Intervention Strategies for Reducing the Spread of Toxicity in Social Networks

A Thesis submitted in partial fulfilment of the requirements

for the award of the Degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE and ENGINEERING (CSE)

Submitted by:

Harsh Bhagat

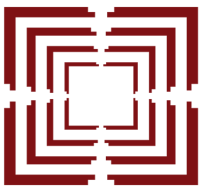
AU2140084

Guided by:

Dr. Amit A Nanavati

Aatman Vaidya

Dr. Seema Nagar



Ahmedabad University

<May 2025>

Declaration

This is to certify that:

1. The thesis titled, "Intervention Strategies for Reducing the Spread of Toxicity in Social Networks" comprises my original work towards the degree of Bachelor of Technology in Computer Science and Engineering (CSE) at Ahmedabad University, and it has not been submitted elsewhere for a degree.

2. Due acknowledgement has been made in the text to all other material used.

Signature of the Candidate

Harsh Bhagat

Ahmedabad University

Certificate

This is to certify that the thesis titled "Intervention Strategies for Reducing the Spread of Toxicity in Social Networks" is the bonafide work carried out by Harsh Bhagat, a student of BTech (CSE) of School of Engineering and Applied Sciences at Ahmedabad University during the academic year 2024-2025, in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology (Computer Science and Engineering) and that the thesis has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

Signature of the Guide
Dr. Amit Nanavati
Professor,
Ahmedabad University

Place: Ahmedabad

Date: 23rd April 2025

Harsh Bhagat

Intervention Strategies for Reducing the Spread of Toxicity in Social Networks

Quick Submit

Quick Submit

Ahmedabad University

Document Details

Submission ID
trn:oid::1:3227472682

Submission Date
Apr 24, 2025, 11:08 PM GMT+5:30

Download Date
Apr 24, 2025, 11:10 PM GMT+5:30

File Name
Final_thesis_report_AU21-40084_Harsh_Bhagat_compressed.pdf

File Size
5.7 MB

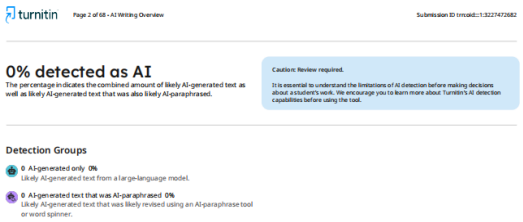
66 Pages

10,137 Words

53,128 Characters

turnitin Page 1 of 68 - Cover Page Submission ID trn:oid::1:3227472682

Figure 0-1: AI plag



turnitin Page 2 of 68 - AI Writing Overview Submission ID trn:oid::1:3227472682

0% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Detection Groups

0 AI-generated only 0%
Likely AI-generated text from a large-language model.

0 AI-generated text that was AI-paraphrased 0%
Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

Figure 0-2: zero AI

Plagiarism Report

Figure 0-3: plag $<5\%$

Abstract

The thesis investigates toxicity propagation in social networks while providing recommendations for platform-based interventions against toxicity. We analysed the movement of toxic conduct among users by extensively studying Twitter, Koo, and Gab social media platform datasets, which were processed to support intervention strategies. Our study demonstrates that peace-enhancing bots introduced through strategic placement work effectively to reduce toxic situations. We employed massive post-preprocessing methods and static/dynamic network analysis to develop time-based user behaviour models through extensive online data inspections. This research establishes key mechanisms of spreading toxicity while proving the effectiveness of intervention approaches for future systems designed to fight toxicity within online communities.

Keywords:

Toxicity Propagation, Social Networks, Hate Speech Mitigation, Network Analysis, User Behavior Modeling

Acknowledgements

It is beyond words to describe how much I appreciate the help of my mentor, Professor Amit Nanavati, who helped me through my tough times and always said, "No worries". I will be forever grateful for his advice and his guidance.

I sincerely thank Dr. Seema Nagar for her important guidance, expertise, and constant help throughout the process. I deeply appreciate the important support she has provided.

I am indebted to the constant support of my senior Aatman Vaidya, whose invaluable insight brought so much to my knowledge and this thesis.

Lastly, I thank my family and friends for their constant support throughout my college journey.

Contents

List of Figures

List of Tables

Acronyms

Acronym	Definition
ADL	Anti-Defamation League
AMP	Amplifier (users who amplify toxicity)
API	Application Programming Interface
ATN	Attenuator (users who reduce toxicity)
BA	Barabási-Albert (scale-free network model)
BERT	Bidirectional Encoder Representations from Transformers
CC	CopyCat (users who mirror toxicity)
CCDH	Center for Countering Digital Hate
CSV	Comma-Separated Values
ESIS	Emotional-Spreader-Ignorant-Stifler model
HC	High/Low Correlation (+/- HC)
HS	High/Low Sensitivity (+/- HS)
IQR	Interquartile Range
JSON	JavaScript Object Notation
LSTM	Long Short-Term Memory (neural network)
NLP	Natural Language Processing
PCC	Pearson Correlation Coefficient
SEIZ	Susceptible-Exposed-Infected-Skeptic model
SIR	Susceptible-Infected-Recovered model
SPA	Spread Activation Model
TLA	Three-Letter Acronym
WCC	Weakly Connected Component

Chapter 1

Introduction

Social media companies, along with their users, now face online toxicity as an urgent matter because it includes hate speech, harassment, and abusive language. New studies show that toxic content exposure adversely affects a person's well-being while reducing public discussion quality and intensifying social divisions before it deteriorates democratic operations [11]. Social media networks maintain toxic behaviour beyond their deployed moderation methods, which include algorithm-based filters and human reviewers, because these strategies are insufficient to address toxic conduct effectively.

Social media platforms now face a significant challenge regarding hate speech proliferation, which requires action from researchers and platform administrators. Hate speech on X (formerly Twitter) rose by 50% after Elon Musk purchased the platform, while homophobic, transphobic, and racist slurs became more prevalent[1] [2] [3]. The Center for Countering Digital Hate (CCDH) together with the Anti-Defamation League (ADL) both confirm that anti-Black slurs increased by 86%. At the same time, anti-LGBT rhetoric exploded following October 2022[4] [5].

[1] https://viterbischool.usc.edu/news/2025/02/a-platform-problem-hate-speech-and-bots-still-thrivin
[2] https://theconversation.com/hate-speech-on-x-surged-for-at-least-8-months-after-elon-musk-takeo
[3] https://www.eurekalert.org/news-releases/1072677
[4] https://counterhate.com/blog/new-report-x-twitter-continues-to-host-posts-reported-by-ccdh-for-
[5] https://fortune.com/2022/11/10/anti-black-racial-slur-twitter-elon-musk-takeover-homophobia-anti

Experts attribute the ongoing increase of hate speech on X to regulatory changes and algorithmic modifications, which unintentionally enhance toxic content [7]. The same patterns of hate speech development appear on the alternative platform Gab, which positions itself as a free speech network and attracts extremist communities, according to Zannettou et al.[27]. Gab has repeatedly been identified by academic research as a platform which facilitates hate speech through both obvious and unregulated forms compared to mainstream social media sites [23] [22].

This research investigates toxicity propagation and intervention efficacy by analysing large-scale cross-platform data supported by modern analytical methods because of the problem's extensive complexity. This research analyses basic toxicity propagation patterns throughout social networks by investigating user behavioural changes and performance assessment of intervention methods. This research draws from existing computational social science and network theory investigations to solve three essential questions.

1. The pathways of toxic behaviour transmission across network links depend on specific network configurations that influence either its rapid propagation or reduction.

2. What determines the stability of user behavioural roles over time, and what elements influence their role changes between amplifier, attenuator, and copycat functions?

3. Which method of placing peace-focused interventions leads to maximum reduction of network toxicity while delivering optimal outcomes?

A comprehensive analysis of three separate platforms named Twitter, Koo, and Gab was conducted to answer those research questions because each platform is distinct regarding user populations, platform architectures, and content regulation. Through analysis of multiple social networking platforms, we discovered patterns of toxicity spread that demonstrate common characteristics alongside distinctive operational patterns for developing customised intervention solutions.

Our work extends previous network evaluations by incorporating dynamic behavioural patterns that track user behaviour changes over time. This research uses natural language processing with toxicity detection and graph-theoretical approaches for network analysis and correlation-based metrics to study behavioural patterns. The combination of various methods delivers in-depth knowledge about network toxicity propagation while demonstrating user behaviour changes caused by evolving information environments.

Our data produced essential benefits for theoretical comprehension and practical application in online harm reduction. Most users (92-95%) duplicate the toxicity levels of their network contacts, but a minority of influential users consistently enhance or minimise toxicity. This research shows that user roles change frequently, and particular patterns indicate when users shift between roles. We empirically prove that peace-enhancing interventions strategically positioned between network users produce effective toxicity reduction when concentrated on users with high correlation coefficients.

Chapter 2

Related Works

This section consolidates scientific developments concerning online toxicity analysis while examining how it spreads and possible intervention solutions on different social media systems.

Automated toxicity detection has experienced substantial advancement in recent years, which has enabled the shift from basic keyword detection to contextual analysis methods. The Perspective API from Google introduced standardised toxicity scoring through machine learning models that received training from human-labelled data [9]. Chakraborty et al. [3] conducted research demonstrating that external triggering events create strong correlations with toxicity spikes based on Perspective API analysis during periods of high political volatility.

2.1 Introduction to Hate Speech on Social Media Platforms

Social media platforms now face the urgent social problem of growing hate speech content, which produces substantial societal effects. Different approaches exist for researchers to define hate speech. The authors Davidson et al. established hate speech as "language that is used to express hatred towards a targeted group or is intended to be derogatory, to humiliate, or to insult the members of the group" [5].

Similarly, Fortuna and Nunes characterized it as "language that attacks or diminishes, that incites violence or hate against groups, based on specific characteristics such as physical appearance, religion, descent, national or ethnic origin, sexual orientation, gender identity or other, and it can occur with different linguistic styles, even in subtle forms or when humor is used" [6].

Twitter serves as a popular platform for analysing hate speech patterns because of its public accessibility through its concise microblogging format. Through its API interface, Twitter provides researchers access to conduct studies about hate speech detection and propagation according to Schmidt and Wiegand [21]. Hate speech in the online space has been thoroughly documented to produce real-life effects. The research of Müller and Schwarz demonstrated that Facebook anti-refugee sentiment directly led to physical attacks against refugees throughout Germany [17]. The Observer Research Foundation identified a direct link between hateful speech online and physical violence in Indian society by tracing cases of mob aggression which started from inflammatory internet content [15].

2.2 Detection and Quantification of Hate Speech

2.2.1 Evolution of Detection Methods

The methods used to identify and measure hate speech have become much more advanced in recent years. In the early stages, researchers depended on pre-defined lexical databases for detecting problematic content [26]. The methods proved ineffective at understanding hate speech context because they produced numerous incorrect detection results.

Machine learning approaches evolved into more advanced systems after the initial methods. The detection of hate speech shows improved performance when deep learning architectures use LSTM networks, as demonstrated by Badjatiya et al. in their research [1]. Mozafari et al. obtained state-of-the-art hate speech identification

results through their work on fine-tuned BERT models according to their research [16].

2.2.2 Perspective API and Toxicity Scoring

The Perspective API, developed by Google, measures text toxicity on a continuous scale ranging from 0 to 1 instead of performing binary classifications [9]. The technique recognises toxicity as a continuous phenomenon; therefore, it supports advanced analytical methods. Perspective's main attribute is TOXICITY, defined as "a rude, disrespectful, or unreasonable comment that is likely to make you leave a discussion" [1].

The API operates with machine learning models that human reviewers labelled through training on millions of comments, thereby establishing itself as a leading tool for toxicity detection in its field. The Perspective API provides additional specialised detection models for threats, insults and identity-based attacks, which expand the capability to analyse complex harmful content [8].

The research by Sap et al. evaluates automated tools while identifying racial and dialectal biases that affect toxicity score assessments [20]. The research conducted by Pavlopoulos et al. proved that context-aware toxicity detection systems outperform context-unaware methods by achieving better accuracy levels [18].

2.3 Models of Information Spread on Social Networks

2.3.1 Epidemic-Based Models and Limitations

Researchers have explored multiple methods to model information spread, which includes hateful content within social networks. Various information diffusion investigations employ epidemic models starting with the Susceptible-Infected-Recovered (SIR) model and its modified versions [12]. The models divide users into distinct

[1] https://developers.perspectiveapi.com/s/about-the-api-attributes-and-languages

states, including susceptibility, infection, and recovery, based on specific information, while showing the state transitions between them.

The SEIZ (Susceptible-Exposed-Infected-Skeptic) model designed by Jin et al. addresses rumor spreading on social media by adding the skeptic state to reflect users who decline information [10] [13]. The ESIS (Emotional-Spreader-Ignorant-Stifler) model created by Wang et al. incorporated emotional elements into the previous approach to account for the emotional aspects of information spreading [25].

