Analysing the Spread of Toxicity on Twitter

Aatman Vaidya
aatman.v@ahduni.edu.in
Ahmedabad University
Ahmedabad, India

Seema Nagar
senagar3@in.ibm.com
IBM India Research Lab
Bangalore, India

Amit A. Nanavati
amit.nanavati@ahduni.edu.in
Ahmedabad University
Ahmedabad, India

ABSTRACT

The spread of hate speech on social media platforms has become a rising concern in recent years. Understanding the spread of hate is crucial for mitigating its harmful effects and fostering a healthier online environment. In this paper, we propose a new model to capture the evolution of toxicity in a network – if a tweet with a certain toxicity (hatefulness) is posted, how much toxic a social network will become after a given number of rounds. We compute a toxicity score for each tweet, indicating the extent of the hatefulness of that tweet.

Toxicity spread has not been adequately addressed in the existing literature. The two popular paradigms for modelling information spread, namely the Susceptible-Infected-Recovered (SIR) and its variants, as well as the spreading-activation models (SPA), are not suitable for modelling toxicity spread. The first paradigm employs a threshold and categorizes tweets as either toxic or non-toxic, while the second paradigm treats hate as energy and applies energy-conversion principles to model its propagation. Through analysis of a Twitter dataset consisting of 19.58 million tweets, we observe that the total toxicity, as well as the average toxicity of original tweets and retweets in the network, does not remain constant but rather increases over time.

In this paper, we propose a new method for toxicity spread. First, we categorize users into three distinct groups: Amplifiers, Attenuators, and Copycats. These categories are assigned based on the exchange of toxicity by a user, with Amplifiers sending out more toxicity than they receive, Attenuators experiencing a higher influx of toxicity compared to what they generate, and Copycats simply mirroring the hate they receive. We perform extensive experimentation on Barabási–Albert (BA) graphs, as well as subgraphs extracted from the Twitter dataset. Our model is able to replicate the patterns of toxicity.

CCS CONCEPTS

• **Information systems** → **Social networks**; **Internet communications tools**; • **Human-centered computing** → *Social networking sites*; **Social network analysis**; **Social networks**.

KEYWORDS

Hate Speech, Online Social Media, Twitter, Network Analysis, Toxicity, Network Dynamics

ACM Reference Format:
Aatman Vaidya, Seema Nagar, and Amit A. Nanavati. 2024. Analysing the Spread of Toxicity on Twitter. In *7th Joint International Conference on Data Science & Management of Data (11th ACM IKDD CODS and 29th COMAD) (CODS-COMAD 2024), January 4–7, 2024, Bangalore, India.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3632410.3632436

1 INTRODUCTION

Social media has become an integral part of our lives; it has changed the way we express ourselves, share information and interact with each other. With that said, there are adverse effects of social media, such as online harassment, cyber-bullying and hate speech. Hate speech spreads like wildfire and is one of the major issues affecting online social media, leading to atrocities like the Pittsburgh synagogue shooting[1], the Rohingya genocide in Myanmar[2] and the shooting at the Sikh temple in Wisconsin[3]. A study found that there was a strong correlation between online hate speech leading to real-world violence [13]. Therefore, understanding hate spread is of utmost importance.

Instead of labelling a tweet as “hateful" or “not hateful", we assign a “toxicity" score in the range [0-1] to each tweet. This score measures *how much* hateful a tweet is. For this, we used the Perspective[4] API on the data published by [18]. The Perspective API defines “toxicity" as *“rude, disrespectful, unreasonable comment that is likely to make someone leave a conversation”*. Each user may send and receive tweets with various toxicities in the range [0-1].

Broadly, past work has modelled the spread of hate in two ways. One, in which non-hateful users get converted to becoming hateful based on how hateful messages are spreading in the network. These are the spreading activation (SPA) models. In such models, usually, a hateful user never becomes non-hateful. The other popular model, considers hate as a disease, where a person can get exposed to hate, become hateful and even recover from it. These are the susceptible-infected-recovered (SIR) models and its variants. In this case, since hate is a disease, a person either has it or not. In both these classes of models, statically or dynamically, users are in a state of being hateful or non-hateful.

In this paper, instead of considering hate as being binary, we treat it as non-binary; something that exists as a spectrum in the range [0-1]. As a result, we can no longer classify users as being hateful or non-hateful at any instant of time. Just for clarity of

CODS-COMAD 2024, January 4–7, 2024, Bangalore, India

ACM ISBN 979-8-4007-1634-8/24/01…$15.00
https://doi.org/10.1145/3632410.3632436

[1] https://www.nytimes.com/2018/10/27/us/active-shooter-pittsburgh-synagogue-shooting.html

[2] https://www.reuters.com/investigates/special-report/myanmar-facebook-hate

[3] https://www.washingtonpost.com/nation/2018/11/30/how-online-hate-speech-is-fueling-real-life-violence/

[4] https://perspectiveapi.com/

exposition, we refer to hatefulness in the range [0-1] as *toxicity*. An immediate consequence of this is that tweets of various toxicities are travelling through the network, and we want to understand how toxicity is spreading across the network and changing with time. In this setting, the natural questions to ask are: (a) How are users responding to tweets with varying toxicities? (b) How is the toxicity distribution changing over time?

Considering the second question first, we found that total and average toxicity of the network is increasing over time. Since toxicity is not conserved, SPA models are not suitable for modelling the toxicity of a network.

We propose a new model to capture the spread of the *spectrum* of hate speech. Our model is based on user behaviour and captures the two important factors, a) toxicity exists as a spectrum and b) toxicity is not conserved.

In a social network, each user is a *conduit of toxicity* – influenced by her (incoming) neighbours and influences her (outgoing) neighbours. Based on some statistical analysis, we could divide the set of users into 3 categories: “Amplifiers", who spread more hate than they receive; “Attenuators", who spread less hate than they receive; and “Copycats", who spread as much hate as they receive. Each type of user essentially administers a *shift* in the toxicity it receives before it sends it out.

