Modelling Harmful Content in a Social Network

Aatman Vaidya

Introduction

Understanding the spread of hate is crucial

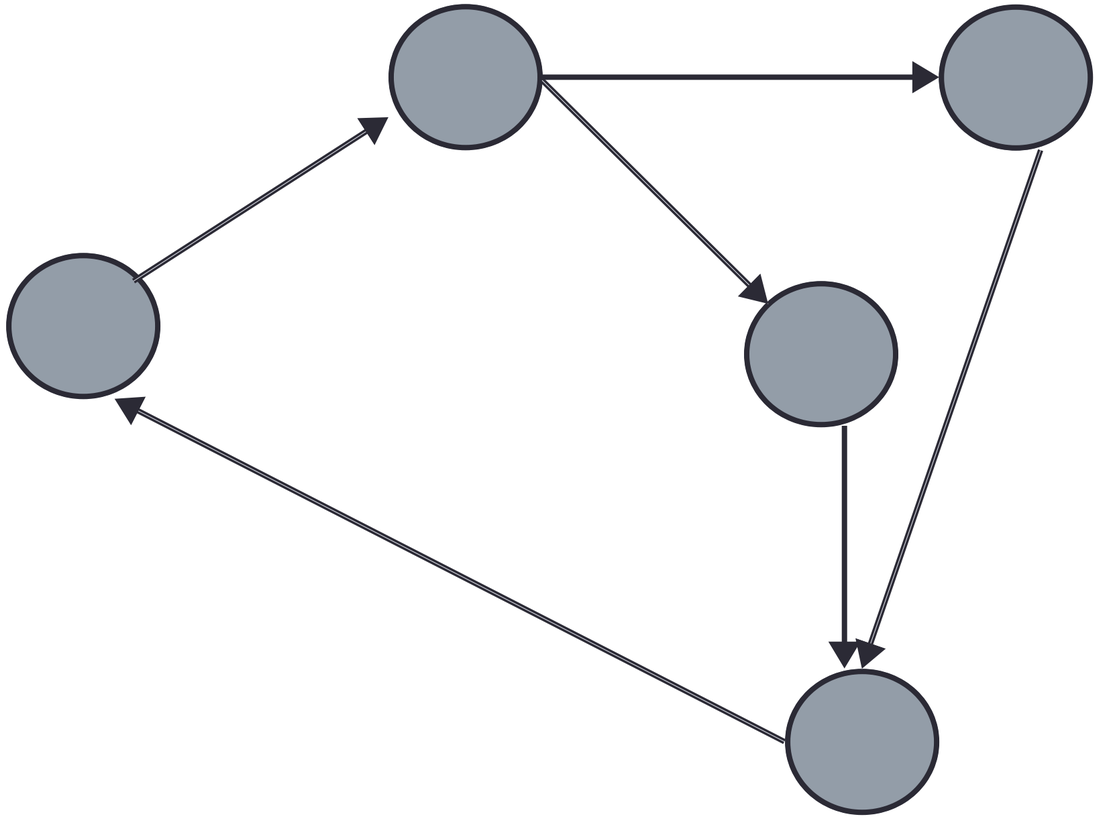
Expressions of hatred can even harm societies, peace and development, as it lays the ground for conflict, tension and human rights violations (UN Report)

What is "Toxicity"?

We refer to Hatefulness in the range [0-1] as **toxicity.**

Toxicity is defined as a "*rude, disrespectful, unreasonable comment that is likely to make someone leave a conversation*".

We want to understand how toxicity is spreading across the network and changing with time.



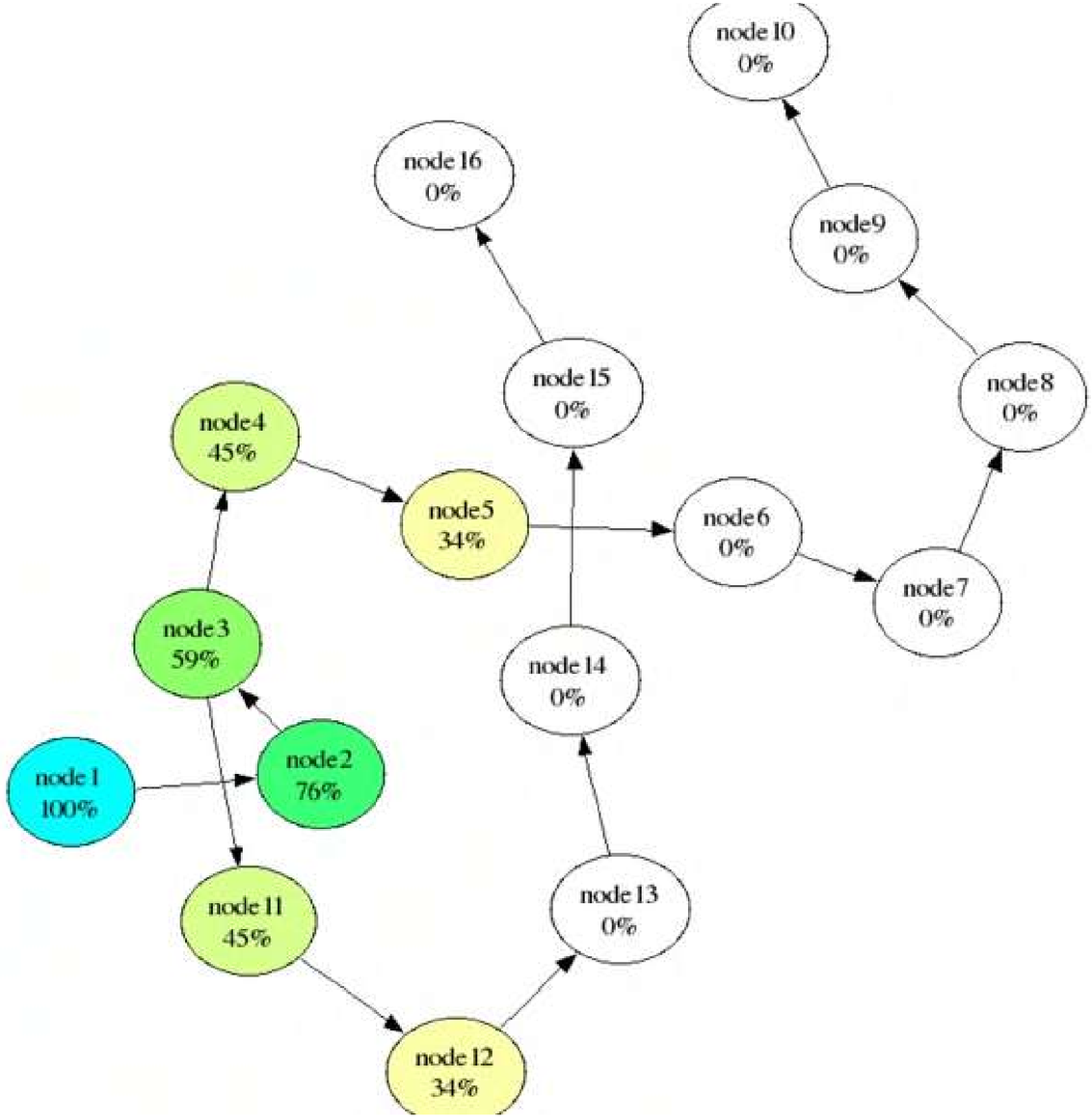
1. How are users responding to toxic tweets?
2. How is the toxicity changing over time?

Past work has modelled the spread of hate in following ways

Spread Activation (SPA) Models

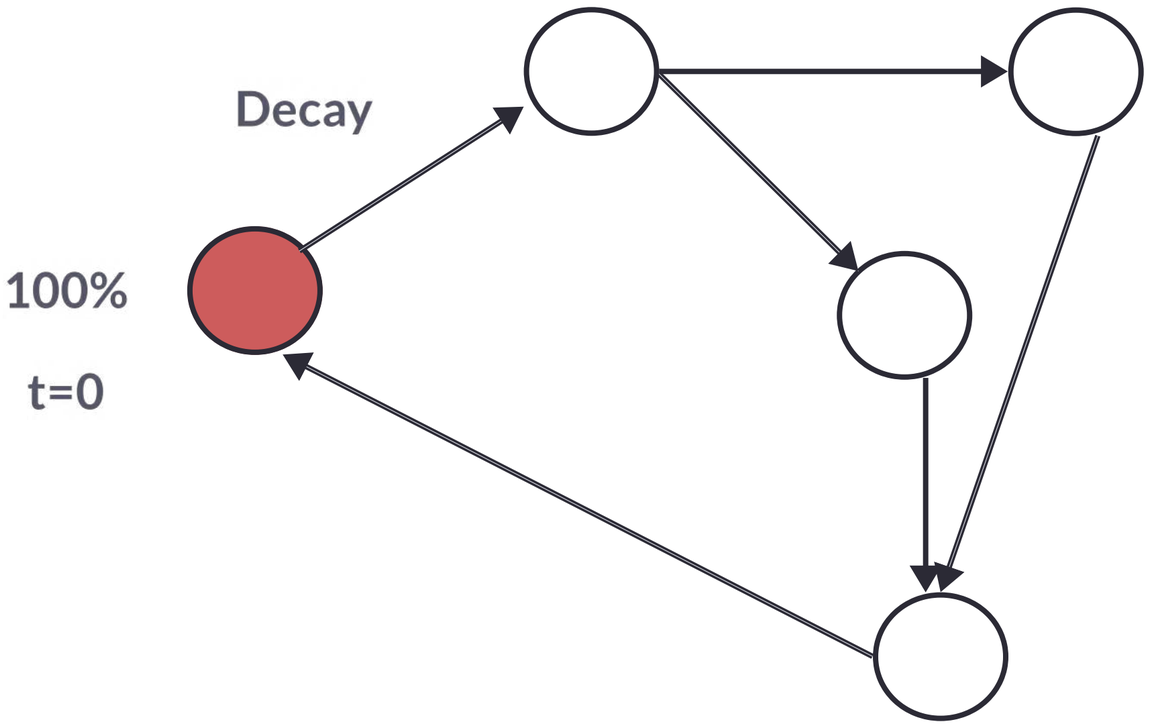
- Used to simulate **information** in a network
- Starts with a set of active nodes (seed nodes) each having initial **weights** or "**activation**".

