2022), p. 101309. ISSN: 1751-1577. DOI: https://doi.org/10.1016/j.joi.2022.101309. URL: https://www.sciencedirect.com/science/article/pii/S175115772200061X.

[3] Ginestra Bianconi et al. “Triadic closure as a basic generating mechanism of communities in complex networks”. In: *Phys. Rev. E* 90 (4 Oct. 2014), p. 042806. DOI: 10.1103/PhysRevE.90.042806. URL: https://link.aps.org/doi/10.1103/PhysRevE.90.042806.

[4] Axel Bruns. “After the ‘APIcalypse’: social media platforms and their fight against critical scholarly research”. In: *Information, Communication & Society* 22.11 (2019), pp. 1544–1566. DOI: 10.1080/1369118X.2019.1637447. eprint: https://doi.org/10.1080/1369118X.2019.1637447. URL: https://doi.org/10.1080/1369118X.2019.1637447.

[5] Maira A de C Gatti et al. “A simulation-based approach to analyze the information diffusion in Microblogging Online Social Network”. In: *2013 Winter Simulations Conference (WSC)*. 2013, pp. 1685–1696. DOI: 10.1109/WSC.2013.6721550.

[6] Gábor Csányi and Balázs Szendr ői. “Structure of a large social network”. In: *Phys. Rev. E* 69 (3 Mar. 2004), p. 036131. DOI: 10.1103/PhysRevE.69.036131. URL: https://link.aps.org/doi/10.1103/PhysRevE.69.036131.

[7] Brian Dopson, Cardavian Lowery, and Deepti Joshi. “Collection and analysis of social media datasets”. In: *J. Comput. Sci. Coll.* 30.2 (Dec. 2014), pp. 254–261. ISSN: 1937-4771.

[8] Shangbin Feng et al. “TwiBot-20: A Comprehensive Twitter Bot Detection Benchmark”. In: *Proceedings of the 30th ACM International Conference on Information &amp Knowledge Management*. ACM, Oct. 2021. DOI: 10.1145/3459637.3482019. URL: https://doi.org/10.1145%2F3459637.3482019.

[9] Alceu Ferraz Costa et al. “RSC: Mining and Modeling Temporal Activity in Social Media”. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’15. Sydney, NSW, Australia: Association for Computing Machinery, 2015, pp. 269–278. ISBN: 9781450336642. DOI: 10.1145/2783258.2783294. URL: https://doi.org/10.1145/2783258.2783294.

[10] Chenquan Gan et al. “Advanced Persistent Threats and Their Defense Methods in Industrial Internet of Things: A Survey”. In: *Mathematics* 11.14 (2023). ISSN: 2227-7390. DOI: 10.3390/math11143115. URL: https://www.mdpi.com/2227-7390/11/14/3115.

[11] Diego Garlaschelli and Maria I. Loffredo. “Patterns of Link Reciprocity in Directed Networks”. In: *Phys. Rev. Lett.* 93 (26 Dec. 2004), p. 268701. DOI: 10.1103/PhysRevLett.93.268701. URL: https://link.aps.org/doi/10.1103/PhysRevLett.93.268701.

[12] Maíra Gatti et al. “Large-scale multi-agent-based modeling and simulation of microblogging-based online social network”. In: *Multi-Agent-Based Simulation XIV: International Workshop, MABS 2013, Saint Paul, MN, USA, May 6-7, 2013, Revised Selected Papers*. Springer. 2014, pp. 17–33.

[13] Tim Groseclose and Jeffrey Milyo. “A Measure of Media Bias”. In: *The Quarterly Journal of Economics* 120.4 (2005), pp. 1191–1237. ISSN: 00335533, 15314650. URL: http://www.jstor.org/stable/25098770 (visited on 04/25/2024).

[14] Bonan Hou et al. “Modeling and simulation of large-scale social networks using parallel discrete event simulation”. In: *SIMULATION* 89.10 (2013), pp. 1173–1183. DOI: 10.1177/0037549713495752. eprint: https://doi.org/10.1177/0037549713495752. URL: https://doi.org/10.1177/0037549713495752.

[15] Mohsen Jamali, Gholamreza Haffari, and Martin Ester. “Modeling the temporal dynamics of social rating networks using bidirectional effects of social relations and rating patterns”. In: *Proceedings of the 20th International Conference on World Wide Web*. WWW ’11. Hyderabad, India: Association for Computing Machinery, 2011, pp. 527–536. ISBN: 9781450306324. DOI: 10.1145/1963405.1963480. URL: https://doi.org/10.1145/1963405.1963480.

[16] Andreas Birkbak Jessamy Perriam and Andy Freeman. “Digital methods in a post-API environment”. In: *International Journal of Social Research Methodology* 23.3 (2020), pp. 277–290. DOI: 10.1080/13645579.2019.1682840. eprint: https://doi.org/10.1080/13645579.2019.1682840. URL: https://doi.org/10.1080/13645579.2019.1682840.

[17] N Gizem Kocak, Seçil Kaya, and Evrim Erol. “Social media from the perspective of diffusion of innovation approach”. In: *The Macrotheme Review* 2.3 (2013), pp. 22–29.