SIR models face important performance boundaries for hate speech research. Since their binary structure fails to represent realistic toxic content found in actual online environments [14]. Epidemic models deliver an approximate view of hate speech because toxicity runs along a continuous spectrum instead of existing as distinct entities.

2.3.2 Spread Activation Models (SPA) for Hate Speech

The Spread Activation Models (SPA) represent the movement of information or energetic signals through networks because nodes (users) actively transmit messages to their neighbours using specified threshold criteria or probability systems. SPA models adapted for studying hate speech contain numeric representations of hate content or toxic elements that distribute across social relationships. The classical interpretation of these models states that the total "energy" persists and remains conserved [4].

2.3.3 Non-conservational Model for Toxicity spread

The study by Vaidya et al developed a new non-conservational spectrum-based model that sought to better understand Twitter hate speech patterns. The researchers based their study on the discovery that the behaviour of online toxicity varies according to user interactions, which can either strengthen or weaken it. The model established three user groups according to spreading practices: amplifiers disseminated more hateful content than received. In contrast, attenuators spread less hateful content than they received, and copycats mirrored the hateful content they encountered. The

Perspective API provided toxicity scoring for tweets, allowing the model to measure its "hatefulness" content continuously. The model effectively generated results that matched empirical insights regarding synthetic Barabási–Albert (BA) graphs [2] and genuine Twitter retweet networks, according to overall toxicity and user behaviour distributions. Research findings showed that evaluating user actions remains essential for studying and managing hate speech spread because strategies directed at sustainable users or network behaviour reforms show superior results versus traditional moderation methods of content review. [24].

2.3.4 This Work

The research findings of Vaidya et al. serve as an essential base for subsequent work, which we need to create dynamic models that explain temporal processes in social media toxicity spread, which we have covered in further chapters.

Chapter 3

Toxicity Spread in Social Networks

3.1 Dataset and Preprocessing

We already had the Ribeiro/ Twitter dataset [19] prepared as we used it in the base paper [24]; all that was left was to use similar techniques to analyse the Koo and Gab social networks. Both networks work on posts, which can be reposted to form a directed repost network $G = (V, E)$, where each node $u \in V$ represents a user, and each edge $(a, b) \in E$ represents a repost in the network; there will be an edge from a to b if b reposts a.

We had around 21.3 million posts for KOO in CSV format; processing this big file at once is impossible. Hence, we split the file into 285 files of roughly 75K posts each. Post is a database entry containing the id, creatorId, title, createdAt and handle. Posts in Koo span from January 2020 to September 2023, which is how long the Koo lasted.

For GAB [1], we had around 21.2 million posts in .pkl format. We had to convert the pickle files into CSV for convenience, and similar to Koo, we had to break down this large number of posts into smaller, manageable files. Hence, we split the file into 212 files of 100K posts each. The post here is an entry containing the post_create_time, user ID, username, post_body, parent_id, and other metadata posts in Koo spanning from October 2016 to June 2018.

[1] https://gab.com

Counts	Koo	Gab
Rows	16907989	20198430
Columns	7	24
Nodes/Users	300831	74087
Edges/Reposts	2352146	2315174

Table 3.1: Shape of the Koo & Gab network post-processing

Using the contents of each post, we run it through Google's Perspective API, which gives back much information about the text, but we are only interested in the Toxicity and Severe Toxicity fields. Adding the row to the final dataset only when we get a positive response from Perspective causes insufficient data for the text to infer. There are limitations to Perspective API as well; as with any machine learning model, it is not perfect; it can hallucinate and cause errors, too [2]. The data input size it can process is also limited; sometimes, the posts may contain foreign languages or URLS which Perspective does not understand, so we must consider these in the background and understand that even with them, we must try to find the solution we hope to find. For example, if the post only contains Foreign languages, URLS and Hashtags, then the API throws an error, which we ignore cause of our large volume of data; missing a few connections won't cause any effect. Because of this, we have discrepancies in the rows and edges, shown in Table 3.1.

Perspective API has a limited number of requests it can handle for free, so we had to request the API with gaps in time so that we don't run over the free limit. We were running around 25 instances of the API on an ubuntu linux machine, with the help of 25 instances of tmux running the same python script with different API key, so that we can some what circumvent the rate limit, we were able to get both the datasets ready in 2 months and started analyzing and cleaning them after that.

In conclusion, after around 3 months of work, we had the Koo and Gab dataset and graph ready for analysis. While this was going on, we were experimenting with ideas on the Twitter network, which led to interesting discoveries further testified on Koo and Gab as well.

[2] https://developers.perspectiveapi.com/s/about-the-api-limits-and-errors

3.2 Analyzing the Dataset

We had already found from the Twitter dataset that AMP, ATN and CC users exist, with amplifiers sending out more toxicity than they receive and attenuators experiencing a higher influx of toxicity than they generate. Copycats mirror the toxicity they receive. Instead of looking at the spread, we want to look at the neighbourhood influence on the user and vice versa. For this, we looked into the correlation between a user's toxicity and their neighbourhood toxicity.

3.2.1 Dynamic Behavior

Earlier, we thought that the AMP, ATN or CC users would stay that way for the entire simulation, but users keep changing their behaviour as time passes. The labels given to users keep changing; one user may be a copycat in the 1st week and an attenuator by the 40th week. So, we explored the correlation between a user's average toxicity and their neighbourhood's average toxicity temporally (across weeks and months).

To do this, we would need active users in almost all the weeks we are checking for causes of missing data points that would make foggy inferences. The Twitter dataset is around 40 weeks, but we narrowed it down to the last 13 weeks because there is more activity. In the last 13 weeks, there have been 11984 users who have been active throughout, from which we had to filter the users who had neighbours, so the number came down to 7170 users, posting and retweeting content. We started by calculating the standard network parameters for the subgraph of these active users.

We can see here in Figure 3-1 that most users have a low degree, and some have a very high degree, supporting the notion that famous or popular tweets get retweeted often. Next, I discovered the page rank of all the nodes, but the top nodes showed no outliers, so we turned to the Pearson correlation coefficient (PCC). We plotted the distribution of the PCC value of the user's toxicity with its immediate neighbourhood's toxicity for each week (Figure 3-2).

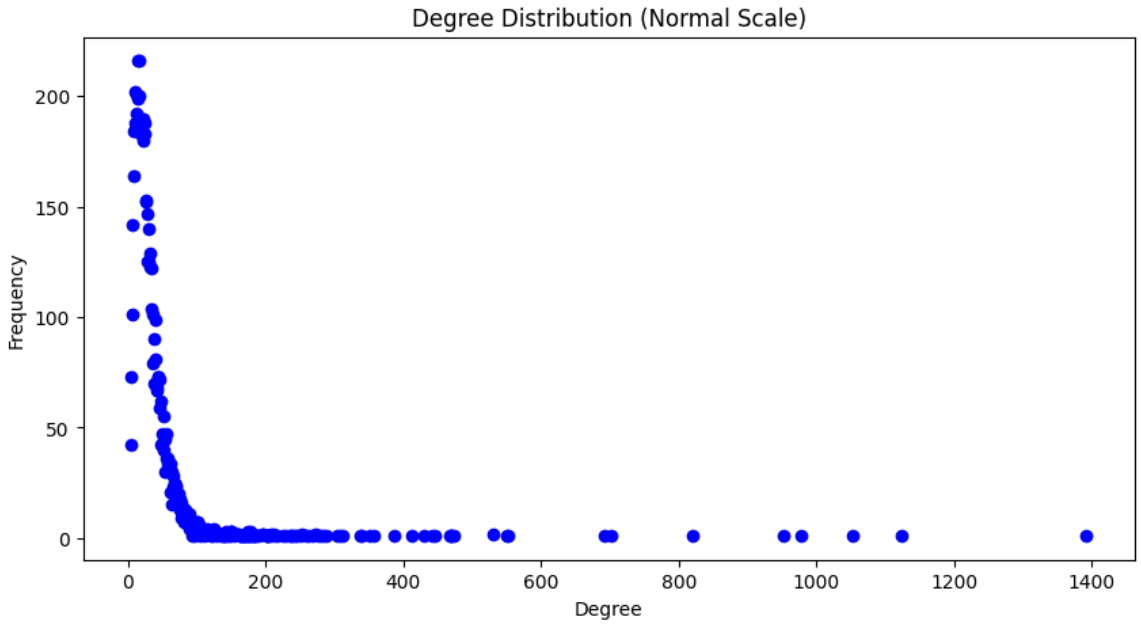


Figure 3-1: Degree Distribution of 7170 subgraphs

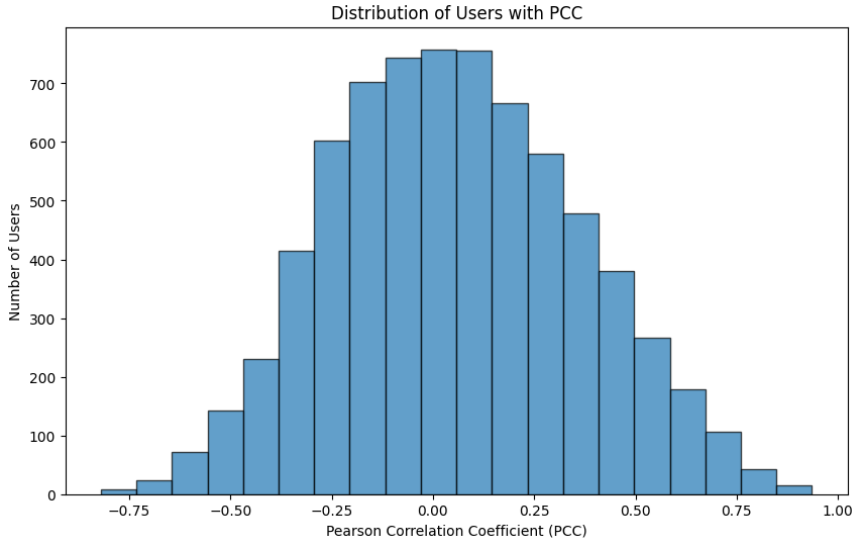


Figure 3-2: Distribution of PCC values of users in the 7170 subgraph

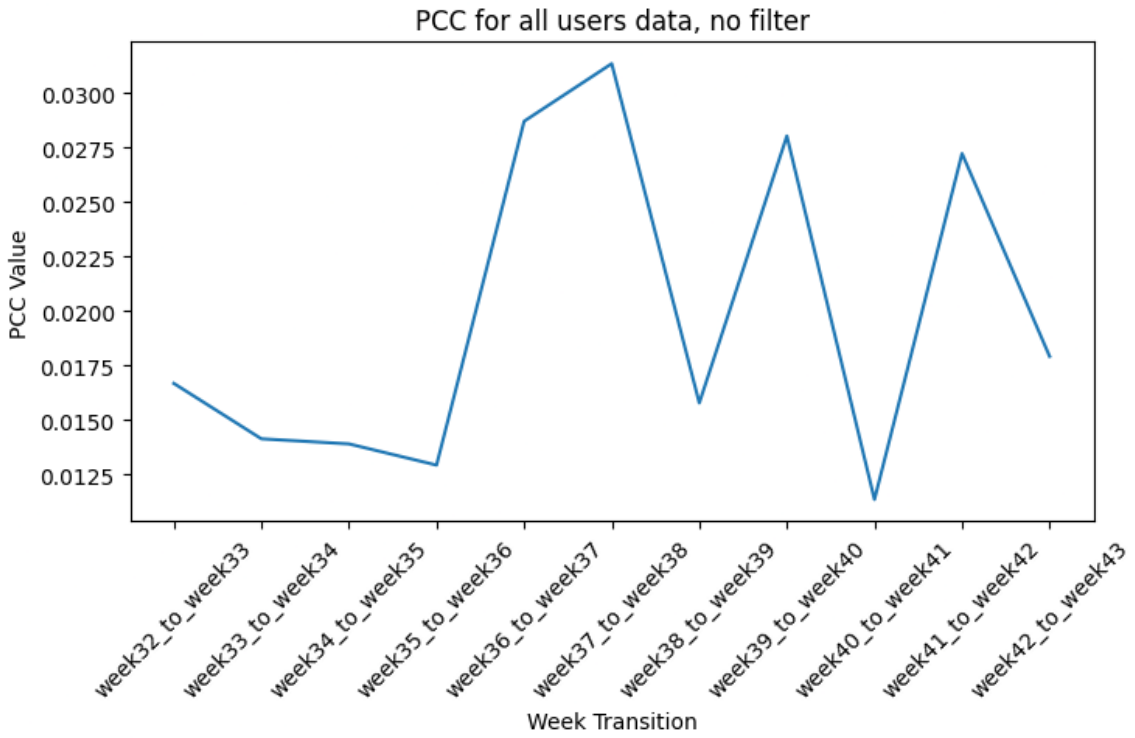


Figure 3-3: change in average PCC values of all users per week

Further, we considered exploring how the change in PCC value occurs for each user and the extent of the change in the user's toxicity as a function of change in its in-degree neighbourhood toxicity (sensitivity) shown in Figure 3-3.