We empirically validate the proposed model on both the simulated Barabási–Albert (BA) graphs of various sizes as well as samples of real worlds graphs from the data we studied. In one variant of the experiments, we add an ageing factor to a (re)tweet so that it does not continue to be retweeted forever. For BA graphs, we use shift values and the user category distribution in three categories based on the observations on empirical data.

Our model is able to reproduce the expected increase in toxicity and also simulate the impact of the distribution and placement of the three categories of users in the social network.

2 REVIEW OF LITERATURE

There is plenty of research work to detect the presence of hate speech in social media content [2, 6, 7, 26, 27], but understanding the dynamics of spread is still in its infancy.

Works such as [18] and [12] use belief propagation to detect hateful users on Twitter and Gab, respectively. Further, they look at the diffusion dynamics of posts generated by hateful vs non-hateful users. They find that posts of hateful users receive a much larger audience and a faster rate. They also find that hateful users are densely connected to one another and produce close to 1/4th of the content on Gab.

[5] showed how the energy in a SPA model is conserved. Recently, [16] modelled detection of hateful users for various hateful topics, using Spread and Activation (SPA) framework. They showed that if a user is interested in a hateful topic, his neighbours may also get interested in that hateful topic, further strengthening the philosophy of birds of a feather flock together[15].

Researchers have customized SIR and SIS models to SEIR (Susceptible Exposed infected Removed)[24], S-SEIR (Single layer – SEIR) [28], SCIR (Susceptible Contacted Infected Removed)[29], ir-SIR (Infection Recovery SIR)[4], FSIR (Fractional SIR) [9] and ESIS (Emotion-based spreader-ignorant-stifler) [25] models to account for nuances of information spreading on social media platforms. All of these models assume that any user is in one of a few sets of states at any given time, such as: “infected (hateful), “exposed", “cured"(not hateful), etc.

[1, 14, 17, 30] implemented deep learning models to improve classification results for hate detection. [20, 21] looked at fear speech on social media. They found that fear speech users gain more followers and occupy central positions in the network. Additionally, they can more effectively engage with normal users than hate-speech users. [10] found out that hateful content written by verified users is more likely to go viral than content written by non-verified users.

3 ANALYSIS OF TOXICITY ON TWITTER

3.1 Dataset and Pre-Processing

We use the dataset[5] published by Ribeiro et al. [19] containing 100, 386 users and 19.58 million tweets. The majority of the tweets in the dataset are from the months of January 2017 to October 2017. The dataset also has a directed retweet graph $G = (V, E)$, where each node $u \in V$ represents a Twitter user. Each edge $(u, v) \in E$ represents a retweet in the network; there will be an edge from u to v if v retweets a tweet from u. The retweet graph has 2, 286, 592 edges. The dataset comes with a few labelled users as hateful or normal. The labels are available for 4,972 users, out of which 544 users are labelled as hateful and the others as normal. However, we do not make use of these labels in our analysis.

In the dataset, tweets are not labelled with a toxicity score.

Given a piece of text, the Perspective API[6] assigns scores in [0-1] for a variety of attributes such as *toxicity*, *severe toxicity*, *profanity*, *identity attack*, and *insult*. In this paper, we assign each tweet a score using the *toxicity* attribute. This score represents the extent of the hatefulness of a tweet. The score is also a probability of how many people might find the text to be rude or disrespectful. The score also represents the likelihood of how rude or disrespectful a text could be perceived by people. Perspective API does not compute toxicity score for a piece of text below a certain length. After passing the tweets through the API, we are left with 17.22 million tweets and 99,980 users in the dataset.

3.2 Analysing the Dataset

We begin by examining the temporal evolution of toxicity. Figure 1 shows the sum of the toxicity of all the tweets across all the weeks on a semilog scale. To determine the total toxicity within a given time period, we aggregate the toxicities of all tweets posted during that period.

We can see that there is a rise in total toxicity over time. However, the number of users and tweets is not constant during this interval due to potential user entry or exit as well as variable tweet count. In order to eliminate this effect, we examine the average toxicity of all the tweets across the weeks as shown in Figure 2. The average toxicity of all the tweets is also increasing over time. For any given week, we divide the total toxicity by the total number of tweets to get the average for the particular week.

[5] https://www.kaggle.com/datasets/manoelribeiro/hateful-users-on-twitter
[6] https://perspectiveapi.com/

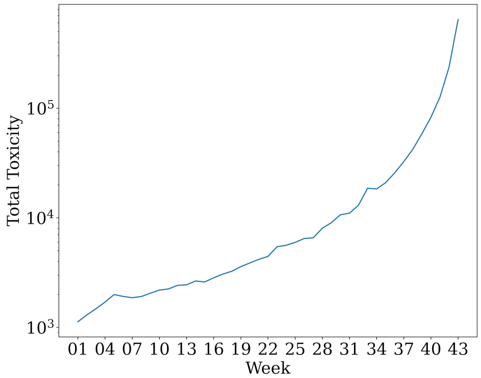


Figure 1: Total Toxicity Distribution over the Weeks. The distribution over the months shows a similar trend.

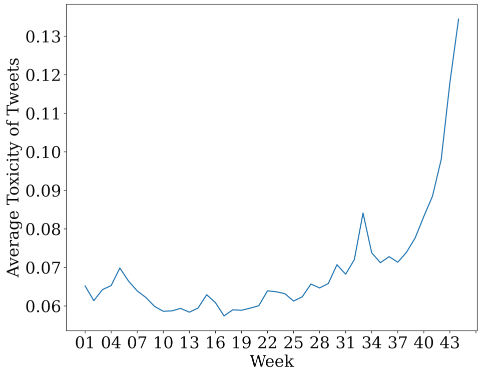


Figure 2: Average Toxicity Distribution of Tweets Across Weeks. We divide the total toxicity by the total number of tweets to get the average for the particular week.