Energy → E(X,0)



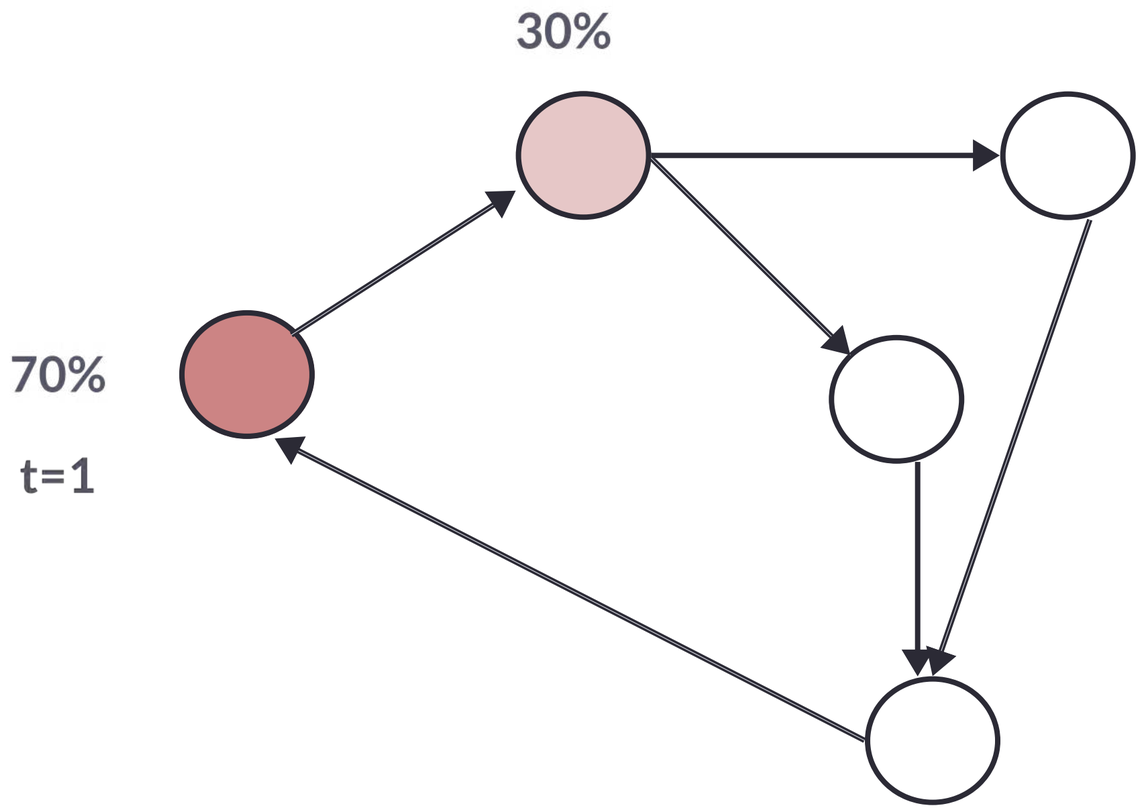
[1] - Koustuv Dasgupta, Rahul Singh, Balaji Viswanathan, Dipanjan Chakraborty, Sougata Mukherjea, Amit A Nanavati, and Anupam Joshi. 2008. Social ties and their relevance to churn in mobile telecom networks. In Proceedings of the 11th international conference on Extending database technology: Advances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

Example of SPA



Total Energy in the system = 100%

Example of SPA

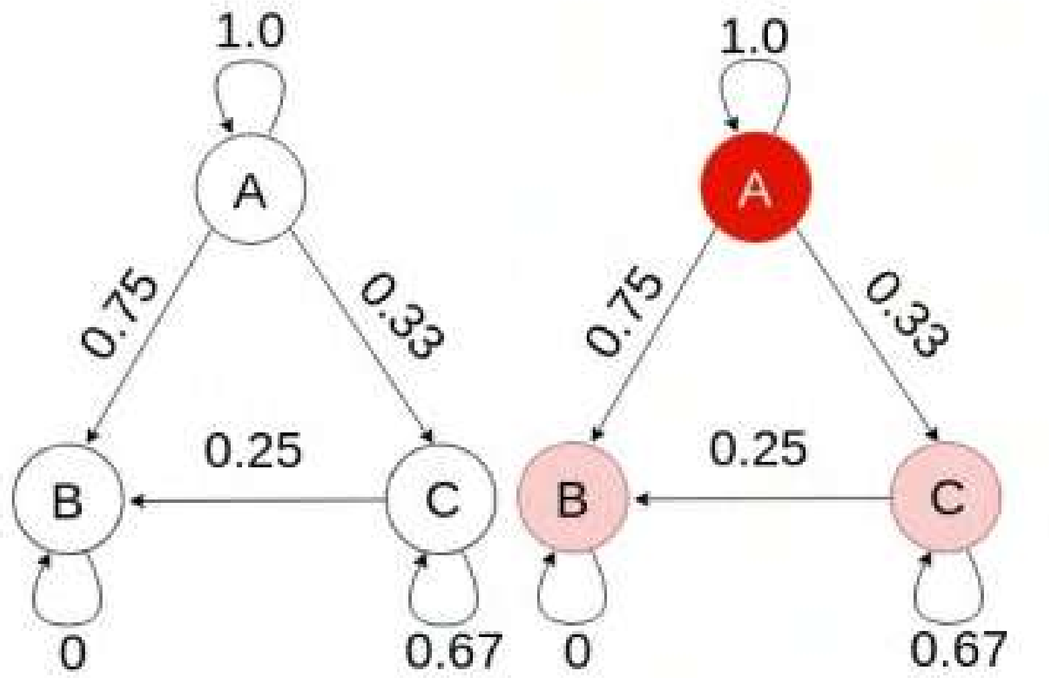


Total Energy in the system = 100%

- Active node transfers a portion of its energy ***d· E(X,i)*** to its neighbours
- while retaining **(1 − d) · E(X,i)** for itself.

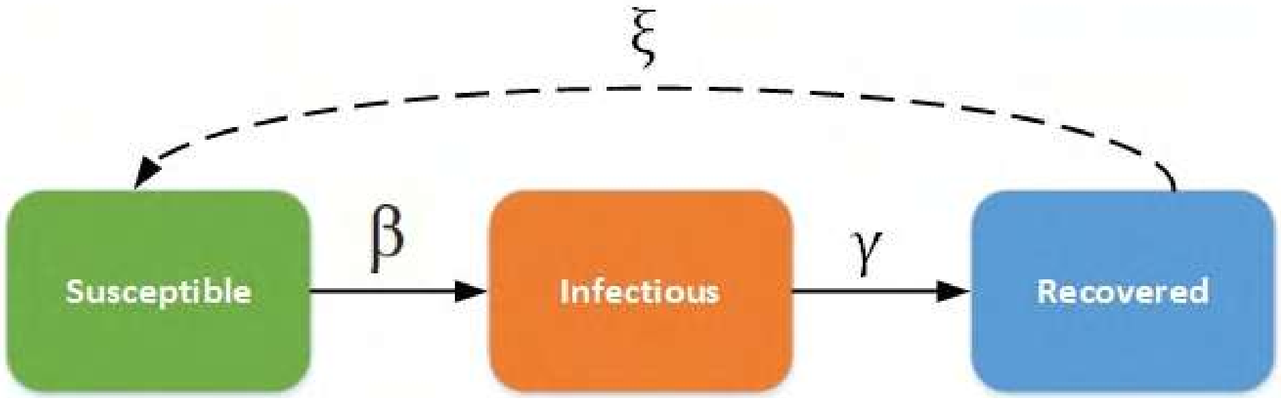
d = Spreading Factor

Energy in a system is always conserved.



Initial beliefs of A, B and C to be 1, 0 and 0

Susceptible-Infected-Recovered (SIR) models



- Compartmental models used in epidemiology [2]
- The models are most often run with **ordinary differential equations** (ODE).

$$\frac{dS}{dt} = -\frac{\beta SI}{N}$$
$$\frac{dI}{dt} = \frac{\beta SI}{N} - \gamma I$$
$$\frac{dR}{dt} = \gamma I$$

[2] - 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.