The users were categorised into four groups based on their correlation values:

- $+HC$ **(High Correlation):** PCC ≥ 0.5.
- $-HC$ **(Low Correlation):** PCC ≤ -0.5.
- $+HS$ **(High Sensitivity):** Change in PCC ≥ 0.5.
- $-HS$ **(Low Sensitivity):** Change in PCC ≤ -0.5.

The distribution of these groups across the network was explored to determine whether they were randomly distributed or neighbours. Graph properties (degree distribution, clustering coefficient, modularity, and significance giant components) were calculated for each subgroup, all shown in Tables 3.2, 3.3, 3.4 and 3.5.

Table 3.2: HC Analysis

Pearson Correlation Range	**Number of Users**
≥ 0.5	595
≤ -0.5	189
$-0.25 \leq PCC \leq 0.25$	4071

Table 3.3: HS Analysis

Pearson Correlation Range	**Number of Users**
≥ 0.5	756
≤ -0.5	384
$-0.25 \leq PCC \leq 0.25$	3682

Table 3.4: Graph Properties of Subgraphs (Part 1)

Subgraph	**Nodes**	**Edges**	**Avg Degree**	**Clustering Coeff.**
Overall Filtered	1609	2240	2.78	0.015
+HC	333	362	2.17	0.000
-HC	136	146	2.15	0.000
+HS	498	585	2.35	0.006
-HS	327	364	2.23	0.000

Table 3.5: Graph Properties of Subgraphs (Part 2)

Subgraph	**Assortativity**	**Largest WCC Size**
Overall Filtered	0.047	140
+HC	0.865	4
-HC	0.938	3
+HS	0.745	8
-HS	0.978	4

We also hypothesised that there might be a lag between the users' behaviour compared to their neighbourhood; if a user's neighbourhood suddenly becomes more toxic, how many weeks will it take for the user themselves to change from ATN to AMP or ATN to CC? We can also visualise the user behaviour on the timeline by looking at the user toxicity and neighbourhoods over the last 13 weeks; figures 3-4 and 3-5 show the same for the top HC and HS users. Looking at the plots of the top 5 HC and HS users, we could say that the PCC value does change with the user's behaviour.

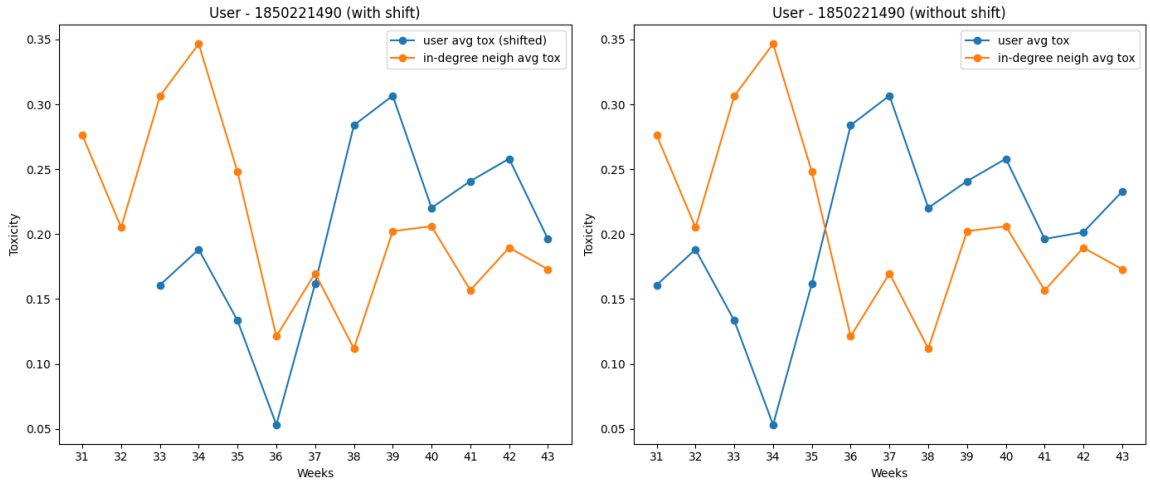


Figure 3-4: top -HC user

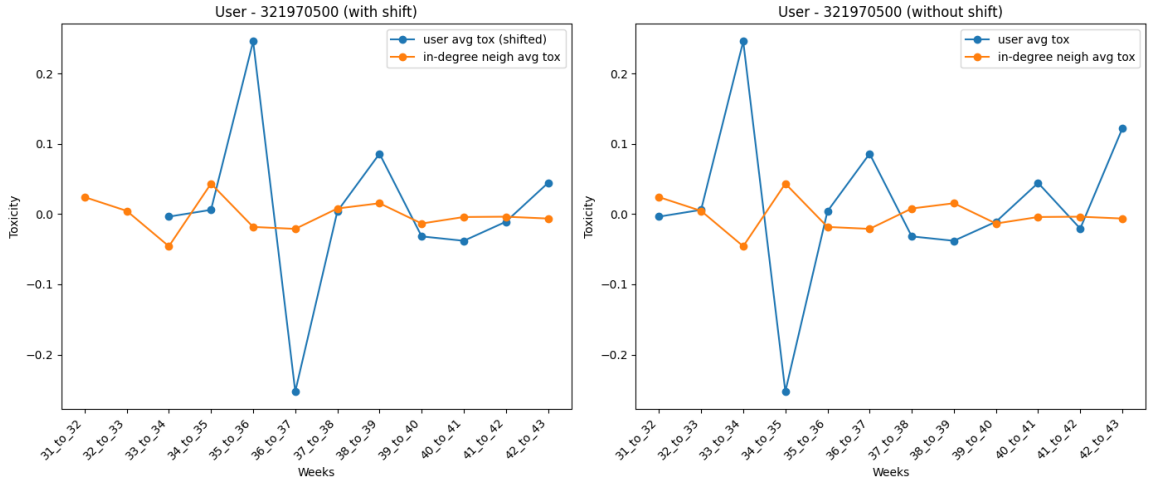


Figure 3-5: top +HS user

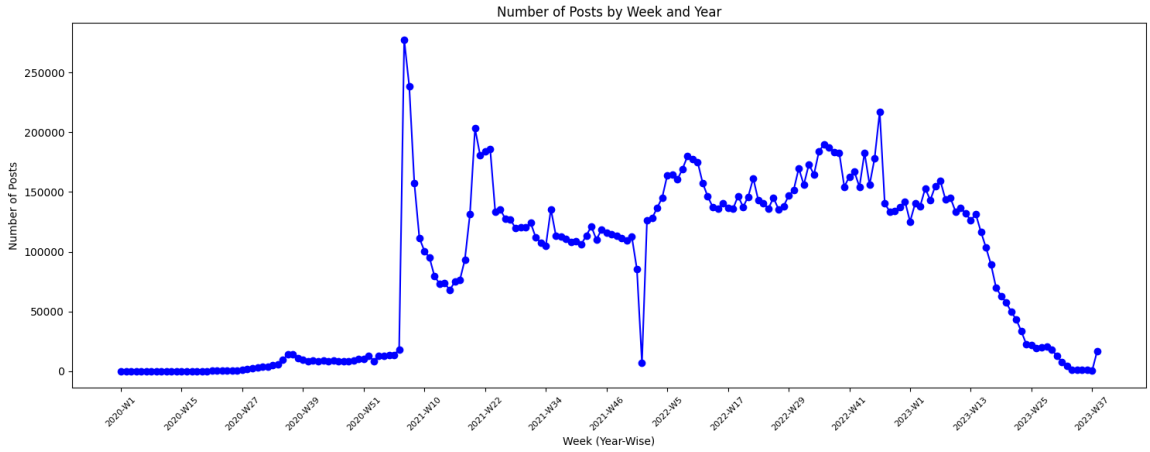


Figure 3-6: Koo Weekly posts distribution

3.2.2 Static Analysis of Koo and Gab

Posts in Koo and Gab are almost spread throughout, as shown in Figures 3-6 and 3-7, unlike Twitter, in which we have a heavy tail because of how it was obtained. Similarly, the toxicity is highly correlated with the number of posts shown in Figures 3-8 and 3-9. Further, we check how the user's toxicity output is related to their neighbourhoods, shown in Figures 3-10 and 3-11; here, the line is just the median line to separate the data into quadrants of user categories, as you can see that much-increased neighbourhood and user activity is going on in Gab compared to Koo and Twitter. We also tried to check for difference (user - neighbours) toxicity to see if it was a normal distribution, but it was not.

Finally, we did the IQR, found the number of CC, AMP, and ATN present in Koo and Gab, and added Twitter for reference, as shown in Table 3.6; as you can see, the toxicity shifts on Gab are highest cause it was made for free speech first than Koo and than twitter.

3.2.3 Dynamic shifts

We wanted to continue exploring the Dynamic Behaviour of users, so we did some plots that would tell us how users change from which category to which category the most. In Figures 3-12, 3-13 and 3-14 respectively. We see that users in Koo and