Figure 3 shows that tweets with higher toxicity values are retweeted significantly more than the tweets with low-toxicity values. This appears to be consistent with another study which showed that hateful tweets are retweeted more significantly than non-hateful tweets [8].

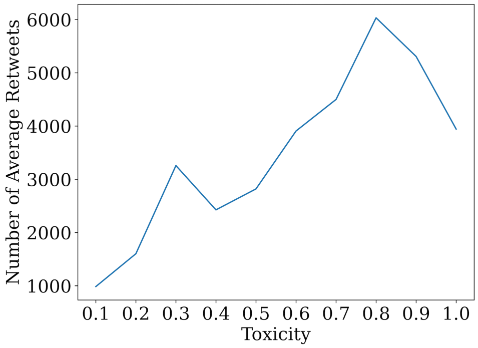


Figure 3: Average Number of Retweets by Toxicity.

3.3 Preliminary Findings

Based on the preceding exploratory analysis, we come to two preliminary conclusions:

(1) Neither the total toxicity nor its average is conserved in the network (Figures 1, 2).
(2) Users respond differently to tweets of different toxicities. This can be observed from Figure 3.

4 A NON-CONSERVATIONAL APPROACH FOR MODELLING TOXICITY SPREAD

4.1 User Classification

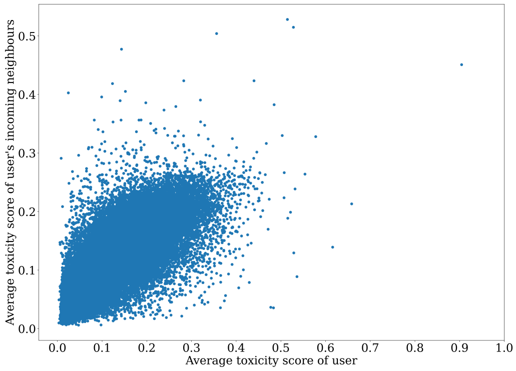


Figure 4: Average Toxicity of a User versus Average Toxicity of its In-Degree Neighbourhood

We now examine user behaviour in order to measure the *influence* of the social network upon a user and vice-versa. For this, we compare a user's average toxicity with the average toxicity of the user's *indegree* neighbours. Figure 4 shows the result. Since we are considering only the indegree neighbours of every node, this plot consists of 92, 847 users (the rest of them do not have indegree neighbours).

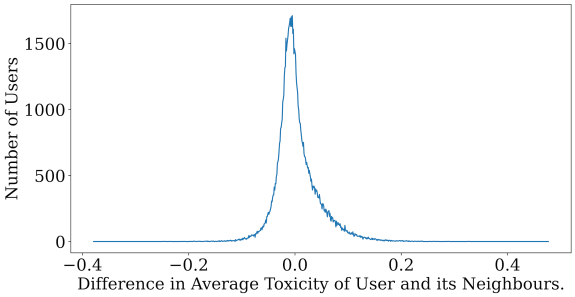


Figure 5: Distribution of the difference in Average Toxicity of User and its Neighbours. This distribution fails the Shapiro–Wilk (SW) and Kolmogorov-Smirnov (KS) normality tests.

For precisely measuring the influence of its indegree neighbours on each user, we calculated the difference between the user's average toxicity and the average toxicity of all its indegree neighbours.

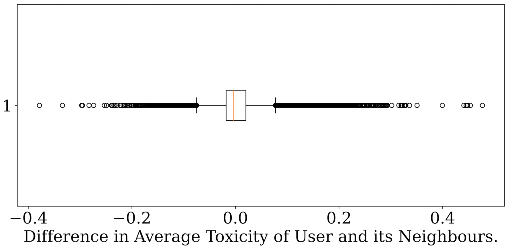


Figure 6: Box and Whisker plot for the difference data in Figure 5. Since the data is not normal, we use the IQR method to detect outliers and separate them from typical users. The outlier users on the right are called *amplifiers*, on the left are called *attenuators* and the remaining are called *copycats*.

Figure 5 shows the distribution of these differences. We used the Shapiro-Wilk test[23] and the Kolmogorov-Smirnov test [11] of normality to determine if the differences among these average toxicities follow a normal distribution. We found that this distribution failed the test and therefore is not normal.

In cases where distribution is not normal, the Interquartile Range (IQR) proves to be a useful measure [22]. The IQR method detects *outliers* and separates them from typical users. Figure 6 shows a box-and-whisker plot of the differences. This approach naturally divides the set of users into 3 disjoint subsets. The outlier users on the right are called *amplifiers*, on the left are called *attenuators* and the remaining (typical users) are called *copycats*. The *amplifiers* send out more toxicity than they receive, the *attenuators* send out less toxicity than they receive and the *copycats* send out almost the same toxicity as they receive.

Table 1: Average Toxicity Shifts and Proportions of User Categories

User Categories	Average Toxicity Shifts	User Proportion
amplifiers	+0.1133	5.33%
attenuators	-0.1022	1.39%
copycats	-0.000497	93.28%

We determined the *average* toxicity shifts for each user category (amplifier/attenuator/copycat) after classifying the users. These shifts indicate the average change (*shift*) in incoming toxicity that the user will apply before transmitting the message further.

Table 1 presents the shifts and user proportions (percentage of users) for each user category.

Building upon the user classification and further to the preliminary findings from section 3.3, we establish further assumptions in order to define our model:

- We consider the network to be static – users do not enter or leave the system.
- Rather than compute each user's toxicity shift for each range of toxicity, we consider only averages.
- A user's classification does not change with time.

4.2 Theoretical Formulation of the Proposed Model

We consider a directed graph $G = (V, \vec{E})$ where V denotes the set of users and $\vec{E}$, the edges connected them, representing the network structure of our study. Each node in the graph represents a user on the social media platform. The set $T = T_1, T_2, \ldots, T_K$ represents the timestamps at which we observe the spread of hate speech in the network.