Dataset Description

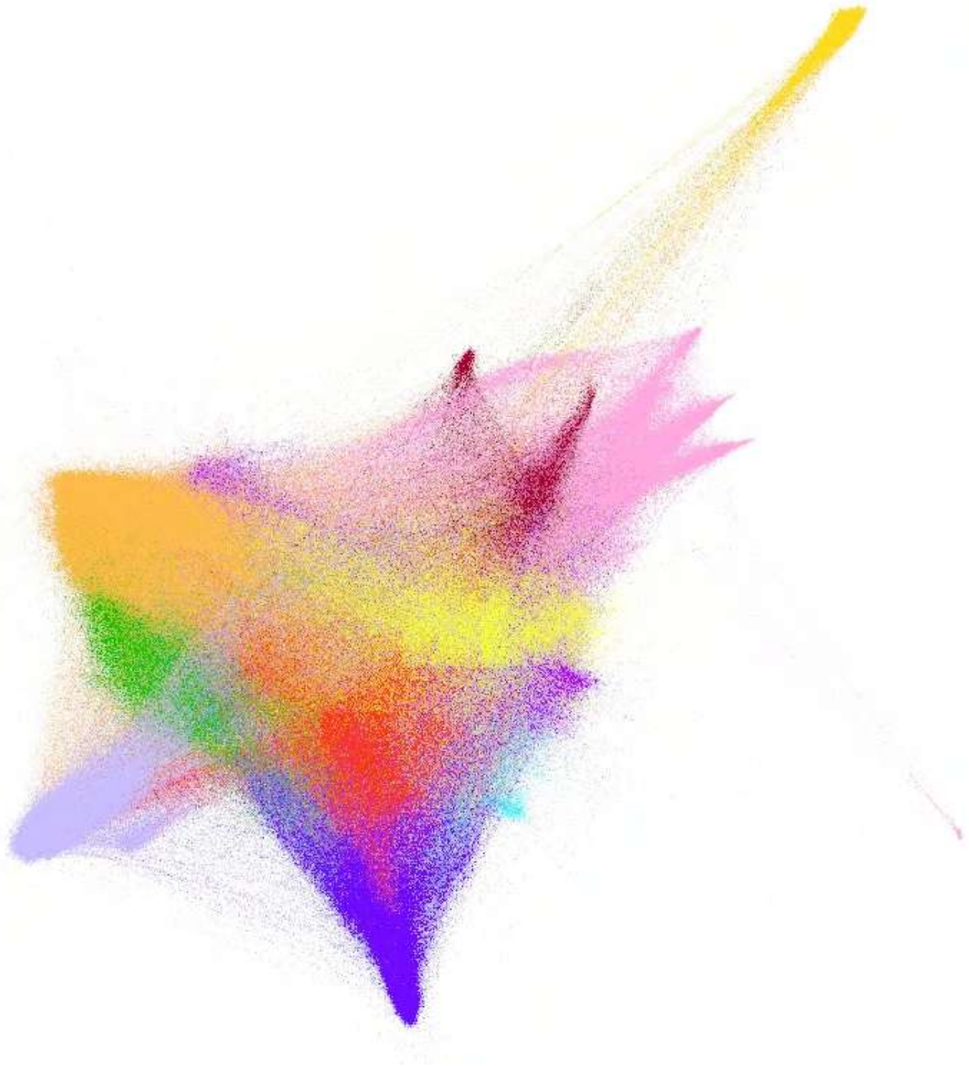
- Contains **100,386 users** and **19.58 million** tweets. [1]

- We use **Perspective API** [2] to assign a toxicity score to each tweet between **[0-1]**.

Directed retweet graph

$$G = (V, E),$$

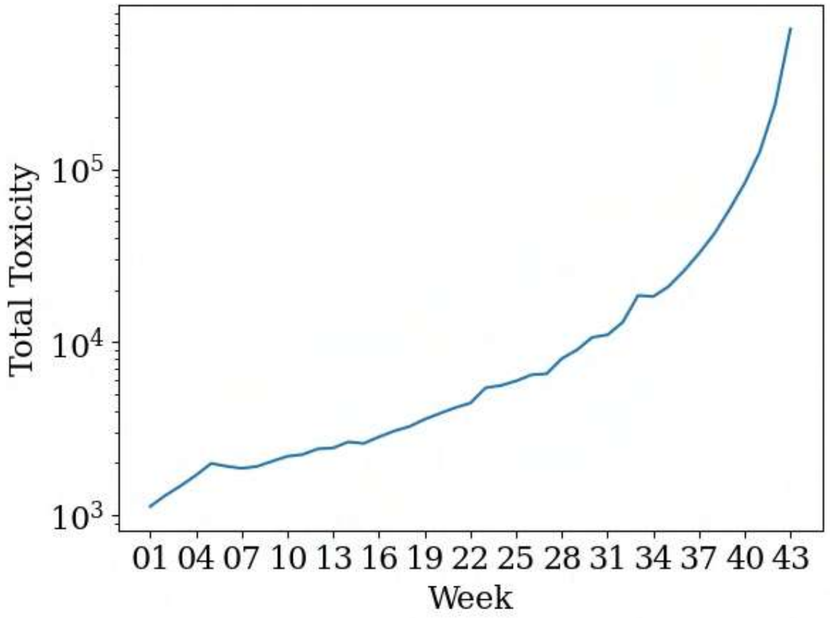
node $u \in V$,
edge $(u, v) \in E$ represents a retweet in the network



[1] - M. H. Ribeiro, P. H. Calais, Y. A. Santos, V. A. Almeida, and W. Meira Jr, "" like sheep among wolves": Characterizing hateful users on twitter," arXiv preprint arXiv:1801.00317, 2017.
[2] - https://perspectiveapi.com/

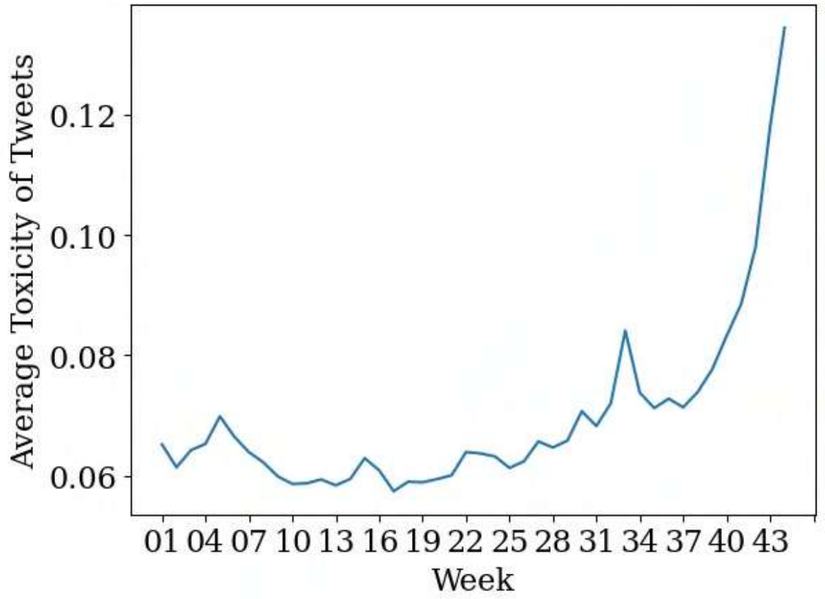
Analysing the Dataset

We begin by examining the temporal evolution of toxicity.



Total Toxicity Distribution over the Weeks.