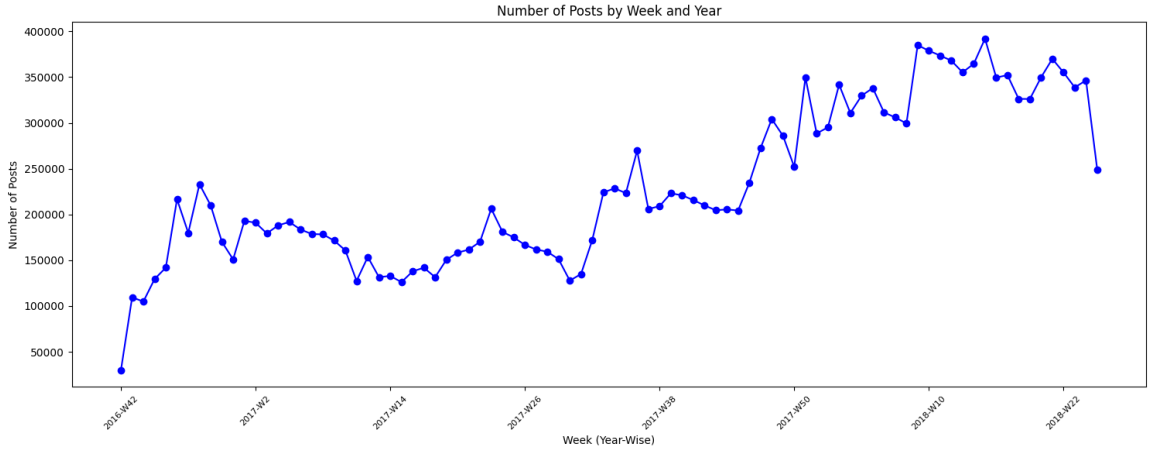


Figure 3-7: Gab Weekly posts distribution

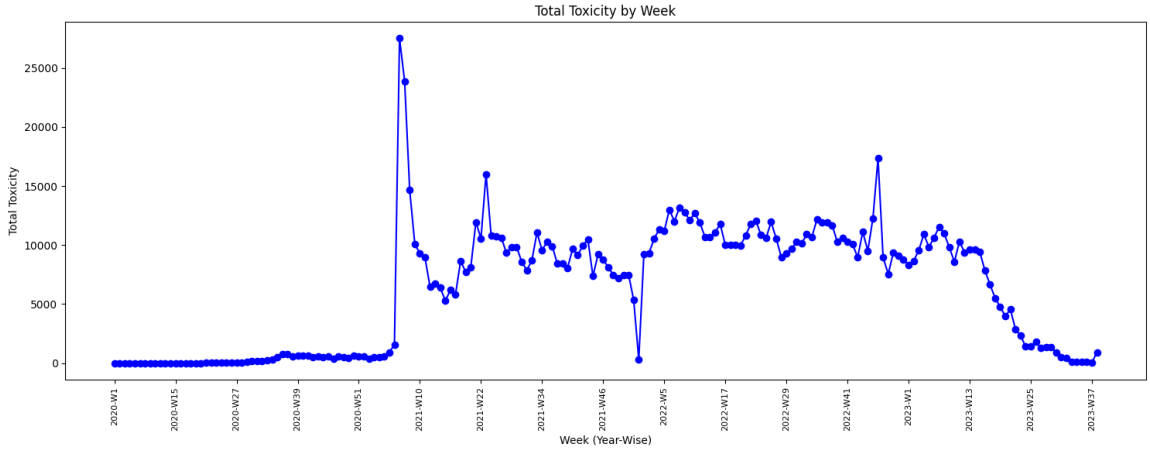


Figure 3-8: Koo Toxicity distribution

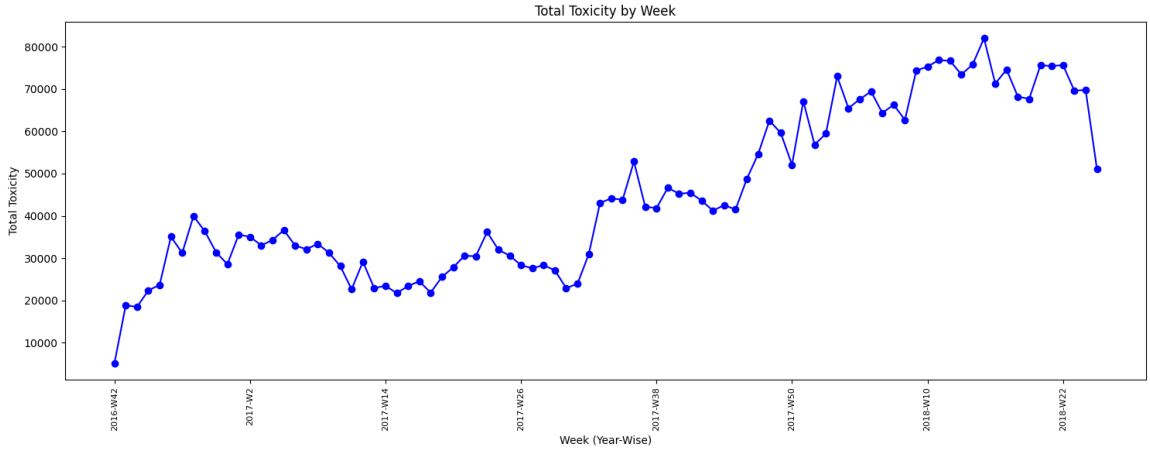


Figure 3-9: Gab Toxicity distribution

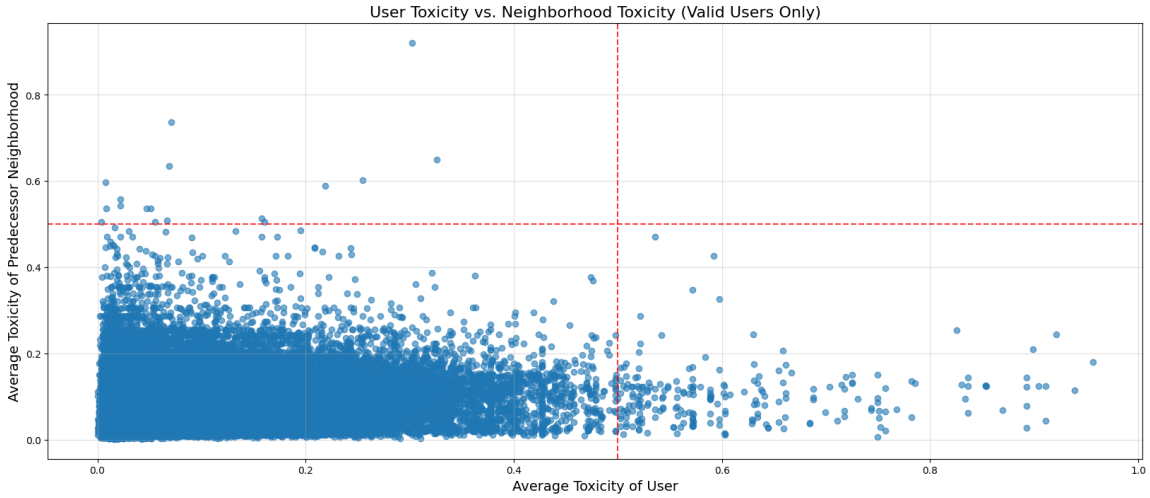


Figure 3-10: Koo user vs neighbourhood toxicity Scatter

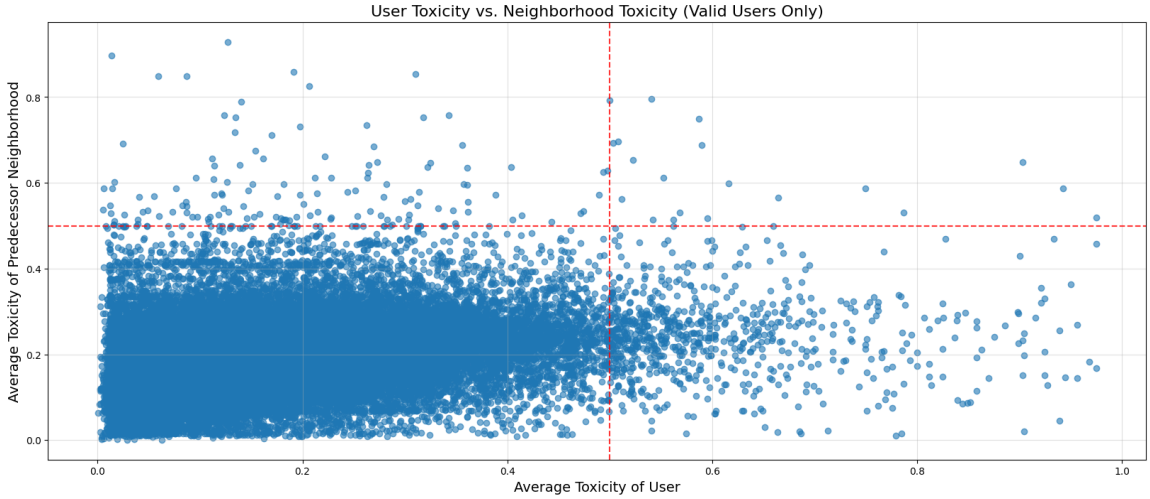


Figure 3-11: Gab user vs neighbourhood toxicity Scatter

Table 3.6: Toxicity Shifts and User Proportions Across Platforms

Category	Platform	Toxicity Shift	User Proportion
Attenuators	Koo	-0.1918	2.05%
	Gab	-0.3587	1.22%
	Twitter	-0.1022	1.39%
Amplifiers	Koo	0.2164	5.45%
	Gab	0.3475	2.84%
	Twitter	0.1133	5.33%
Copycats	Koo	-0.0058	92.49%
	Gab	-0.0226	95.94%
	Twitter	-0.0005	93.28%

Gab are very different from Twitter cause the majority of them switch between being copycats and amplifiers on Twitter, where we see attenuators, too.

We also know now that we do not have all of the active 7170 users remaining in the same category. Almost all of them keep changing their category. Hence, we get a user who is AMP-CC, ATN-CC, pure-ATN, pure-AMP, pure-CC, ATN-AMP or all: AMP-ATN-CC. To compare this among datasets, we needed first to find the active users in Koo and Gab.

For Koo, we found 802 users are present in all the months of the dataset, but that is a minimal number compared to the users found on Twitter and the size of the Koo dataset itself (around 3 lakh users), so we dug a bit more and found an optimal window of 2022-08 to 2023-03, which is 8 months, we have 5645 users present. In this, too, when we remove users with no predecessors, we get 3204 users with whom we can work.

Similarly, for Gab, we found 538 users present in all the months; hence, we found an optimal window of 2017-11 to 2018-06, which is 8 months and has 4808 active users; filtering the non-predecessor users, we get 4622 users.

Breaking down these user categories, we find that Twitter consists primarily of ATN-CC, AMP-CC, and pure-CC users, totalling 6274 out of the 7170 active users. The remaining categories—AMP-ATN and AMP-ATN-CC—are present in very low