Algorithm 1 *tox-spread*

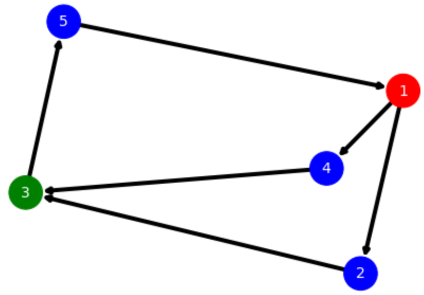
```
Require: The network of users $G = (V, \vec{E})$, list of users with their shifts in shift[].
Require: The number of iterations, $T$.
Require: The initial tweet(s) with their toxicity values stored in $toxVal[v]$.
Ensure: The final values of toxicity for each user stored in $toxVal[]$, and toxicity of the whole networks in $toxicity[T]$.
1: for each timestamp $t$ in $[1, \ldots, T]$ do
2:   $toxicity(t) \leftarrow \sum_{i=1}^{n} toxVal[v_i]$, $n \in$ number of nodes {Total toxicity at timestamp $t$}
3:   for node $v$ in toxVal that are not empty i.e. in the current timestamp do
4:     for successors of node $v$ do
5:       Let $v_{nbr}$ be the receiving neighbor of $v$
6:       if $v_{nbr}$ in a certain user category then
7:         shift(nbr(v)) ⟵ shift(v) {Check for user category; attenuators, amplifiers and copycats have their respective shifts}
8:       end if
9:       tox ⟵ tox+shift(nbr(v)) {A neighbour who has received a tweet applies the shift to the received tweet and further forwards a tweet with updated toxicity}
10:      if $tox > 1$ then
11:        $tox \leftarrow 1$
12:      else if $tox < 0$ then
13:        $tox \leftarrow 0$
14:      end if
15:      $toxVal[v_{nbr}] \leftarrow tox$
16:    end for
17:  end for
18:  $toxVal[v] \leftarrow clear$ {clear the values of node v}
19: end for
```

To model the spread of toxicity in the network at each subsequent timestamp, we present Algorithm 1 *tox-spread*. At the initial timestamp T_0 (line 1), a specific node v posts a tweet with a toxicity value denoted as *tox* (a node could also post multiple tweets with different toxicity values). This toxicity value represents the degree of hatefulness of the tweet. The shift applied by each node in the network for the toxicity received will depend on its user category and can be denoted as *shift*(v). These shift values capture the individual propensity of nodes to amplify or suppress the toxicity of the content they receive. The total toxicity of the network at T_0 is $toxicity(T_0) = tox$ (line 2). Each neighbour of node v will have a specific shift based on its user category denoted by *shift(nbr(v))* (line 3-8). This shift will be added to *tox* and the updated value of toxicity for that tweet will be *tox* = *tox* + *shift(nbr(v))* (line 9).

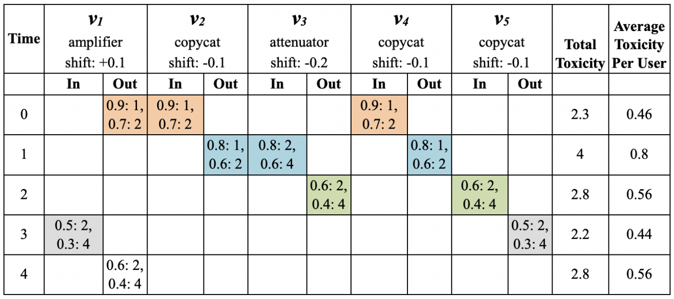
4.3 Demonstration on an Example Graph

We demonstrate the working of the model on a small graph. Figure 7b shows the simulation for our model on the graph in Figure 7a. The format for denoting the tweet's toxicity and their count is "toxicity value: count of tweets". We initialise Node 1 with multiple tweets as follows - "$0.9 : 1, 0.7 : 2$". In this example, v_i is an amplifier, v_2, v_4 are copycats, and v_3 is an attenuator. The toxicity shift for amplifiers is +0.1, for attenuators is −0.2, and for copycats is −0.1. The shifts and initial toxicity values are arbitrary, just for the purpose of a simple demonstration. For clarity, in Figure 7b, the incoming tweet toxicities are shown for each node. The final two columns show the sum and average of each row in the table.

We can easily observe that the total/average toxicity in the network is not conserved. The changes in the total toxicity also depend on the configuration of the copycats, attenuators and amplifiers.



(a) An example graph: blue nodes are copycats, green nodes are attenuators and red nodes are amplifiers.



Time	v_1 amplifier shift: +0.1		v_2 copycat shift: -0.1		v_3 attenuator shift: -0.2		v_4 copycat shift: -0.1		v_5 copycat shift: -0.1		Total Toxicity	Average Toxicity Per User
	In	Out	In	Out	In	Out	In	Out	In	Out		
0		0.9: 1, 0.7: 2	0.9: 1, 0.7: 2				0.9: 1, 0.7: 2				2.3	0.46
1				0.8: 1, 0.6: 2	0.8: 2, 0.6: 4			0.8: 1, 0.6: 2			4	0.8
2						0.6: 2, 0.4: 4			0.6: 2, 0.4: 4		2.8	0.56
3	0.5: 2, 0.3: 4									0.5: 2, 0.3: 4	2.2	0.44
4		0.6: 2, 0.4: 4									2.8	0.56

(b) This table shows the running to Algorithm *tox-spread* on the network in Figure 7a above. Each node "shifts" the input toxicity (column "In") by *shift* and sends it to its outgoing neighbours (column "Out") based on whether it is an amplifier, attenuator or a copycat. In the first row (at Time 0), the output of v_1 become the input for v_2 and v_4. In the second row, the outputs of v_2 and v_4 are inputs for v_3.