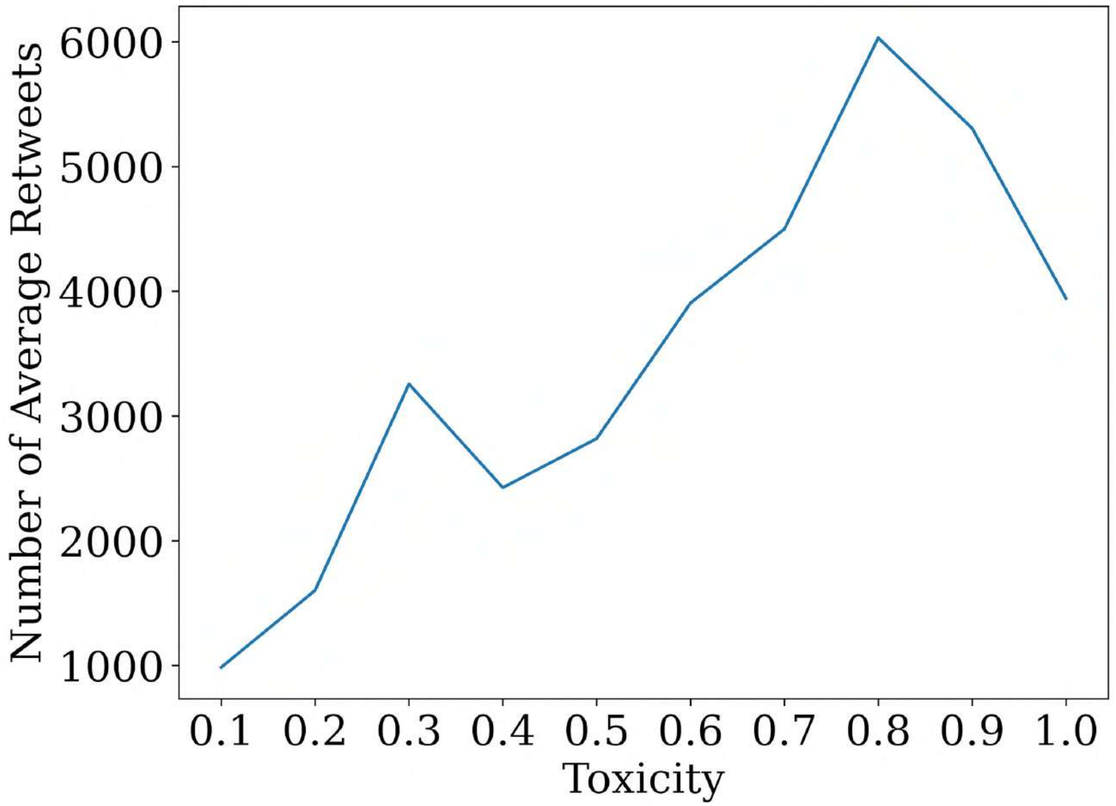
There is a rise in toxicity over time



Average Toxicity Distribution of Tweets Across Weeks.

There is a rise in average toxicity over time

"Neither the total toxicity nor its average is conserved in the network"



Average Number of Retweets by Toxicity

Tweets with higher toxicity values are retweeted significantly more than the tweets with low-toxicity values

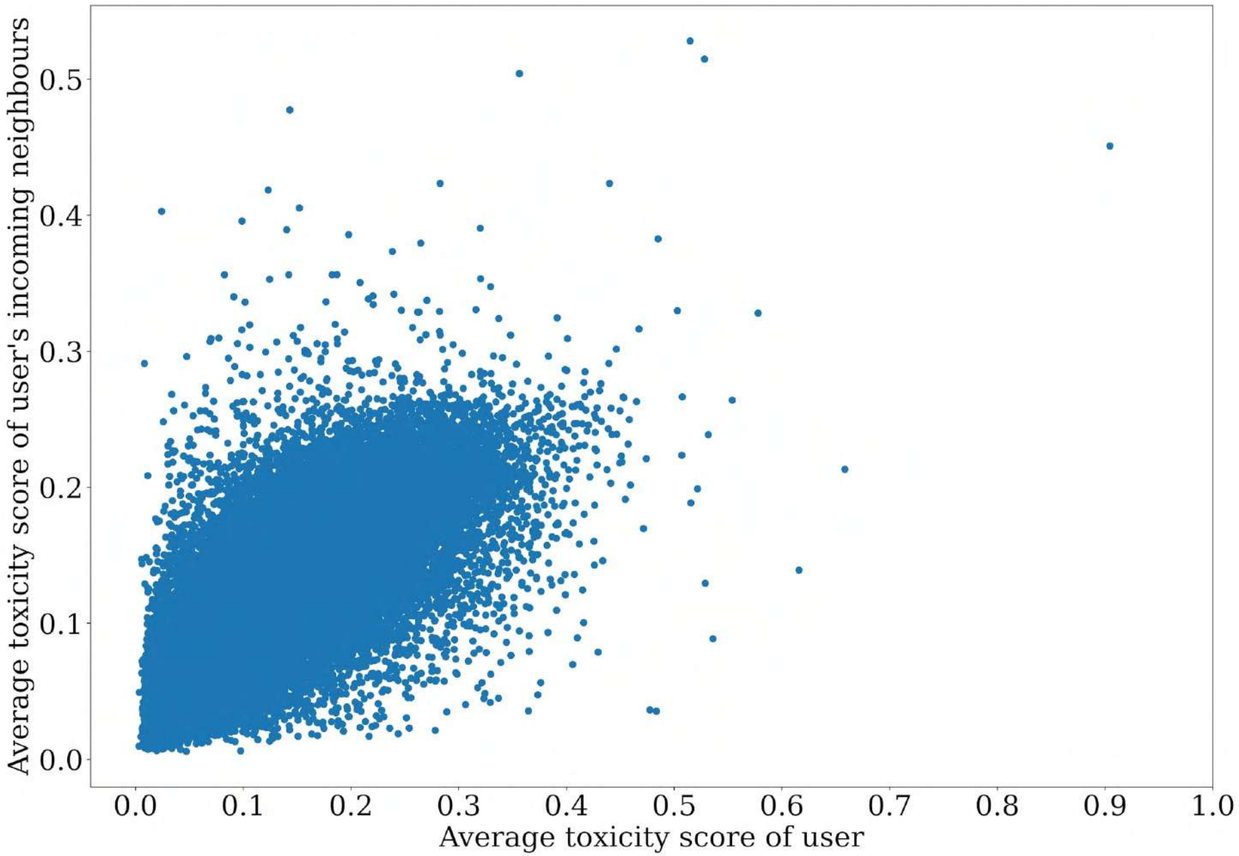
"Users respond differently to tweets of different toxicities"

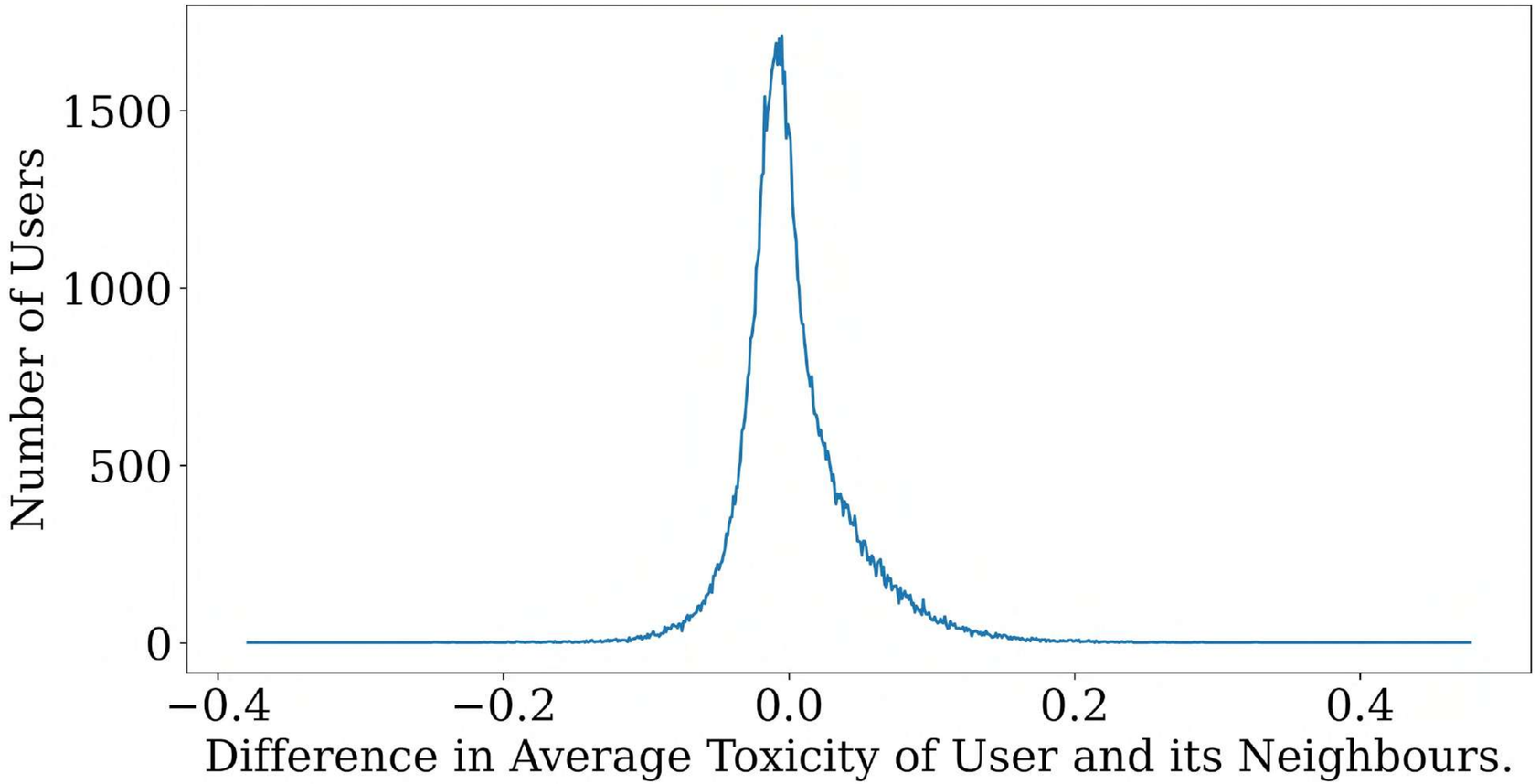
A Non-Conservational Approach for Modelling Toxicity Spread

User Classification

- We examine user behaviour in order to measure the *influence* of the social network upon a user and vice-versa.