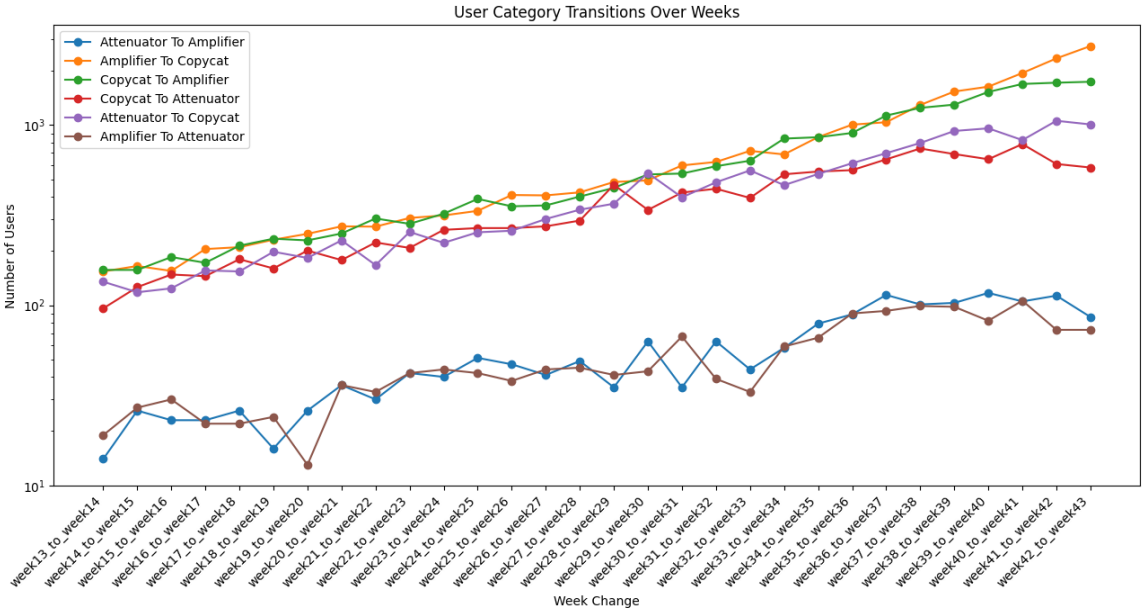


Figure 3-12: Twitter User Category Transitions

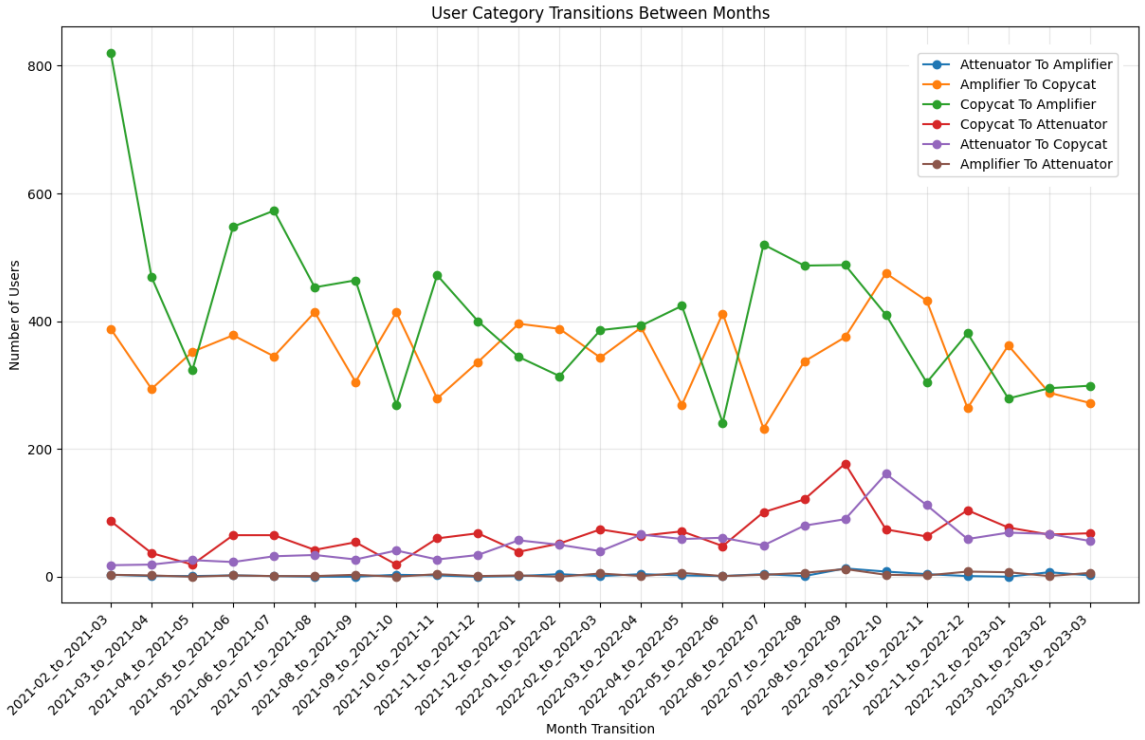


Figure 3-13: Koo User Category Transitions

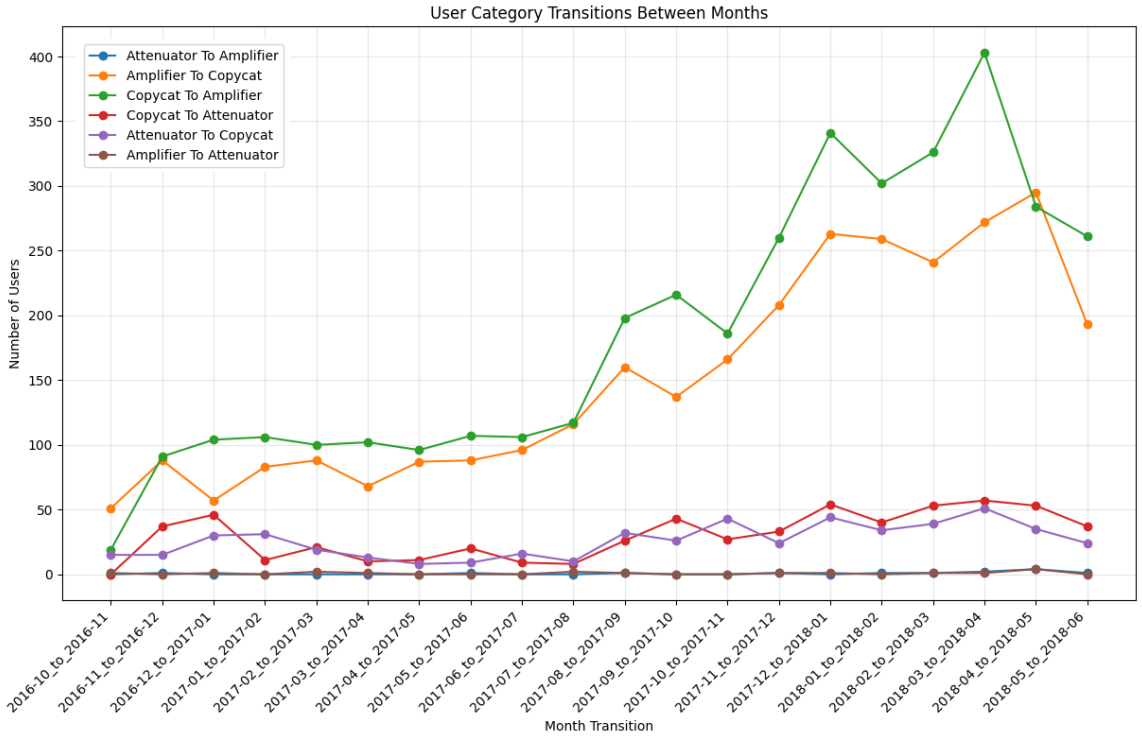


Figure 3-14: Gab User Category Transitions

numbers. In contrast, Gab and Koo exhibit users in all seven categories; however, the counts in many are minimal, so we have chosen to ignore them. The results are presented in Table 3.7. The results also show that this method is correct as the data support the overall picture in Figures 3-12, 3-13 and 3-14, respectively.

Table 3.7: Breakdown of AMP-CC, ATN-CC, and Pure-CC Users (Counts and Percentages)

Category	**Twitter**	**Koo**	**Gab**
AMP-CC	1654 (26.37%)	583 (18.56%)	300 (6.49%)
ATN-CC	817 (13.03%)	113 (3.60%)	27 (0.58%)
Pure-CC	3803 (60.60%)	2443 (77.85%)	4289 (92.84%)

Let's come back to the main subject of dynamic shifts. The previous model also called the static model described in [19], considered the users to have static characteristics, as we now know that is not true, as shown in Table 3.8; hence, we now needed more shifts per category we calculated it for Twitter only as we have the results of the static model in Twitter only.

Table 3.8: Change Probabilities for State Transitions in Twitter

Current State	Transition to Attenuator	Transition to Amplifier
copyCat	0.1737	0.2826
attenuator	0.1874	0.0347
amplifier	0.2903	0.0314

To calculate these dynamic shifts, we need to look at the user's incoming toxicity (neighbours' toxicity) and the shifts they apply afterwards in the real-world data. So we started with plotting Figures of Shifts vs Predecessor toxicity (toxicity difference) for all the datasets and all the standard categories available to them, shown in Figures 3-15 and 3-16. From the figures, we decided that users do not have stable shifts, so taking an average will also not work.

So, we looked at the distribution of these toxicity values to find if there is any standard pattern from which we can sample shifts per category based on the user's input toxicity. Firstly, we tried to use the Python package of a power law [3], and we were able to get the results for each individual user's incoming and outgoing toxicities to see if they follow some standard distribution; the results for Twitter 7170 are in Table 3.9.

Further, we explored the frequency distribution of shift values experienced by all the users, which are shown in Figures 3-15 and 3-16. The Figures are consistent with the user behaviour as AMP see +ve shift values, ATN see -ve shift values and CC are in the middle of it all. To find which distribution shifts belong to we used the Fitter [4] python package designed to take data and see if it fits any known distribution, so we took all the shift values for the different categories and used Fitter to see which top 5 distributions they fit into so that we could sample from them. Still, the results did not follow any pattern, as shown in Figure 3-17.

[3] https://pypi.org/project/powerlaw/
[4] https://fitter.readthedocs.io/en/latest/

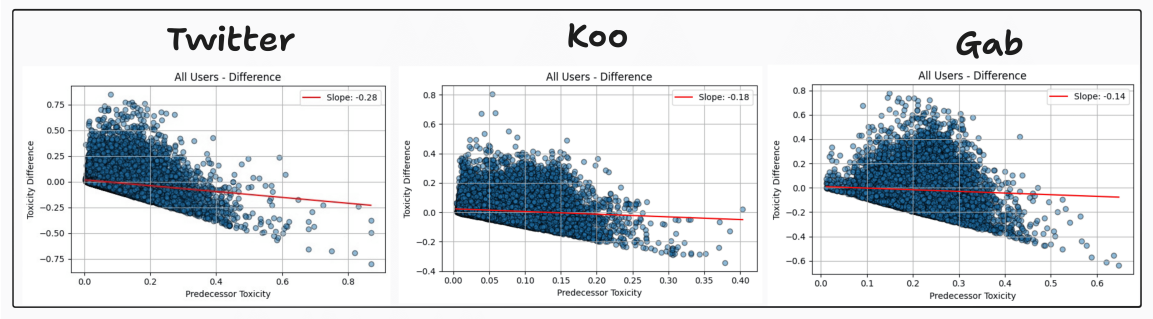


Figure 3-15: Shifts vs Predecessor toxicity all active users in all datasets

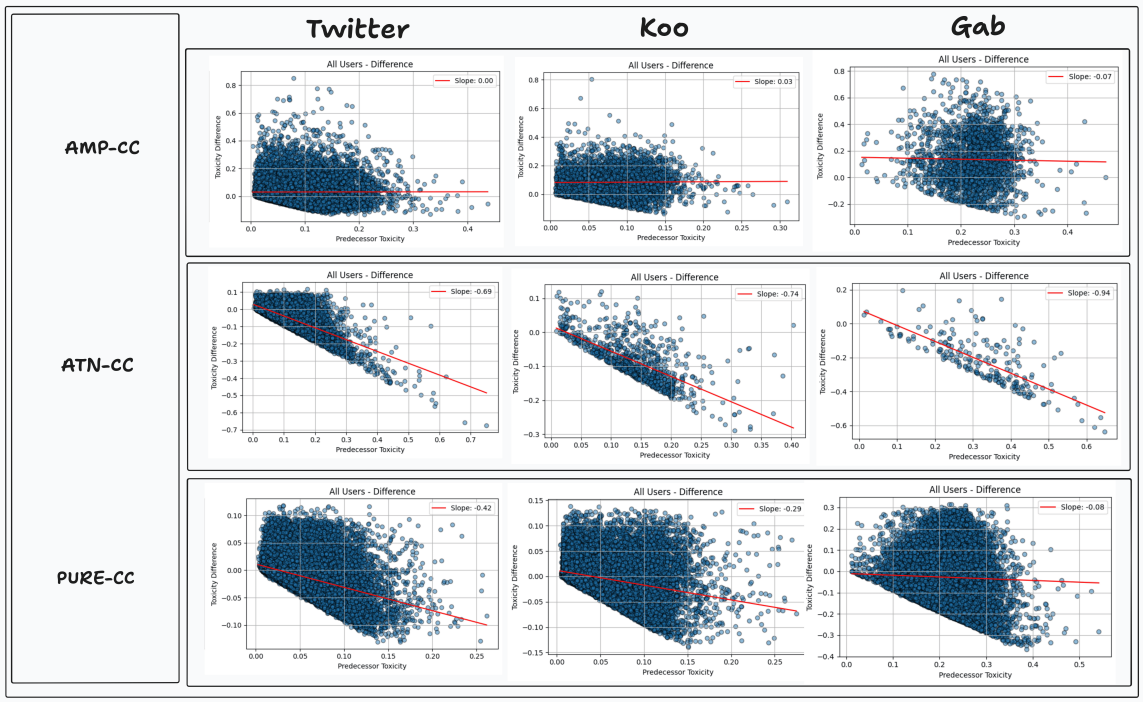


Figure 3-16: Shifts vs Predecessor toxicity per category in all datasets

Type	Exponential	Lognormal	Stretched Exponential	Truncate Power Law	Total
Input tox	1469	340	1575	277	3661
Output tox	2838	397	1934	506	5675
Grand Total	4307	737	3509	783	9336

Table 3.9: Frequency Distribution of Input and Output Toxicities (7170 users) Across Fitted Distributions

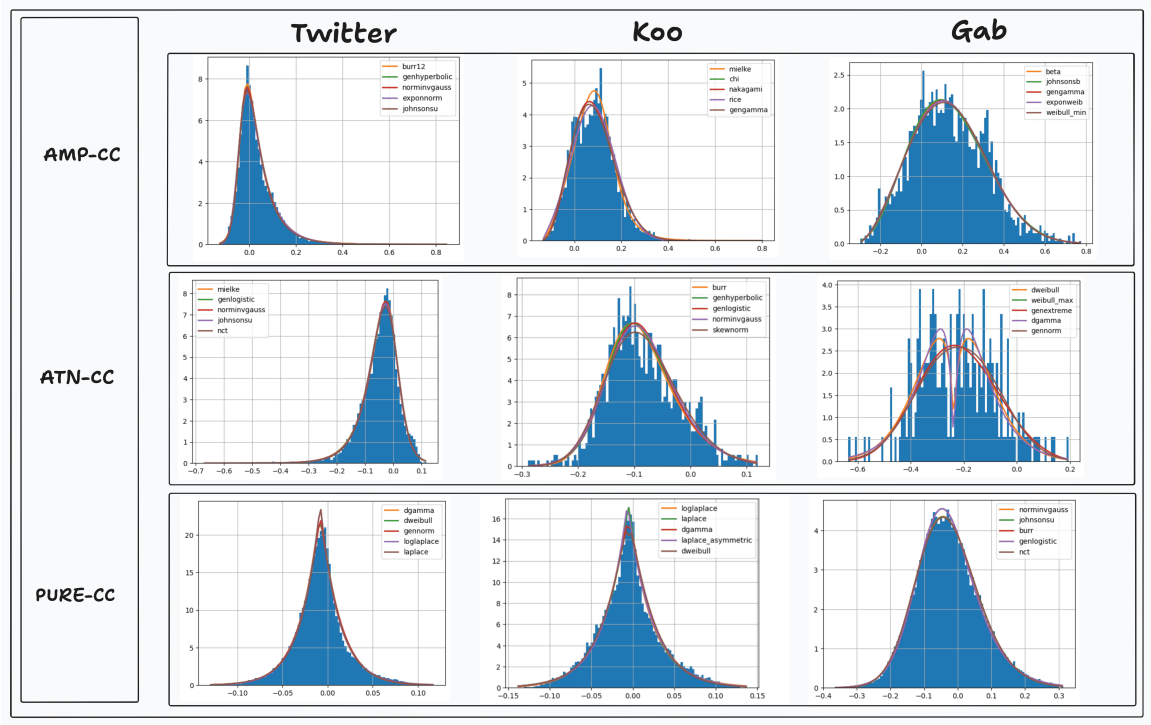


Figure 3-17: Fitter package to see the best fitting distribution per category

3.3 Search for Intervention Among HC Users

We aimed to identify interventions that could introduce peace by decreasing user toxicity. HC users, particularly +HC users, appeared to be the best starting point. Therefore, we conducted simulations on BA [2] graphs. The purpose is simple: HC users are more consistent with their neighbourhood; if we wish to intervene and can only target a few people, converting an HC-AMP to HC-ATN would be very beneficial to decrease the toxicity of the entire network.

We created a Trial Model based on the model presented by Vaidya et al. and modified it to test the hypothesis of HC users given here as Algorithm 1. In this model, we keep the change that HC users have different shift values, given in Table

Category	Toxicity Shift	HC Toxicity Shift
CC	0.000497	0.006154
AMP	0.1022	0.155664
ATN	-0.1133	-0.154797

Table 3.10: Comparison of toxicity shift values, derived from Twitter

3.10. After numerous trials and refinements, we devised the following 12 experiments shown in the Table 3.11.

Algorithm 1 Simulation of Toxicity Spread in a Network

```
Initialise starting nodes
for each timestamp t in total timestamps do
    for each hop in num_hops do
        : Identify active nodes
        for each node in active nodes do
            if node is HC user then
                Use HC toxicity shift values
            else
                Use regular toxicity shift values
            end if
            Compute toxicity shift based on category (CC, AMP, ATTEN)
            Propagate toxicity to successors
        end for
    end for
    Compute average toxicity per user
    Update user behaviour states based on probabilities
end for
```

In Algorithm 1, you can see that we start with a node with a high indegree for better results, as it is more connected. Then we get all the active nodes, which are nodes that have some toxicity to pass on to their successors. We move the toxicity around based on the user's category. The shift will be applied, and with some probability, the user will change their behaviour. Taking all the new points that we got into account, starting with +hc users, as we have kept different shifts for HC users, we are trying to see if it will allow us to get to more consistent users, like having persistent AMP behaviour or CC behaviour, making them the prime target for intervention.