Figure 7

4.4 Amplifier, Attenuator and Copycat Characteristics

In this section, we analyse the characteristics of amplifiers, attenuators and copycats, both in terms of their network properties and their behaviour. Several questions arise. For example, *How does the average toxicity of an amplifier vary with its outdegree?*

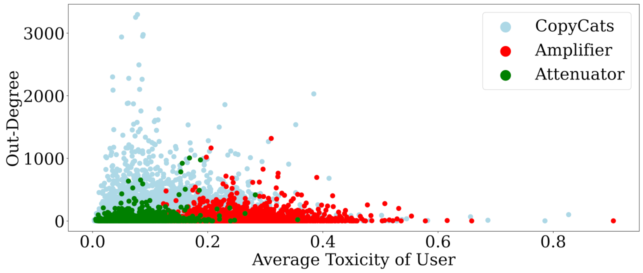


Figure 8: Average Toxicity of a User versus its Out-Degree

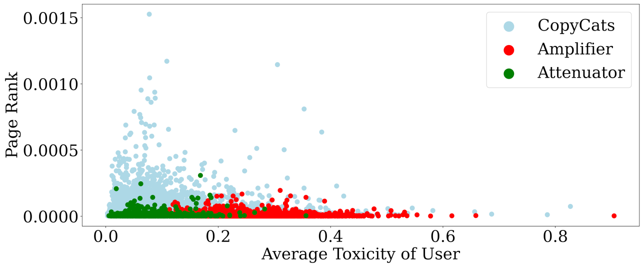


Figure 9: Average Toxicity of a User versus its Hub Value (PageRank on outdegrees)

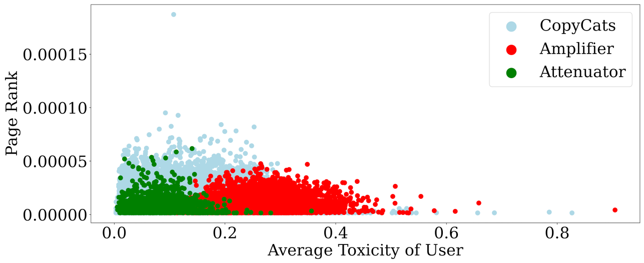


Figure 10: Average Toxicity of a User versus its PageRank - InDegree

Figure 8 shows the scatterplot of the average toxicity of each node plotted against its outdegree for amplifiers, attenuators and copycats. We observe that the outdegrees of the copycats are among the highest. The average toxicities of the attenuators are less than those of the amplifiers, but for a given outdegree, there are more amplifiers than attenuators. There are quite a few copycats with large average toxicity but low outdegree.

Figure 9 shows a similar plot with hubvalues on the Y-axis. The highest hubvalues belong to the copycats, clearly highlighting their dominant role in the spread of toxicity.

Figure 10 shows a similar plot with PageRank on the Y-axis. The highest pagerank values belong to the copycats, indicating their importance as recipients of links.

Figure 11 show the number of users of each category active across the combinations of the actions of sending original tweets,

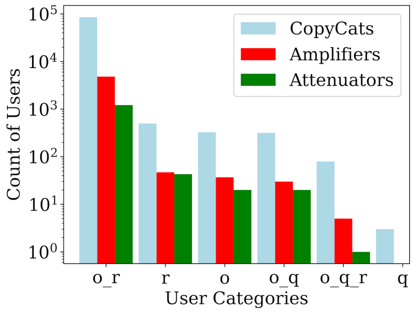


Figure 11: Tweet Distribution of each User Category

retweets and quoted tweets. In each bucket, once again, we find that the copycats have an order of magnitude more number of users participating in the set of activities represented by the bucket. Further, note that the number of amplifiers who are sending original tweets and retweets is almost 5 times the number of attenuators indulging in the same two activities. The same is true of the amplifiers and attenuators doing all the 3 activities.

Table 2: Attribute Assortativity Co-efficient for User Category

User Category	Attenuator	Amplifier	CopyCats
Attenuator	7.26e-18	0.02068	0.02844
Amplifier	0.02068	0	0.10937
CopyCats	0.02844	0.10937	0

How are the amplifiers, attenuators and copycats distributed in the network? Are the amplifiers (attenuators) well-connected among themselves? Table 2 shows a lack of evidence for both homophily and inverse homophily for all types of users, since all the values in the Table are close to 0. This suggests that the amplifiers, attenuators and copycats, are almost randomly connected, without any preference or prejudice among each other.

5 EXPERIMENTS

5.1 Experiments Overview

We validate our proposed model on simulated BA graphs [3] as they resemble real-world networks, as well as the subgraphs sampled from the real data we studied in this paper. BA graphs follow the power law of degree distribution and exhibit the *rich-gets-richer* phenomenon seen in many real-world networks.

In summary, we aim to address the following questions through our experiments:

(1) Does the total toxicity increase when we run our model on BA graphs?
(2) How does the network topology – the placement of attenuators, amplifiers, and copycats – matter? What effect does it have on total toxicity?
(3) In real-world networks, tweets don't get retweeted indefinitely. What happens to the total toxicity when tweets die out with time? How does the placement of attenuators, amplifiers, and copycats matter in this case?

5.2 Experimental Test Bed

We use the Python library NetworkX[7] to generate the BA graphs of various sizes, i.e. $5,000$, $10,000$, $25,000$, and $50,000$ nodes, all with an m value of 5. The m value determines the number of edges connected from a new node to the existing nodes. We modify the graph by making the generation directed and reversing the direction of the edges to closely resemble the retweet graph in the dataset. We use toxicity shifts and user proportions from real data as shown in Table 1. These graphs allowed us to test the model on a range of different networks with sufficient variation in size and scale. In our simulations, we initialise a node with a single tweet of toxicity value 0.0985 that lives on forever and undergoes changes in its toxicity based on the type of users it encounters.

We also simulate our model on the retweet graph described in section 3.1. For better comparison, we extract subgraphs of the retweet graph that are similar in size to the BA graphs. After that, for each subgraph, we calculate the toxicity shifts and find the amplifiers, attenuators and copycats by the method we followed in section 4.1. To find a starting point for the simulation, we find the largest strongly connected component in the graph and find a node with the highest eccentricity.