- We compare a **user's average toxicity** with the **average toxicity of the user's indegree neighbours**.

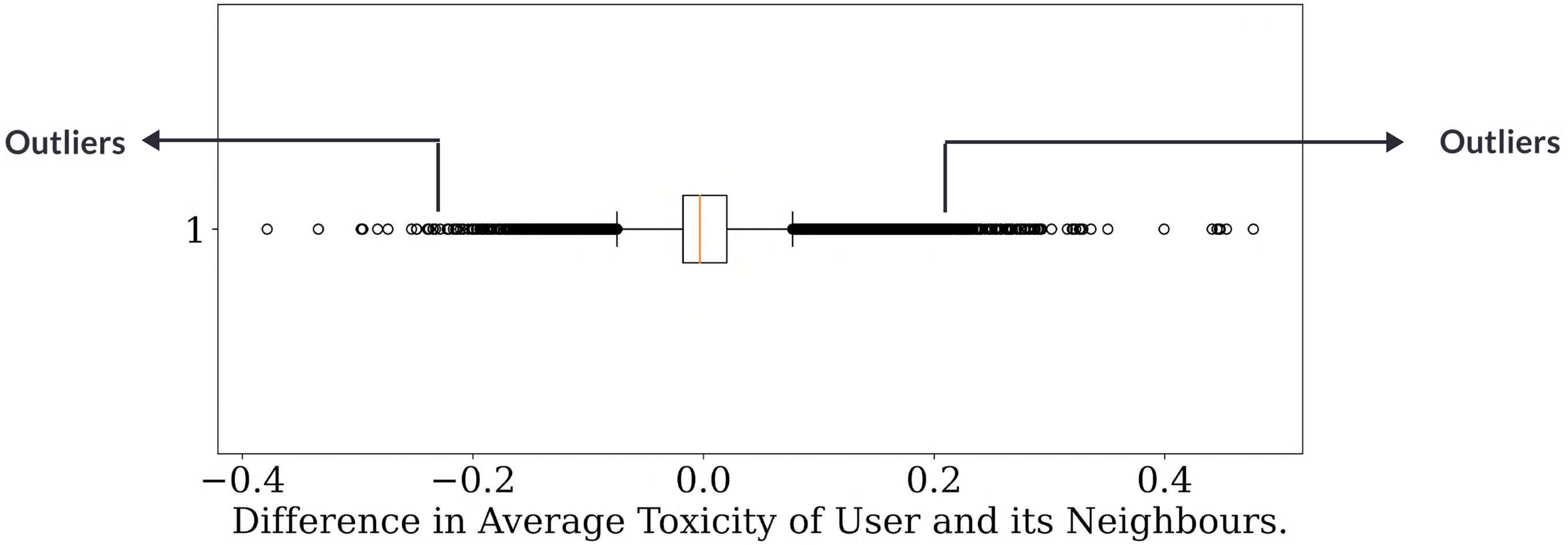




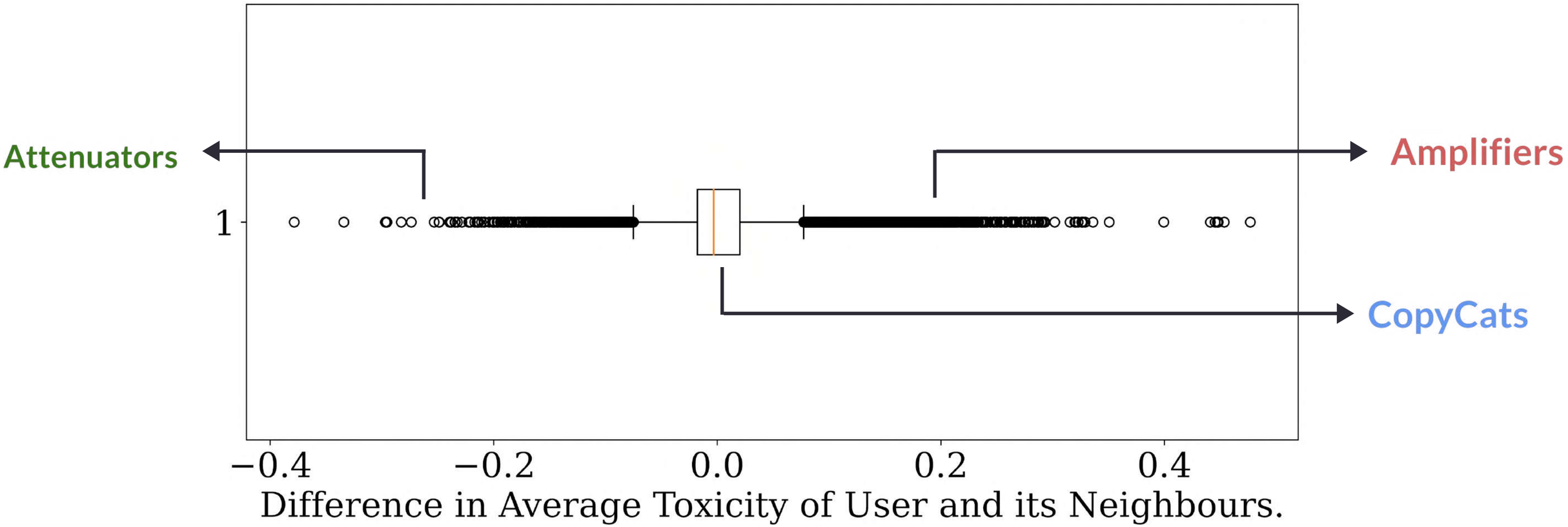
- For precisely measuring the influence → we calculated the **difference** between the **user's average toxicity** and the **average toxicity of all its indegree neighbours.**

- This distribution failed the Shapiro-Wilk and Kolmogorov-Smirnov normality tests → therefore is **not** a **normal distribution.**

- When distribution is not normal → **Interquartile Range (IQR)** helps us detect **outliers.**



- This approach naturally divides the set of **users** into **3 disjoint subsets**



Amplifiers - send out more toxicity than they receive

Attenuators - send out less toxicity than they receive

CopyCats - send out almost the same toxicity as they receive.

- We calculate average toxicity shifts for each user category.

- These **shifts** indicate the *average change* (shift) in incoming toxicity that the user will apply before transmitting the message further.

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

- The shifts are the average value of the difference distribution for each user category.

In order to define a model, we establish few assumptions

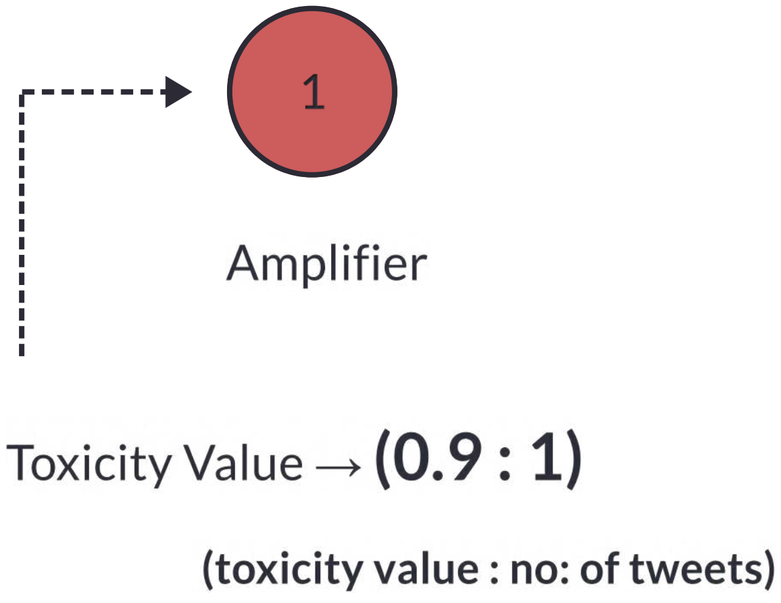
1) We consider the network to be static – users do not enter or leave the system.

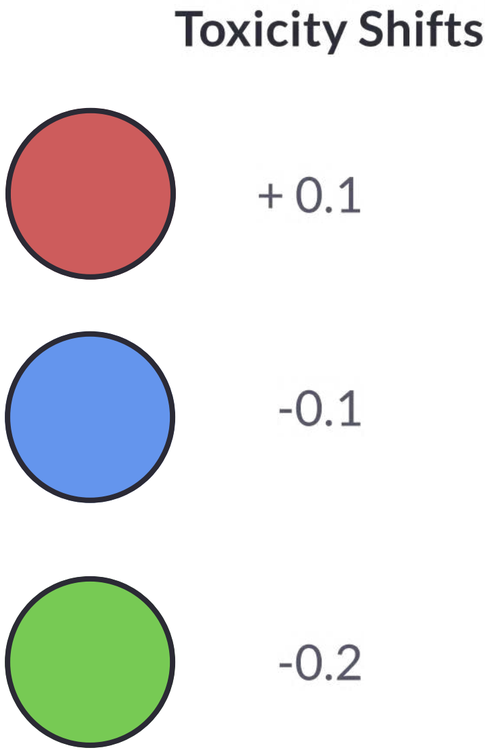
2) Rather than compute each user's toxicity shift for each range of toxicity, we consider only averages.