Table 3.11: Overview of Simulation Experiments: seed selection, HC involvement, number of seeds (N), and starting toxicity

Exp	Seed Nodes	+HC?	N	Start Tox.
1	In-degree neighbours of ONE CC	No	1	0.900
2	In-degree neighbours of ONE CC	No	1	0.891
3	In-degree neighbours of ONE CC	No	1	0.855
4	In-degree neighbours of N CC	No	10	0.900
5	In-degree neighbours of N CC	No	10	0.891
6	In-degree neighbours of N CC	No	10	0.855
7	In-degree neighbours of ONE CC	Yes	1	0.900
8	In-degree neighbours of ONE CC	Yes	1	0.891
9	In-degree neighbours of ONE CC	Yes	1	0.855
10	In-degree neighbours of N CC	Yes	10	0.900
11	In-degree neighbours of N CC	Yes	10	0.891
12	In-degree neighbours of N CC	Yes	10	0.855

We ran them not just for CC but also for AMP and ATN. These experiments aim to show that toxicity should increase when HC users are present and remain below otherwise. Figures 3-18 and 3-19 show one of the iterations of these experiments. In this average toxicity per user VS timestamp (just a notion of time for the simulation, hops are just progressions in the simulation), the toxicity increases, peaks, and declines steadily over time. Also, experiments 1-6 are performed with the baseline NON-HC users, and experiments 7 - 12 are performed with HC users. So, we must see that later experiments show higher toxicity than the former.

We could not conclusively say that intervention for HC users would be beneficial; Causing having HC behaviour does not mean you will have higher toxicity or intervening with a peace-bot will reduce it successfully, the Figures do show that toxicity increases but we need to explore further to say for sure.

3.4 Conclusion of the Analysis

We analysed all three datasets available with methods used in [24], tested one model using the novel +hc approach, and explored dynamic shifts and temporal aspects

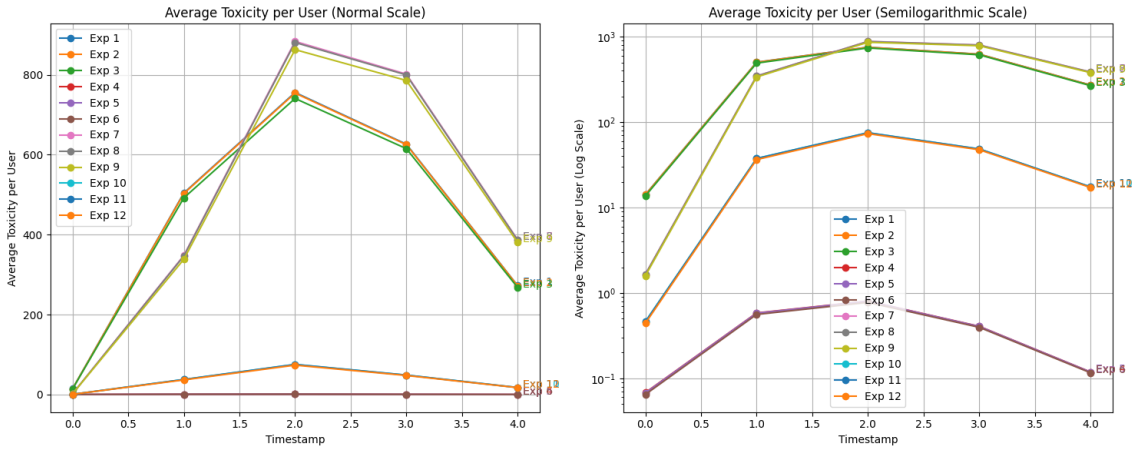


Figure 3-18: Results on 50K node graph

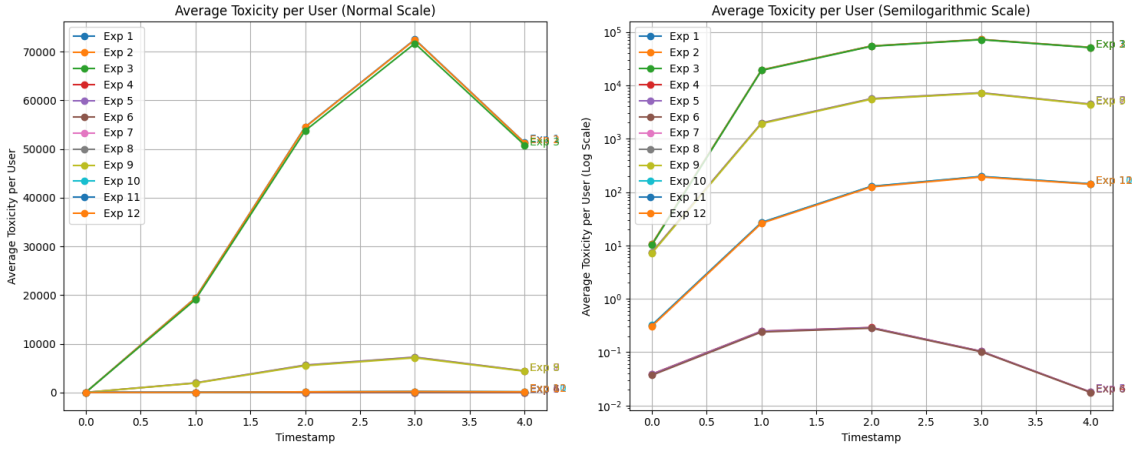


Figure 3-19: Results on 75K node graph

of toxicity spread. In the further chapters, we will assemble all this knowledge and develop a model that improves upon the older methods by incorporating time and dynamic behaviour.

Chapter 4

Dynamic Model for Reducing Toxicity

With all our learnings and setbacks covered in Chapter 3, the trial model in section 3.3, we tried to create a model to explore this temporal aspect of toxicity change and provide a clearer view of what happens in a social network.

4.1 Model setup

In the base paper, the model is static (user behaviour) and does not consider time [24]. We noticed that this is not true, so we created something that is more dynamic (user behaviour) and changes over hops in simulation, that keep repeating over weeks (timestamp in simulation), hence we run the simulation For example, a Random BA graph of 5000 edges, for 8 weeks and three hops, therefore 24 times. Also, the users keep changing their behaviour with a certain probability, calculated from Twitter data only.

To make it easy for the model to run, we preprocessed the user transitions using the probabilities found in Twitter; we created a user_transition_map, now the main logic can easily be executed. The shifts that are applied based on the user behaviour (6 categories), we knew that shifts cannot be sampled cause we did not find a uniform distribution of shifts throughout the datasets per category.

Hence, we came up with the idea to take out all the shifts from Twitter and store the shifts based on the incoming toxicity buckets. Now, in the simulation, we can

check per user what Toxicity they are seeing around them and sample randomly the shift based on the buckets assigned, basically saving the Figure 3-16 in a JSON file. This strategy on paper looked nice, but when we tried to run it, we found that in the simulation, the users' Toxicity is not always seen in the real-world Twitter graph, so we must shift the buckets, cause some buckets are empty. Hence, we also added a logic that samples randomly from the nearest bucket to the one the user sees.

Further, there was another complication: Toxicity, defined by Perspective API, remains within the bounds of 0 to 1. Still, during the simulation, when a user gets Toxicity from their predecessors, they can receive even more than 1, or due to the shift, it can go below zero while passing down, so we cap it at 0 and 1 to keep the integrity of the simulation.

Here is the line-by-line explanation of Algorithm 2

1. We initialize $initial\_nodes$ with the nodes we want to start the simulation with.
2. $values[v] \leftarrow \{start\_tox : 1\}$ means that we start with that inital node v and assign it 1 tweet of toxicity $start\_tox$.
3. $values\_main[v]$ is used to keep track of all the user toxicity and allows us to find the *active* users.
4. Now we start the simulation per *timestamp* and *hop* level, for each *active* node.
5. We first propagate the values of tweets to each successor of that node. Hence, we have spread the Toxicity forward.
6. Now it's time to intervene. We will review all the *active* nodes again and check if they are on the $peace\_bot\_nodes$ list.
7. If yes, we deploy a peace bot with 0.05 toxicity in their neighbourhood (decreasing their average Toxicity). Else, ignore it.
8. Once the peace bot is applied, we can sample each user's shifts and apply them too.
9. Finally, save everything for plotting and further analysis.

Algorithm 2 Toxicity Propagation Simulation

Initialize:
initial_nodes ← Predecessors of *starting_nodes*
values[v] ← {*start_tox* : 1} **if** $v \in$ *initial_nodes* **else** {}
values_main[v] ← *start_tox* **if** $v \in$ *initial_nodes* **else** 0
for week $t = 1$ **to** *timestamp* **do**
    for hop $h = 1$ **to** *num_hops* **do**
        incoming_values[v] ← $\emptyset$ $\forall v \in G$
        for each active node v **do**
            Propagate *values_main*[v] to successors
            Reset *values*[v]
        end for
        for each node $v \in G$ **with** *incoming_values*[v] $\neq \emptyset$ **do**
            avg_incoming ← mean(*incoming_values*[v])
            if $v \in$ *peace_bot_nodes* **then**
                Adjust *avg_incoming* with peace value 0.05
                Record intervention statistics
            end if
            user_category ← *standardized_category_map*[v]
            Find *pred_tox* bucket in *category_shift_buckets*
            if current bucket empty **then**
                Select the nearest non-empty bucket
            end if
            Sample *shift* from bucket's distribution
            new_tox ← $\max(0, \min(1, avg\_incoming + shift))$
            for each successor v_{nbr} **do**
                values[v_{nbr}] ← *new_tox*
            end for
        end for
        for each node $v \in G$ **do**
            values_main[v] ← mean(*values*[v])
        end for
    end for
end for

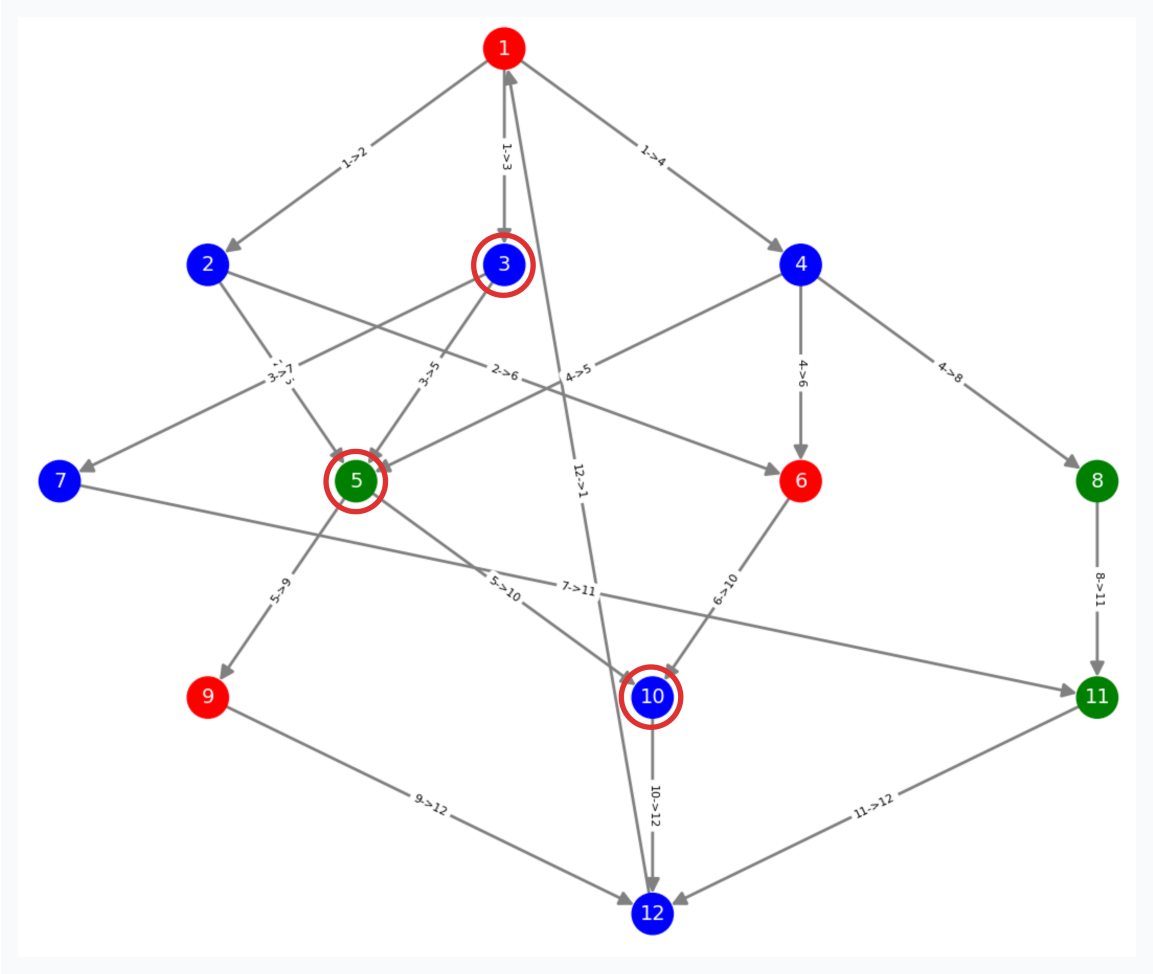


Figure 4-1: Toy graph of 12 nodes, Orange (AMP), Blue (CC) and Green (ATN), Red circle (Peace bots applied)

4.2 Toy graph example

The algorithm runs on our sample graph to demonstrate toxicity spread and peace-bot activity. In this visualisation, we will monitor toxicity value changes across nodes during each timestamp hop level.