5.3 Results

We wanted to understand the impact of the placement of these user types in the network. For example, what would happen to the overall toxicity in the network if all the high-outdegree nodes were attenuators as opposed to being randomly assigned?

We devise five scenarios to assign nodes as amplifiers, attenuators and copycats while maintaining the distribution in the three categories same as observed in the real data:

(1) Case 1 - All nodes are randomly assigned a user category.
(2) Case 2 - Nodes with High Out-Degree are assigned as Attenuators, and the remaining nodes are randomly assigned.
(3) Case 3 - Nodes with High Out-Degree are assigned as Amplifiers, and the remaining nodes are randomly assigned.
(4) Case 4 - Nodes with Low Out-Degree are assigned as Attenuators, and the remaining nodes are randomly assigned.
(5) Case 5 - Nodes with Low Out-Degree are assigned as Amplifiers, and the remaining nodes are randomly assigned.

Case 1, where all the assignments are done randomly, provides us with a baseline for comparison with the other cases.

Table 3 presents the results of our model for different node sizes, with an average of 5 simulations recorded for each case. We make the following observations:

- The total toxicity increases with time and graph size.
- Expectedly, Case 3 has the highest total toxicity of them all, and Case 2 the lowest (even lower than the baseline).
- Each entry in Case 3 is greater than Case 4, confirming that the effect of assigning amplifiers to high outdegree nodes is stronger than that of assigning attenuators to low outdegree

[7] https://networkx.org/

Table 3: Highest Value of Total Toxicity in all 5 cases when the model is simulated on BA Graphs. Note that Case 3, where the amplifiers have the highest outdegree results in the largest toxicity.

Nodes	m	Edges	Time	Case 1	Case 2	Case 3	Case 4	Case 5
5,000	5	24,846	2	2.45×10^2	5.81×10^1	6.3×10^2	3.12×10^2	2.21×10^2
			4	4.75×10^6	1.06×10^6	2.85×10^7	5.74×10^6	2.91×10^6
			6	2.11×10^7	5.83×10^6	1.23×10^8	2.4×10^7	1.14×10^7
			8	4.06×10^7	1.55×10^7	2.66×10^8	4.53×10^7	2.52×10^7
			36	8.26×10^7	2.46×10^7	**3.87×10^8**	6.66×10^7	3.31×10^7
10,000	5	49,784	2	3.94×10^2	8.2×10^1	1.01×10^3	3.72×10^2	3.59×10^2
			4	1.38×10^7	3.37×10^6	8.23×10^7	1.27×10^7	7.66×10^6
			6	6.89×10^7	1.78×10^7	3.69×10^8	6.34×10^7	3.28×10^7
			8	1.48×10^8	4.21×10^7	8.62×10^8	1.4×10^8	7.47×10^7
			40	5.31×10^8	1.48×10^8	**2.52×10^9**	4.09×10^8	2.32×10^8
25,000	5	124,812	2	5.81×10^2	8.78×10^1	1.55×10^3	5.7×10^2	4.95×10^2
			4	9.66×10^7	2.34×10^7	5.76×10^8	8.42×10^7	5.39×10^7
			6	6.11×10^8	1.54×10^8	3.56×10^9	7.08×10^8	3.15×10^8
			8	1.69×10^9	4.71×10^8	1.03×10^{10}	1.72×10^9	9.13×10^8
			46	5.81×10^9	1.93×10^9	**3.25×10^{10}**	5.64×10^9	2.25×10^9
50,000	5	249,772	2	3.49×10^3	3.8×10^2	1.34×10^4	3.48×10^3	3.02×10^3
			4	5.06×10^8	1.11×10^8	2.77×10^9	4.71×10^8	2.47×10^8
			6	3.35×10^9	9.03×10^8	1.92×10^{10}	3.14×10^9	1.7×10^9
			8	1.24×10^{10}	3.1×10^9	6.24×10^{10}	1.12×10^{10}	5.13×10^9
			52	4.85×10^{10}	1.25×10^{10}	**2.15×10^{11}**	4.86×10^{10}	1.5×10^{10}

Table 4: Highest Value of Total Toxicity in all 5 cases when the decay variation of model is simulated on BA graphs. Note that there is very little difference in the toxicity values between 8 and 10 timesteps.

Nodes	m	Edges	Time	Case 1	Case 2	Case 3	Case 4	Case 5
5,000	5	24,846	2	5.48×10^2	1.1×10^1	1.09×10^3	4.29×10^2	3.86×10^2
			4	1.66×10^4	1.10×10^1	6.82×10^4	1.49×10^4	1.28×10^4
			6	1.76×10^4	1.11×10^1	8.38×10^4	1.83×10^4	1.43×10^4
			8	1.81×10^4	1.11×10^1	9.06×10^4	2.12×10^4	1.46×10^4
			10	2.29×10^4	1.09×10^1	8.20×10^4	1.95×10^4	1.44×10^4
10,000	5	49,784	2	8.35×10^2	1.54×10^1	2.53×10^3	1.05×10^3	8.42×10^2
			4	4.54×10^4	1.49×10^1	2.13×10^5	4.71×10^4	3.67×10^4
			6	5.22×10^4	1.48×10^1	2.64×10^5	5.22×10^4	4.06×10^4
			8	6.03×10^4	1.50×10^1	2.59×10^5	5.47×10^4	3.91×10^4
			10	4.76×10^4	1.45×10^1	2.40×10^5	5.12×10^4	3.88×10^4
25,000	5	124,812	2	1.60×10^3	2.65×10^1	4.42×10^3	1.56×10^3	1.35×10^3
			4	1.32×10^5	2.71×10^1	7.01×10^5	1.39×10^5	1.05×10^5
			6	1.91×10^5	2.69×10^1	8.68×10^5	1.70×10^5	1.32×10^5
			8	**1.74×10^5**	**2.68×10^1**	**8.26×10^5**	**1.66×10^5**	**1.15×10^5**
			10	**1.72×10^5**	**2.67×10^1**	**8.39×10^5**	**1.55×10^5**	**1.30×10^5**
50,000	5	249,772	2	1.13×10^3	2.19×10^1	3.14×10^3	1.13×10^3	1.12×10^3
			4	3.06×10^5	2.20×10^1	1.65×10^6	3.52×10^5	2.36×10^5
			6	3.42×10^5	2.19×10^1	1.92×10^6	3.49×10^5	2.56×10^5
			8	**3.95×10^5**	**2.19×10^1**	**1.90×10^6**	**3.49×10^5**	**2.68×10^5**
			10	**3.28×10^5**	**2.20×10^1**	**1.85×10^6**	**3.65×10^5**	**2.58×10^5**