3) Rather than compute each user's toxicity shift for each range of toxicity, we consider only averages.

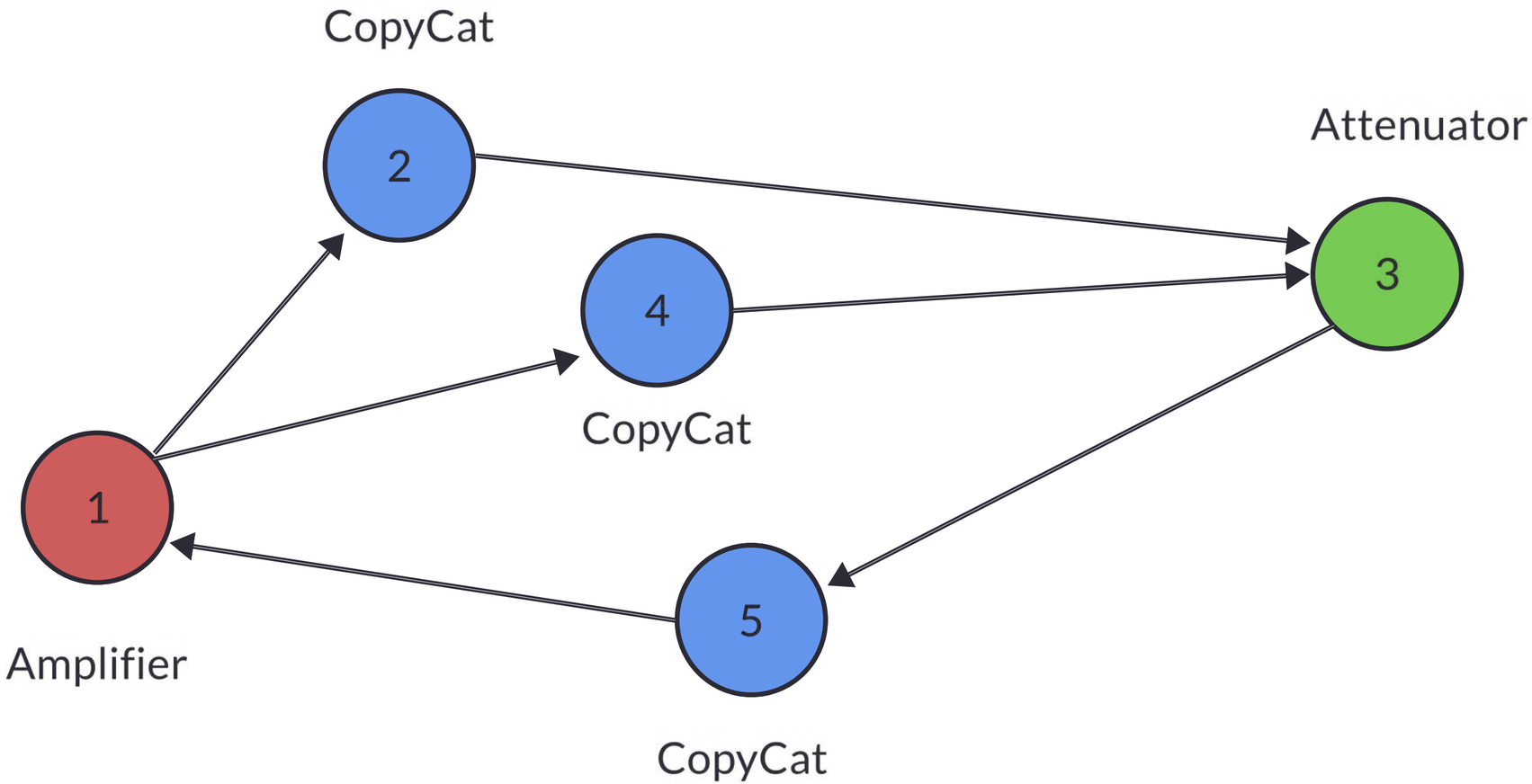
Proposed Model

- Demonstration on a small example graph.
- Initialize each node with a set of tweets and corresponding toxicity values.