4.2.1 Initial Setup

From the toy graph in Figure 4-1, we have:

- 12 nodes with different community designations (Orange: AMP, Blue: CC, Green: ATN)
- Peace bots deployed at nodes 3, 5, and 10 (indicated by red circles)
- Initial toxicity value of 0.9 at node 1
- For simplicity, we have taken the user categories not to change and the shift values as ATN: -0.1, AMP: 0.1, CC: 0.05.

Table 4.1: Toxicity Propagation at Timestamp 0

Node	Initial	After Hop 0	After Hop 1	After Hop 2
1	**0.9**	0	0	0
2	0	0.95	0	0
3	0	0.525*	0	0
4	0	0.95	0	0
5	0	0	0.5187*	0
6	0	0	1.0	0
7	0	0	0.575	0
8	0	0	0.85	0
9	0	0	0	0.6187
10	0	0	0	0.5729*
11	0	0	0	0.6125
12	0	0	0	0

* *Values marked with asterisks indicate nodes where peace bot intervention occurred, reducing toxicity.*

4.2.2 Narrative of Toxicity Propagation

Timestamp 0:

- **Initial State**: Node 1 starts with toxicity value 0.9.
- **Hop 0**:
  - Node 1 propagates its toxicity to nodes 2, 3, and 4.
  - Peace bot at node 3 reduces incoming toxicity from 0.95 to 0.525 ($\frac{(0.9+0.05[peace\_tox])}{2} + 0.05[CC\_tox]$).

Table 4.2: Toxicity Propagation at Timestamp 1

Node	Start of T1	After Hop 0	After Hop 1	After Hop 2
1	0	0	0.7514	0
2	0	0	0	0.8014
3	0	0	0	0.4507*
4	0	0	0	0.8014
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	0	0	0	0
9	0.6187	0	0	0
10	0.5729	0	0	0
11	0.6125	0	0	0
12	0	0.6514	0	0

Table 4.3: Toxicity Propagation at Timestamp 2

Node	Start of T2	After Hop 0	After Hop 1	After Hop 2
1	0.7514	0	0	0
2	0.8014	0	0	0
3	0.4507	0	0	0
4	0.8014	0	0	0
5	0	0.4259*	0	0
6	0	0.9014	0	0
7	0	0.5007	0	0
8	0	0.7014	0	0
9	0	0	0.5259	0
10	0	0	0.5091*	0
11	0	0	0.5011	0
12	0.6514	0	0	0.562

  - Nodes 2 and 4 receive toxicity of 0.95.

- **Hop 1**:

  - Nodes 2, 3, and 4 propagate to successors.

  - Node 5 receives toxicity from multiple nodes; peace bot reduces average to 0.5187 ($\frac{(0.95 \cdot 2 + 0.525 + 0.05)}{4} - 0.1$).

  - Nodes 6, 7, and 8 receive full/partial toxicity.

- **Hop 2**:

- Node 10 has Peace Bot intervention, reducing its value to 0.5729.
- End of timestamp: nodes 9, 10, and 11 receive moderate toxicity.

Timestamp 1:

- **Hop 0**:
  - Nodes 9, 10, and 11 propagate to node 12, raising its toxicity to 0.6514.
- **Hop 1**:
  - Node 12 feeds back into node 1, increasing its toxicity to 0.7514.
- **Hop 2**:
  - Node 1 spreads to 2, 3, and 4 again.
  - Peace Bot reduces node 3's toxicity to 0.4507.
  - Nodes 2 and 4 receive 0.8014 each.

Timestamp 2:

- Node 5 is intervened by a Peace Bot, resulting in 0.4259.
- Nodes 6–8 pick up propagated toxicity from nodes 2–4.
- Node 10 again sees intervention (now to 0.5091).
- By the end, toxicity loops back to node 12, reaching 0.562.

Node	Without Peace Bots	With Peace Bots
3	0.95 → 0.90	0.95 → 0.525
5	0.825 → 0.785	0.825 → 0.5187
10	0.7 → 0.65	0.7 → 0.5729 (T0), 0.5091 (T2)

Figure 4-2: Peace Bot Intervention Effectiveness

The node placement strategy at positions 3, 5, and 10 creates effective toxin reduction points that block toxic content throughout the network as shown in Figure 4-2. When peace bots intervene at stage three, they establish the most substantial influence on all toxicity results from that point forward.

4.3 Peace Bots and Experiments

To reduce Toxicity, we came up with the concept of peace bots, initally when we thought of +HC users (section 3.3), we wanted the same to apply to them, these bots are suppose to reduce Toxicity of the network at user level, by adding a tweet of their own to targetted audience, the more the audience the more significant the impact, hence we try to target the users with Lowest out-degree so that we can reach everyone evenly with more lavish spread, we think that having a bot that sends out specific amount of low Toxicity (0.05) would help reduce the overall average Toxicity of the network, the bot will get activated once the user assigned *peace_bot_nodes* comes into the simulation, then the bot will keep on sending the low Toxicity out everytime that user reposts. Keeping this in mind, we held a few experiments to check this hypothesis. Shown in Table 4.4

Experiment	**Start Tox**	**# Peace Bots**	**Value**	**Selection Method**
1	0.9	0	–	None
2	0.9	$F * \log_2 n$	0.05	Random
3	0.9	$F * \log_2 n$	0.05	Lowest out-degree
4	0.9	$2F * \log_2 n$	0.05	Lowest out-degree

Table 4.4: Summary of Simulation Experiments with Peace Bot Configurations

1. Experiment 1 covers the baseline, no peace bot part, to see how much we can lower the Toxicity.

2. Experiment 2 covers where to deploy these bots, keeping it random.

3. Experiment 3 covers whether a Low out-degree is a good way to start.

4. Experiment 4 will be used to experiment with the number of peace bots.

In further subsections, we will cover the essential questions of the properties of peace bots. We are still running these experiments on the BA graph and Koo/Gab's Giant connected component for better results. These results would allow us to answer

questions, such as the number of bots, which category to apply the bots to, and so on.

4.3.1 How many peace bots?

One purpose of running the experiments is to know how many peace bots will be required to impact the network's Toxicity significantly; that is where the F factor comes in, and we wanted to steadily increase the number of peace bots. Hence, we took $log_2 n$ the peace bots, where n is the total number of nodes in the graph. We needed to find the ideal value of this multiplicative factor, so we started with the baseline number $F = 20$, then increased $F = [30, 50, 100, 200]$.

After running all the experiments mentioned in section 4.2, which means now we have 4 experiments total and avg user data for graph sizes of $25k, 50k, 75k\&100k$, for each factor F, and we ran each for 5 times. That's a lot of data points so we needed to take an average of these too. Summarizing all the results into the Tables **??** and 4.6. Also, the figures 4-3 and 4-4 show the Bot's effectiveness in reducing toxicity on average during the 5 runs and across all the sizes. Figures 4-5 and 4-6 show the impact of the bots.

Table 4.5: Toxicity Reduction (%) by Scaling Factor (F) and Network Size (n)

F	Network Size (n)					
	25000	50000	75000	100000	275039 (Koo)	72891 (Gab)
20	4.51	3.71	3.30	2.23	0.14	0.47
30	6.27	3.57	3.61	3.19	0.12	0.94
50	7.76	5.55	5.54	4.76	0.38	0.92
100	12.06	8.12	6.84	6.38	0.52	1.72
500	28.08	20.02	16.37	14.81	2.81	9.19

Note: For BA graphs, Double Bot Placement is shown and for koo and gab Random Bot Placement, as they have the highest reductions