nodes. A similar statement can be made while comparing Case 5 and Case 2.

- Each entry in Case 5 is less than that in Case 1, suggesting that restricting amplifiers to low outdegree nodes makes the total toxicity less than the baseline.

To make our simulations on the BA graphs even more realistic, we now prevent tweets from being retweeted indefinitely. We incorporate an age factor for each tweet, ranging from [0-1], that gradually decays over time. Tweets are initialised with a (remaining) age factor of 1 for the simulation. The age factor decreases by 0.1

Table 5: Total Toxicity when the model is simulated on the Retweet Graph

Nodes	Edges	Time	Total Toxicity
5,278	2,447	2	1.2×10^{-1}
		4	1.38×10^{1}
		6	5.34×10^{2}
		7	5.38×10^{2}
		8	2.85×10^{4}
10,262	8,071	2	5.74×10^{-2}
		4	1.34×10^{-1}
		6	2.71×10^{-1}
		8	3.34×10^{-1}
		9	3.58×10^{-1}
24,118	56,214	2	2.29×10^{-1}
		4	1.85×10^{-1}
		6	8.33×10^{-1}
		8	1.09×10^{1}
		44	5.08×10^{35}
51,358	429,639	2	1.4×10^{0}
		4	2.6×10^{2}
		6	5.18×10^{7}
		8	1.08×10^{14}
		11	2.24×10^{20}

at each time step during the simulation. The age factor represents the probability of a tweet being forwarded in the simulation. Once the age factor reaches zero, the tweet is removed from the system. Since this is done probabilistically, a tweet may be deleted from the system before its age factor reaches zero.

Table 4 shows the results of the modified model for different node sizes, with an average of 5 simulations recorded for each case on BA graphs. All the observations made in the case of Table 3 hold true for Table 4 as well. Further, we observe that *almost all values in Table 4 are less than the corresponding values in Table 3.*

Table 5 shows us the results of the model on the retweet subgraphs. *In each case, we see a rise in total toxicity.*

We can now answer the questions we raised earlier in Section 5.1. We found that:

(1) Total toxicity rises in our model even with (a) random assignments of nodes to categories and (b) an age factor driven expiry of retweets.
(2) The placement of the amplifiers, attenuators and copycats matter.
(3) Total toxicity of the subgraphs also rises.

5.4 Discussion

In the dataset we used [19], 543 users out of 4,972 were labelled hateful using crowdsourcing. Instead of labelling users as “hateful” vs. “non-hateful", our analyses and model led us to a complementary division of users (based on how they receive and spread toxicity) as “amplifiers", “attenuators" and “copycats". These two approaches may be viewed as different dimensions of user categorisation. A user may be a “hateful-attenuator", a “non-hateful-copycat", etc. Of the 543 labelled hateful users, we found that 95 are amplifiers, 5 are attenuators, and 443 are copycats. This is not a contradiction. As seen through various plots in this paper, the attenuators, amplifiers and copycats have average toxicity values in a wide range, suggesting that any of them may or may not be hateful.

In future work, it may be interesting to consider both these axes as dimensions, and maybe analyse the 6 combinations: hateful amplifiers, hateful attenuators, hateful copycats, non-hateful amplifiers, non-hateful attenuators, non-hateful copycats and analyse their roles and behaviours in detail. Although the analysis in the current paper suggests that there might be few (if any) hateful attenuators.

6 CONCLUSION

We propose a new model for capturing the spread of toxicity in a social network, with two important differences from previous approaches: rather than considering tweets as hateful or non-hateful, we consider their toxicity (hatefulness) in the range [0-1], and do not label users as being hateful or non-hateful, either statically or dynamically. Through empirical analysis and observations, we classify users into three distinct categories: Amplifiers, Attenuators, and Copycats. This categorisation allows us to model the spread of toxicity more effectively by considering how users amplify, suppress, or mimic the hatefulness they receive. To validate the efficacy of our proposed model, we conduct experiments on both simulated Barabási–Albert (BA) graphs and a real-world dataset and our model successfully reproduces the increase in total toxicity and average toxicity observed in the empirical data.

This model also raises many new questions. If more relevant data becomes available, then one might ask:

- What is the effect of users entering and leaving the system? How do we model it?
- How consistent are the shifts applied by the users in the 3 categories? In this paper, we calculated the average shift in each category of users. Do we need a more refined picture? Does an amplifier(/attenuator/copycat) apply the same shift to all levels of toxicity?
- Is this behaviour of a user constant? Over time, can an attenuator become an amplifier?

We hope that future work will address these and related interesting questions.

REFERENCES

[1] Sai Saketh Aluru, Binny Mathew, Punyajoy Saha, and Animesh Mukherjee. 2020. Deep learning models for multilingual hate speech detection. *arXiv preprint arXiv:2004.06465* (2020).

[2] Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. In *WWW*. 759–760.

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

[4] John Cannarella and Joshua A Spechler. 2014. Epidemiological modeling of online social network dynamics. *arXiv preprint arXiv:1401.4208* (2014).