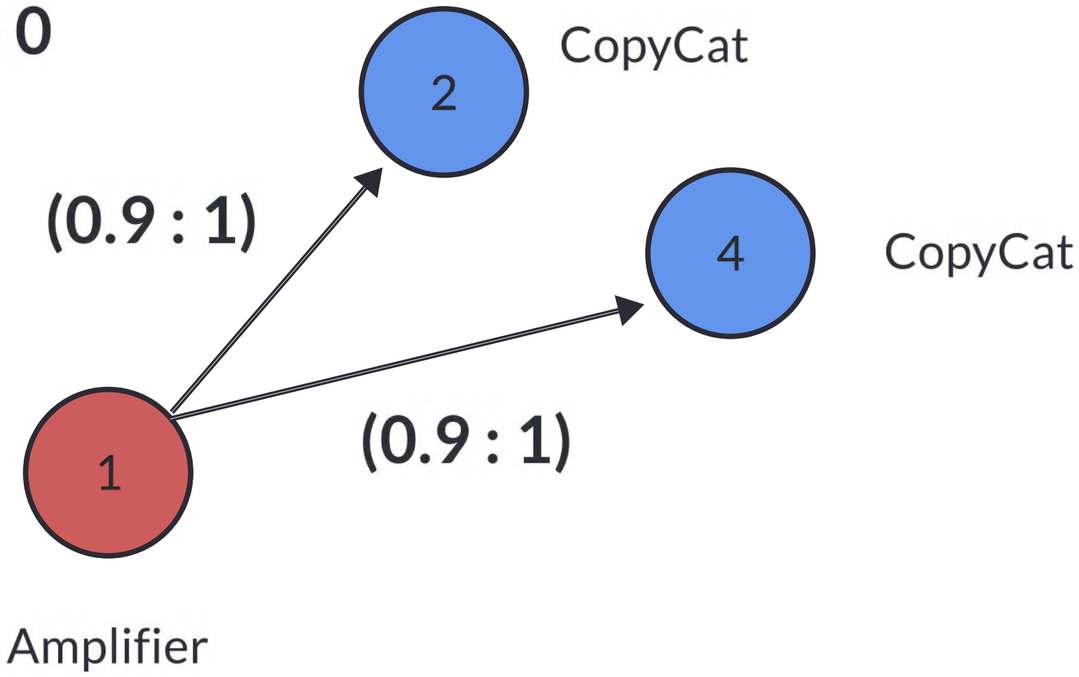


Simulation



Simulation

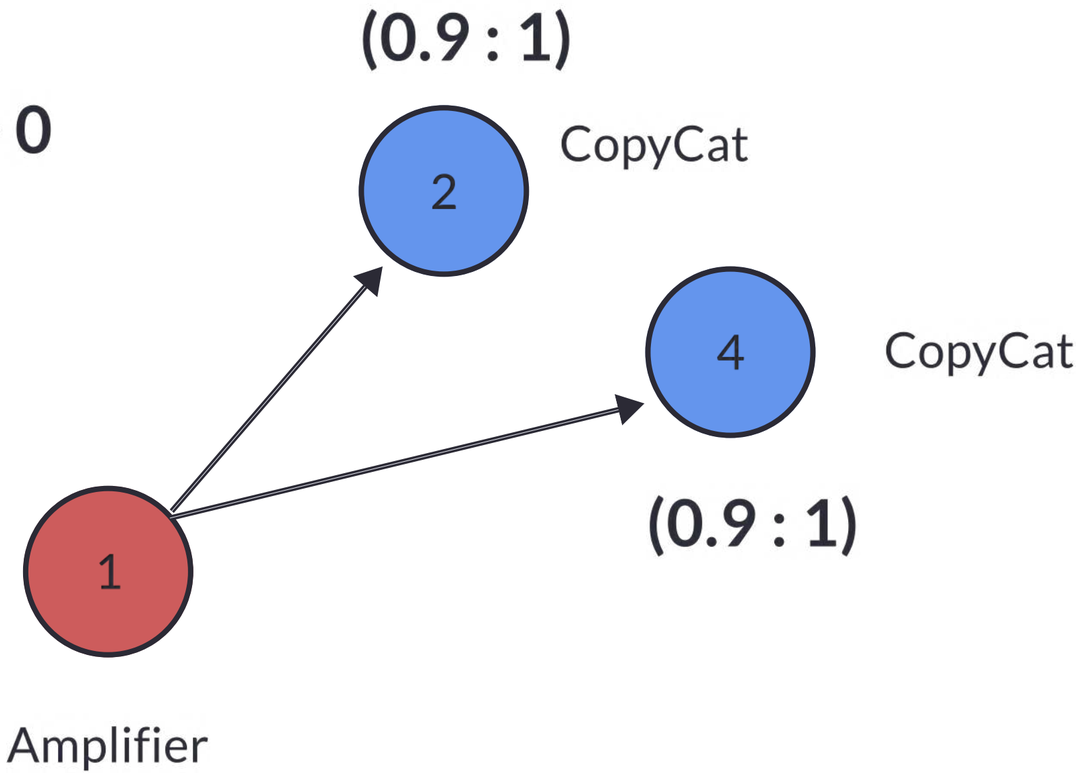
Time - 0



Toxicity Value → **(0.9 : 1)**

Simulation

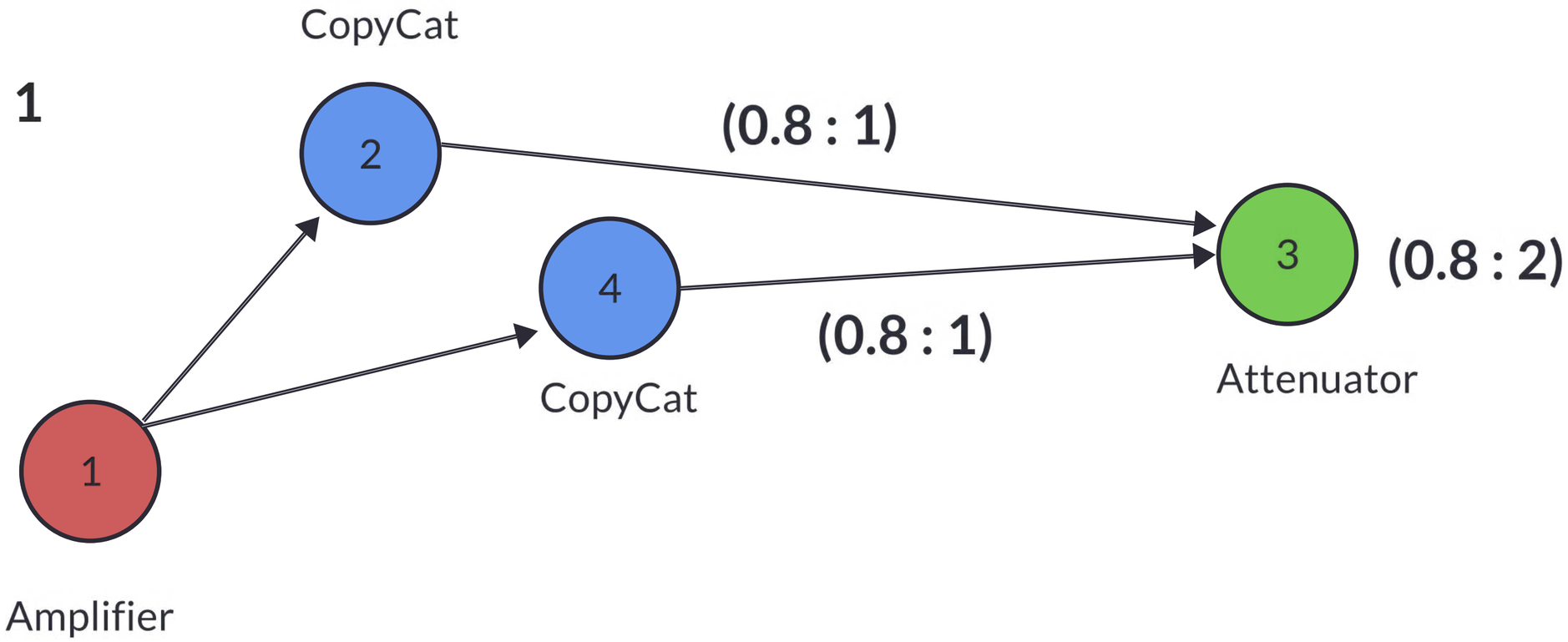
Time - 0



Total Toxicity in the network ⟶ 1.8

Simulation

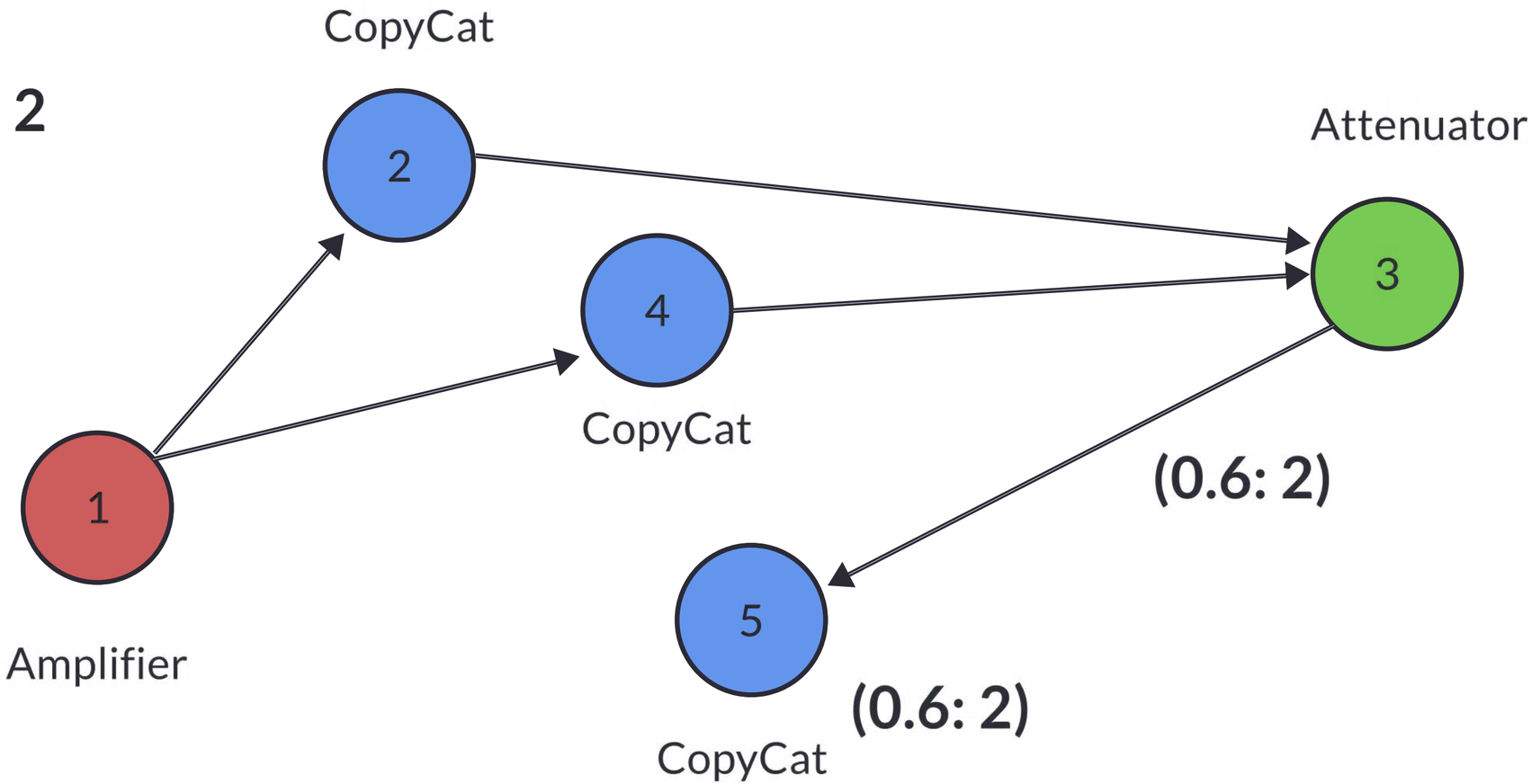