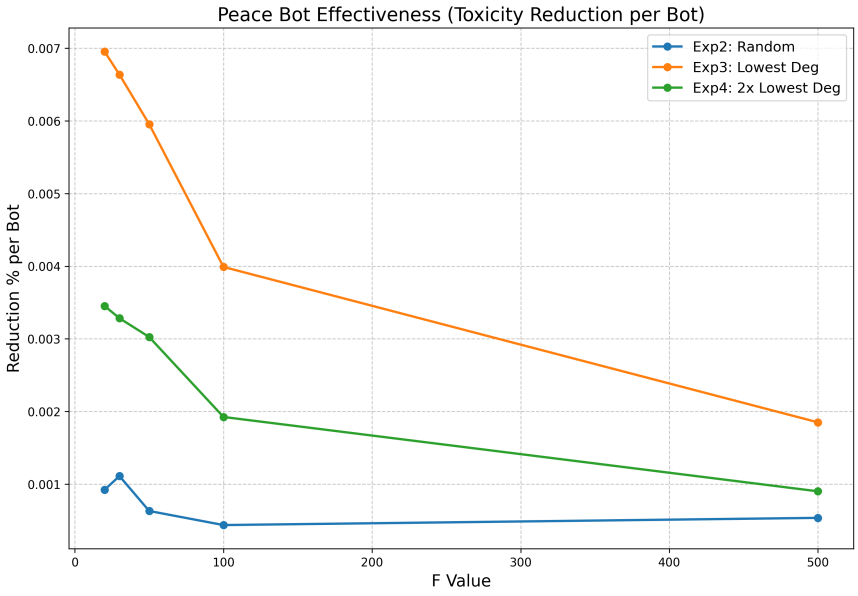


Figure 4-3: Bot effectiveness for 100K nodes BA graph

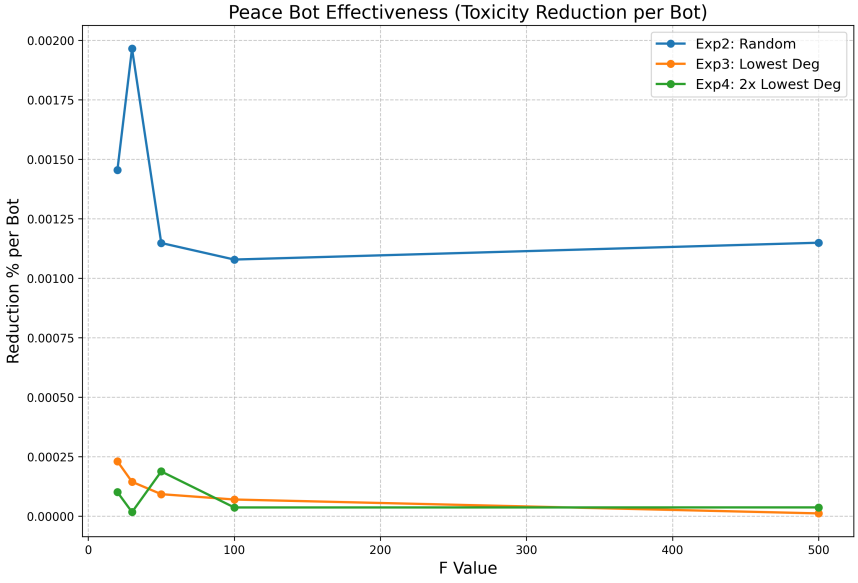


Figure 4-4: Bot effectiveness for Gab network

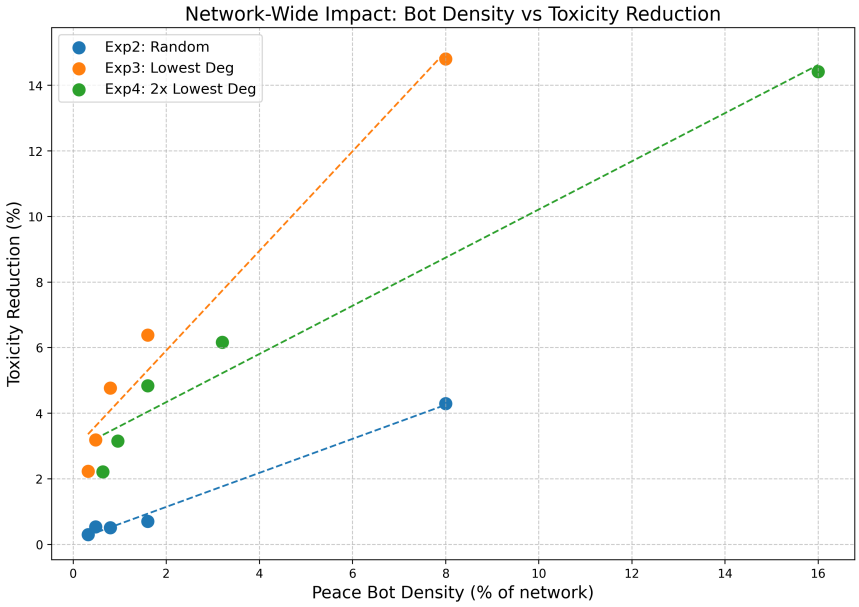


Figure 4-5: Bot Density Impact for 100K nodes BA graph

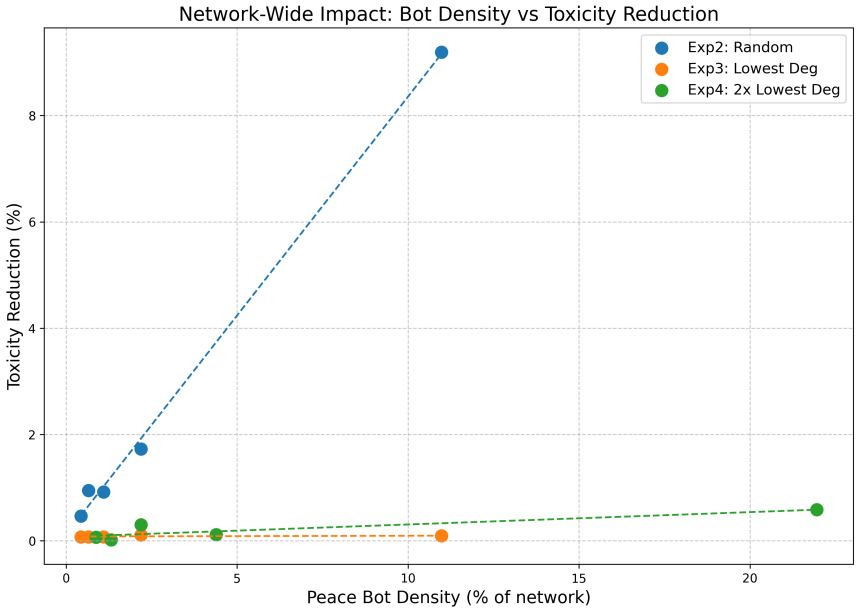


Figure 4-6: Bot Density Impact for Gab network

Table 4.6: Strategy Effectiveness Comparison

Network Size (n)	**Base vs. Random (%)**	**Base vs. Low (%)**	**Base vs. Double (%)**
25,000 (F=20)	0.64	4.51	4.92
50,000 (F=20)	0.40	3.71	3.69
75,000 (F=20)	0.19	3.30	3.30
100,000 (F=20)	0.30	2.23	2.21
275,039 (Koo, F=50)	0.38	0.04	0.05
72,891 (Gab, F=20)	0.47	0.07	0.06

Note: Values reflect toxicity reduction percentages for the recommended scaling factor (F)

"Base" = No peacebots; "Random" = Random placement; "Low" = Lowest-Degree placement; "Double" = 2F*log(n) bots with Lowest-Degree placement.

Chapter 5

Results

This chapter discusses the experimental findings regarding toxicity spread mechanisms and peacebot intervention performance on manufactured synthetic and organic social networking environments. The results validated our dynamic model, proving that proper strategic interventions can reduce toxicity.

5.1 Simulation Outcomes

Our simulations on synthetic Barabási-Albert (BA) graphs and real-world datasets (Twitter, Koo, Gab) revealed critical insights into toxicity dynamics:

- Over time, Peace bots brought down toxicity measurements in the network. Deploying bots at low out-degree nodes within a 100k-node BA graph resulted in **14.81%** decreased average toxicity as per Table 4.5.
- The mathematical model demands bot deployment based on the logarithm of the network size ($F \cdot \log_2 n$). The highest toxicity decrease occurred when $F = 20$ was applied to BA graphs (Table 4.5). The algorithm that selected low out-degree nodes succeeded in synthetic networks by achieving better results than random placement strategies by an **8.80%–5.12%** margin (Table 4.6).
- **Platform-Specific Behavior**: The echo-chamber structure on Gab network limited the effectiveness of interventions since baseline toxicity scores were al-

ready high (Table 3.6). Copycats (92.49%) were the primary form of behaviour observed on Koo while platform-wide toxicity measurements remained under control. Random placement of bots resulted in a **0.38%** reduction of toxicity in the system. The dynamic changes in user roles in Twitter platforms enabled better intervention success, according to Table 3.8.

5.2 Peace Bot Effectiveness

The experiments highlighted two key strategies for deploying peace bots:

- Low out-degree nodes showed strategic significance because placing bots at these points reduced the spread of toxicity in synthetic networks. Bot placement within 20k-node BA networks at low out-degree nodes decreased toxicity by **4.92%** better than random placement methods (Figure 4-5). Since high in-degree nodes of Gab amplify content propagation, they demonstrated better effectiveness than other nodes (Figure 4-6).

- During peace bot post configuration, selecting a toxicity value set at **0.05** achieved a suitable combination between visibility reach and impact success. The selected toxicity value created a risk of making the platform show content with minor toxicity, while values too low did not affect the outcome significantly. Graphs with multi-hop propagation extending from 3 to 5 hops achieved the maximal distribution. They simultaneously reduced toxicity by **3.71%** throughout 50k-node networks according to Table 4.6.

5.3 Dynamic User Behavior

Our analysis of user role transitions revealed:

- **Instability of Roles**: Over 60% of users changed roles (AMP, ATN, CC) within 13 weeks (Figures 3-12, 3-13 and 3-14, respectively).

- **Toxicity Buckets**: Shifts in user toxicity followed power-law distributions (Table 3.9), enabling probabilistic modelling of role transitions.

5.4 Limitations

- **API Constraints**: Perspective API's rate limits and language biases affected data completeness, particularly for non-English posts on Koo and Gab.
- **Toxicity as a term**: Defined by Perspective, toxicity is supposed to remain between 0 and 1; hence, during the simulation, if it goes above/below that, we had to cap it off.
- **Simulation Scalability**: Experiments on Koo and Gab's full network (300k, 72K nodes) required approximations, potentially underestimating bot efficacy.
- The model comprises many random variables, which may need to be changed based on the studied network. The behaviour-change probabilities and user shifts are coming from the Twitter dataset.

Chapter 6

Conclusion

The research investigated toxicity propagation in social networks while testing different intervention approaches to understand user activities. Our research findings establish new directions for theoretical and practical solutions which combat online danger.

6.1 Key Contributions

- **Dynamic Behaviour Modelling**: We demonstrated that user roles (AMP, ATN, CC) are fluid, with **92–95%** of users acting as copycats (Table 3.6). Role transitions follow probabilistic patterns influenced by neighbourhood toxicity (Table 3.8), necessitating dynamic over static models.

- **Intervention Efficacy**: Peace bots reduced toxicity by **2.23–4.92%** in synthetic networks (Table **??**). Strategic placement at low out-degree nodes or central hubs (depending on the platform) maximised impact.

- **Cross-Platform Insights**: Gab's unmoderated environment amplified toxicity shifts (**-0.3587** for attenuators vs. **-0.1022** on Twitter).

6.2 Future Work

- **Targetted Bots**: Maybe we can apply bots to only one particular behaviour and see how that changes the toxicity of the network.
- **Multilingual Toxicity Detection**: Expanding Perspective API's coverage to underrepresented languages (e.g., Hindi on Koo).

6.3 Final Remarks

Reducing toxicity demands detailed knowledge of network operations and user conduct patterns. This paper presents a method to develop better online communities by integrating computational models and planned interventions. Social networks will keep evolving, and our defence methods must adapt by maintaining ethical measures that protect users from digital threats.

List of Algorithms

Bibliography

[1] Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th international conference on World Wide Web companion*, pages 759–760, 2017.

[2] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[3] Sunandan Chakraborty, Joyojeet Pal, Priyank Chandra, and Daniel M. Romero. Political tweets and mainstream news impact in india: A mixed methods investigation into political outreach. In *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*, COMPASS '18, New York, NY, USA, 2018. Association for Computing Machinery.

[4] Allan M Collins and Elizabeth F Loftus. A spreading-activation theory of semantic processing. *Psychological review*, 82(6):407, 1975.

[5] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. *Proceedings of the International AAAI Conference on Web and Social Media*, 11(1):512–515, 2017.

[6] Paula Fortuna and Sérgio Nunes. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51(4):1–30, 2018.

[7] Daniel Hickey, Daniel MT Fessler, Kristina Lerman, and Keith Burghardt. X under musk's leadership: Substantial hate and no reduction in inauthentic activity. *PloS one*, 20(2):e0313293, 2025.

[8] Jigsaw. Perspective api documentation. https://developers.perspectiveapi.com/s/about-the-api. Accessed: 2023-04-15.

[9] Jigsaw and Google's Counter Abuse Technology team. Perspective api. https://www.perspectiveapi.com, 2017. Accessed: 2023-04-15.

[10] Fang Jin, Edward Dougherty, Parang Saraf, Yang Cao, and Naren Ramakrishnan. Epidemiological modeling of news and rumors on twitter. In *Proceedings of the 7th Workshop on Social Network Mining and Analysis*, SNAKDD '13, New York, NY, USA, 2013. Association for Computing Machinery.

[11] Jin Kim, Andrew Guess, Brendan Nyhan, and Jason Reifler. The distorting prism of social media: How self-selection and exposure to incivility fuel online comment toxicity. *Journal of Communication*, 71, 09 2021.

[12] Maryam Maleki, Mohammad Arani, Esther Mead, Joseph Kready, and Nitin Agarwal. Applying an epidemiological model to evaluate the propagation of toxicity related to covid-19 on twitter. 2022.

[13] Maryam Maleki, Esther Mead, Mohammad Arani, and Nitin Agarwal. Using an epidemiological model to study the spread of misinformation during the black lives matter movement. *arXiv preprint arXiv:2103.12191*, 2021.

[14] Omar Melikechi, Alexander L Young, Tao Tang, Trevor Bowman, David Dunson, and James Johndrow. Limits of epidemic prediction using sir models. *Journal of Mathematical Biology*, 85(4):36, 2022.

[15] Maya Mirchandani. *Digital hatred, real violence: Majoritarian radicalisation and social media in India*. Observer Research Foundation, 2018.

[16] Marzieh Mozafari, Reza Farahbakhsh, and Noël Crespi. Bert for hate speech detection: A transfer learning approach for cyber-security. *Security and Communication Networks*, 2019:1–10, 2019.

[17] Karsten Müller and Carlo Schwarz. Fanning the flames of hate: Social media and hate crime. *Journal of the European Economic Association*, 19(4):2131–2167, 2021.

[18] John Pavlopoulos, Jeffrey Sorensen, Lucas Dixon, Nithum Thain, and Ion Androutsopoulos. Toxicity detection: Does context really matter? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4296–4305, 2020.

[19] M. H. Ribeiro et al. "like sheep among wolves": Characterizing hateful users on twitter, 2017.

[20] Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A Smith. The risk of racial bias in hate speech detection. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 1668–1678, 2019.

[21] Anna Schmidt and Michael Wiegand. A survey on hate speech detection using natural language processing. *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10, 2017.

[22] Gautam Kishore Shahi and Tim A Majchrzak. Hate speech detection using cross-platform social media data in english and german language. *arXiv preprint arXiv:2410.05287*, 2024.

[23] Nga Than. *A Sociology of Gab: A Computational Analysis of a Far-Right Social Network*. PhD thesis, City University of New York, 2023.

[24] Aatman Vaidya, Seema Nagar, and Amit A Nanavati. Analysing the spread of toxicity on twitter. In *Proceedings of the 7th Joint International Conference on Data Science & Management of Data (11th ACM IKDD CODS and 29th COMAD)*, pages 118–126, 2024.

[25] Qiyao Wang, Zhen Lin, Yuehui Jin, Shiduan Cheng, and Tan Yang. Esis: Emotion-based spreader-ignorant-stifler model for information diffusion. *Knowledge-Based Systems*, 81, 02 2015.

[26] Zeerak Waseem and Dirk Hovy. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93, 2016.

[27] Savvas Zannettou, Barry Bradlyn, Emiliano De Cristofaro, Haewoon Kwak, Michael Sirivianos, Gianluca Stringini, and Jeremy Blackburn. What is gab: A bastion of free speech or an alt-right echo chamber. In *Companion Proceedings of the The Web Conference 2018*, pages 1007–1014, 2018.