[18] Preethi Lahoti, Kiran Garimella, and Aristides Gionis. “Joint Non-negative Matrix Factorization for Learning Ideological Leaning on Twitter”. In: *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. WSDM ’18. Marina Del Rey, CA, USA: Association for Computing Machinery, 2018, pp. 351–359. ISBN: 9781450355810. DOI: 10.1145/3159652.3159669. URL: https://doi.org/10.1145/3159652.3159669.

[19] Tobias Lemke and Michael W Habegger. “Foreign Interference and Social Media Networks: A Relational Approach to Studying Contemporary Russian Disinformation”. In: *Journal of Global Security Studies* 7.2 (Apr. 2022), ogac004. ISSN: 2057-3170. DOI: 10.1093/jogss/ogac004. eprint: https://academic.oup.com/jogss/article-pdf/7/2/ogac004/43510245/ogac004.pdf. URL: https://doi.org/10.1093/jogss/ogac004.

[20] Haokai Lu, James Caverlee, and Wei Niu. “BiasWatch: A Lightweight System for Discovering and Tracking Topic-Sensitive Opinion Bias in Social Media”. In: *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. CIKM ’15. Melbourne, Australia: Association for Computing Machinery, 2015, pp. 213–222. ISBN: 9781450337946. DOI: 10.1145/2806416.2806573. URL: https://doi.org/10.1145/2806416.2806573.

[21] Yury A. Malkov and Alexander Ponomarenko. “Growing Homophilic Networks Are Natural Navigable Small Worlds”. In: *PLOS ONE* 11.6 (June 2016), pp. 1–14. DOI: 10.1371/journal.pone.0158162. URL: https://doi.org/10.1371/journal.pone.0158162.

[22] Nikolaos Mavridis. “Artificial agents entering social networks”. In: *A Networked Self.* Routledge, 2010, pp. 299–311.

[23] Monica Murero. “Coordinated inauthentic behavior: An innovative manipulation tactic to amplify COVID-19 anti-vaccine communication outreach via social media”. In: *Frontiers in Sociology* 8 (2023). ISSN: 2297-7775. DOI: 10.3389/fsoc.2023.1141416. URL: https://www.frontiersin.org/articles/10.3389/fsoc.2023.1141416.

[24] David F. Nettleton. “Data mining of social networks represented as graphs”. In: *Computer Science Review* 7 (2013), pp. 1–34. ISSN: 1574-0137. DOI: https://doi.org/10.1016/j.cosrev.2012.12.001. URL: https://www.sciencedirect.com/science/article/pii/S1574013712000445.

[25] Bo Pang and Lillian Lee. 2008. DOI: 10.1561/1500000011.

[26] Andrew T. Stephen and Olivier Toubia. “Explaining the power-law degree distribution in a social commerce network”. In: *Social Networks* 31.4 (2009), pp. 262–270. ISSN: 0378-8733. DOI: https://doi.org/10.1016/j.socnet.2009.07.002. URL: https://www.sciencedirect.com/science/article/pii/S0378873309000367.

[27] Stefan Stieglitz et al. *Do Social Bots Dream of Electric Sheep? A Categorisation of Social Media Bot Accounts*. 2017. arXiv: 1710.04044 [cs.HC].

[28] Chenhao Tan et al. *User-level sentiment analysis incorporating social networks*. 2011. arXiv: 1109.6018 [cs.CL].

[29] Bao Tran Truong et al. *Quantifying the Vulnerabilities of the Online Public Square to Adversarial Manipulation Tactics*. 2023. arXiv: 1907.06130 [cs.CY].

[30] Alex Hai Wang. “Detecting Spam Bots in Online Social Networking Sites: A Machine Learning Approach”. In: *Data and Applications Security and Privacy XXIV*. Ed. by Sara Foresti and Sushil Jajodia. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 335–342. ISBN: 978-3-642-13739-6.

[31] Mudasir Ahmad Wani, Muzafar Ahmad Sofi, and Suheel Yousuf Wani. “Why Fake Profiles: A study of Anomalous users in different categories of Online Social Networks”. In: *International Journal of Engineering, Technology, Science and Research* 4 (2017), pp. 320–329.

[32] Xiaochao Wei, Haobo Gong, and Lin Song. “Product diffusion in dynamic online social networks: A multi-agent simulation based on gravity theory”. In: *Expert Systems with Applications* 213 (2023), p. 119008. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2022.119008. URL: https://www.sciencedirect.com/science/article/pii/S0957417422020267.

[33] Kurt Wirth, Ericka Menchen-Trevino, and Ryan T. Moore. “Bots By Topic: Exploring Differences in Bot Activity by Conversation Topic”. In: *Proceedings of the 10th International Conference on Social Media and Society*. SM-Society ’19. Toronto, ON, Canada: Association for Computing Machinery, 2019, pp. 77–82. ISBN: 9781450366519. DOI: 10.1145/3328529.3328547. URL: https://doi.org/10.1145/3328529.3328547.

[34] Junkai Zhou et al. *Knowledge Boundary and Persona Dynamic Shape A Better Social Media Agent*. 2024. arXiv: 2403.19275 [cs.CL].