[5] Koustuv Dasgupta, Rahul Singh, Balaji Viswanathan, Dipanjan Chakraborty, Sougata Mukherjea, Amit A Nanavati, and Anupam Joshi. 2008. Social ties and their relevance to churn in mobile telecom networks. In *Proceedings of the 11th international conference on Extending database technology: Advances in database technology*. 668–677.

[6] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, Vol. 11. 512–515.

[7] Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. 2015. Hate speech detection with comment embeddings. In *WWW*. 29–30.

[8] Bojan Evkoski, Andraž Pelicon, Igor Mozetič, Nikola Ljubešić, and Petra Kralj Novak. 2022. Retweet communities reveal the main sources of hate speech. *Plos one* 17, 3 (2022), e0265602.

[9] Ling Feng, Yanqing Hu, Baowen Li, H Eugene Stanley, Shlomo Havlin, and Lidia A Braunstein. 2015. Competing for attention in social media under information overload conditions. *PloS one* 10, 7 (2015).

[10] Abdurahman Maarouf, Nicolas Pröllochs, and Stefan Feuerriegel. 2022. The Virality of Hate Speech on Social Media. *arXiv preprint arXiv:2210.13770* (2022).

[11] Frank J Massey Jr. 1951. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American statistical Association* 46, 253 (1951), 68–78.

[12] Binny Mathew, Ritam Dutt, Pawan Goyal, and Animesh Mukherjee. 2019. Spread of hate speech in online social media. In *Proceedings of the 10th ACM Conference on Web Science*. 173–182.

[13] Maya Mirchandani. 2018. Digital hatred, real violence: Majoritarian radicalisation and social media in India. *ORF Occasional Paper* 167 (2018), 1–30.

[14] Khouloud Mnassri, Praboda Rajapaksha, Reza Farahbakhsh, and Noel Crespi. 2022. BERT-based Ensemble Approaches for Hate Speech Detection. In *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 4649–4654.

[15] Seema Nagar, Sameer Gupta, C. S. Bahushruth, Ferdous Ahmed Barbhuiya, and Kuntal Dey. 2021. Homophily - a Driving Factor for Hate Speech on Twitter. In *Complex Networks & Their Applications X - Volume 2, Proceedings of the Tenth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2021, Madrid, Spain, November 30 - December 2, 2021 (Studies in Computational Intelligence, Vol. 1016)*, Rosa María Benito, Chantal Cherifi, Hocine Cherifi, Esteban Moro, Luis M. Rocha, and Marta Sales-Pardo (Eds.). Springer, 78–88. https://doi.org/10.1007/978-3-030-93413-2_7

[16] Seema Nagar, Sameer Gupta, Ferdous Ahmed Barbhuiya, and Kuntal Dey. 2022. Capturing the Spread of Hate on Twitter Using Spreading Activation Models. In *Complex Networks & Their Applications X: Volume 2, Proceedings of the Tenth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2021 10*. Springer, 15–27.

[17] Juan Manuel Pérez, Franco M Luque, Demian Zayat, Martín Kondratzky, Agustín Moro, Pablo Santiago Serrati, Joaquín Zajac, Paula Miguel, Natalia Debandi, Agustín Gravano, et al. 2023. Assessing the impact of contextual information in hate speech detection. *IEEE Access* 11 (2023), 30575–30590.

[18] Manoel Ribeiro, Pedro Calais, Yuri dos Santos, Virgilio Almeida, and Wagner Meira Jr. 2017. "Like Sheep Among Wolves": Characterizing Hateful Users on Twitter. *MIS2 Workshop at WSDM'2018* (2017).

[19] Manoel Horta Ribeiro, Pedro H Calais, Yuri A Santos, Virgílio AF Almeida, and Wagner Meira Jr. 2017. " Like Sheep Among Wolves": Characterizing Hateful Users on Twitter. *arXiv preprint arXiv:1801.00317* (2017).

[20] Punyajoy Saha, Kiran Garimella, Narla Komal Kalyan, Saurabh Kumar Pandey, Pauras Mangesh Meher, Binny Mathew, and Animesh Mukherjee. 2023. On the rise of fear speech in online social media. *Proceedings of the National Academy of Sciences* 120, 11 (2023), e2212270120.

[21] Punyajoy Saha, Binny Mathew, Kiran Garimella, and Animesh Mukherjee. 2021. "Short is the Road that Leads from Fear to Hate": Fear Speech in Indian WhatsApp Groups. In *Proceedings of the Web conference 2021*. 1110–1121.

[22] Kristin L Sainani. 2012. Dealing with non-normal data. *Pm&r* 4, 12 (2012), 1001–1005.

[23] Samuel Sanford Shapiro and Martin B Wilk. 1965. An analysis of variance test for normality (complete samples). *Biometrika* 52, 3/4 (1965), 591–611.

[24] Chao Wang, Xu-ying Yang, Ke Xu, and Jian-feng MA. 2014. SEIR-based model for the information spreading over SNS. *Acta Electonica Sinica* 42, 11 (2014), 2325.

[25] Qiyao Wang, Zhen Lin, Yuehui Jin, Shiduan Cheng, and Tan Yang. 2015. ESIS: emotion-based spreader–ignorant–stifler model for information diffusion. *Knowledge-based systems* 81 (2015), 46–55.

[26] William Warner and Julia Hirschberg. 2012. Detecting hate speech on the world wide web. In *Proceedings of the second workshop on language in social media*. 19–26.

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

[28] Ruzhi Xu, Heli Li, and Changming Xing. 2013. Research on information dissemination model for social networking services. *International Journal of Computer Science and Application (IJCSA)* 2, 1 (2013), 1–6.

[29] DING Xuejun. 2015. Research on propagation model of public opinion topics based on SCIR in microblogging. *Computer Engineering and Applications* 8 (2015), 6.

[30] Ziqi Zhang, David Robinson, and Jonathan Tepper. 2018. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15*. Springer, 745–760.