Time - 1



Total Toxicity in the network ⟶ 1.6

Simulation

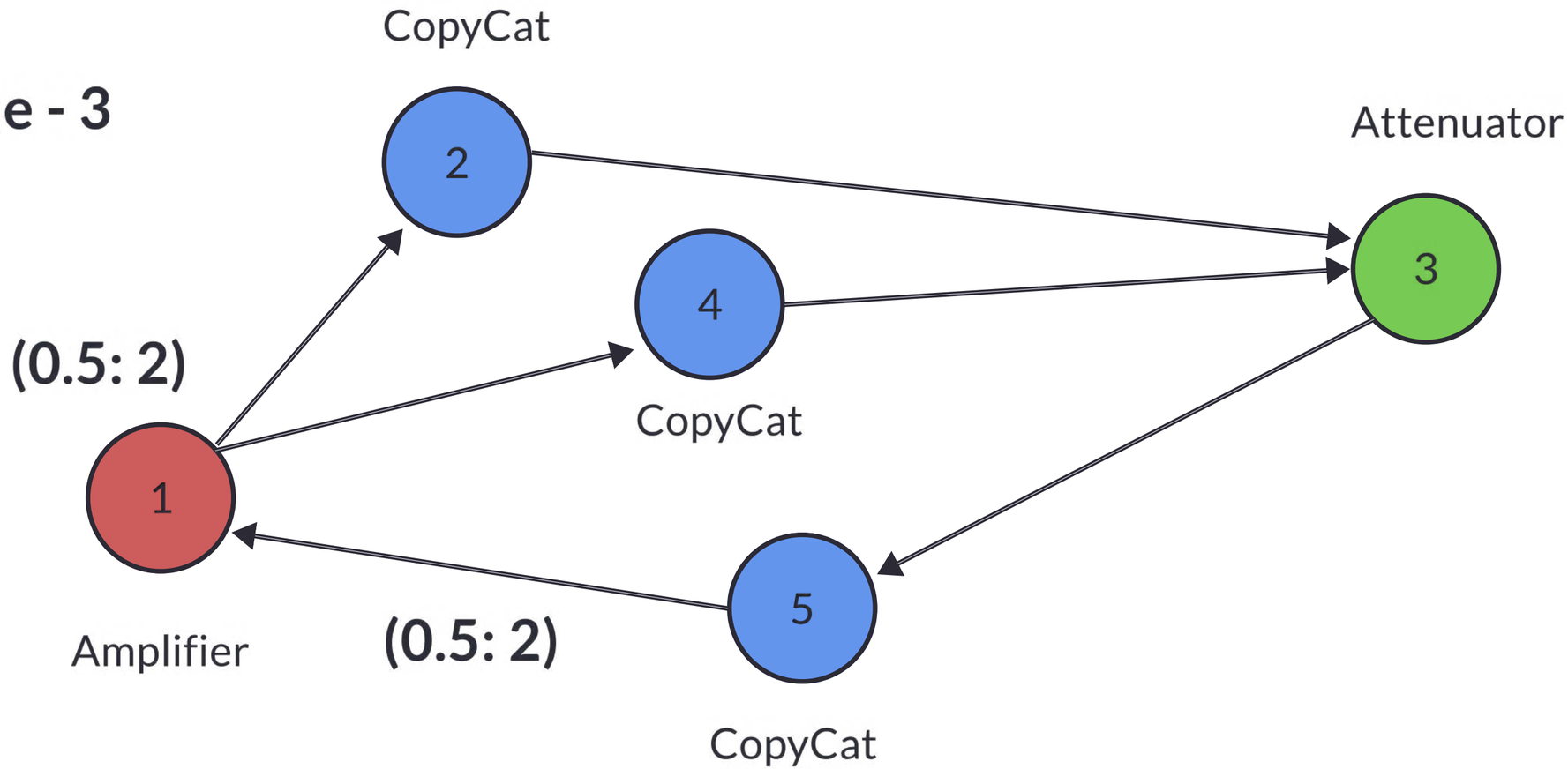
Time - 2



Total Toxicity in the network ⟶ 1.2

Simulation

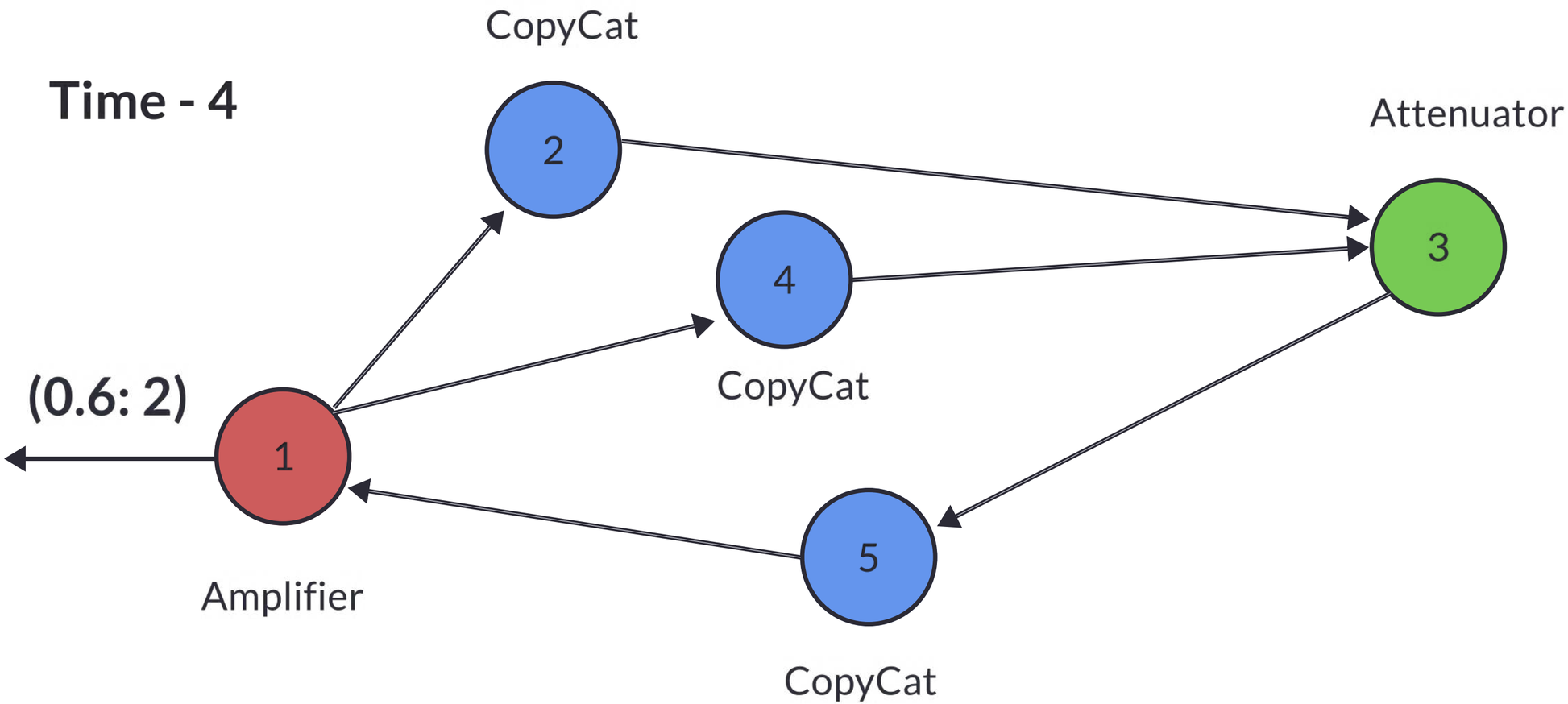
Time - 3



Total Toxicity in the network ⟶ 1.0

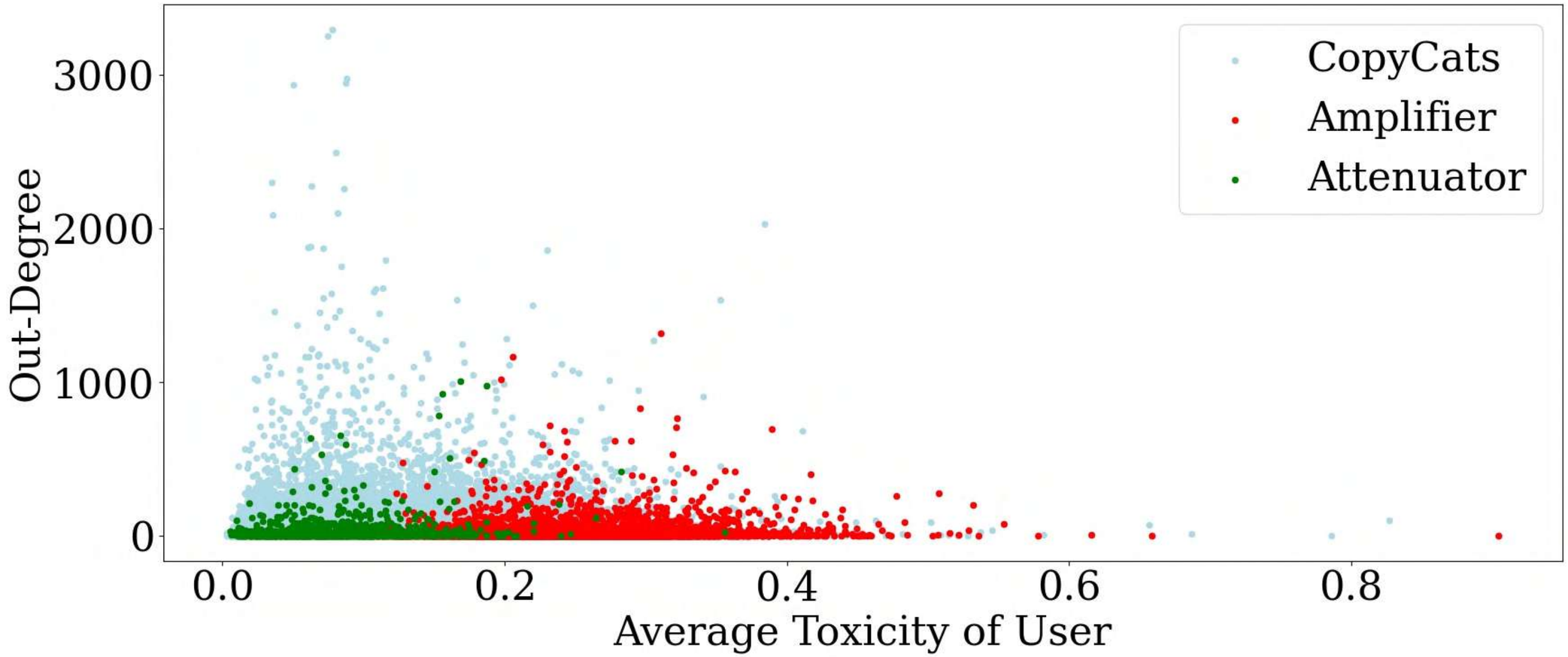
Simulation

Time - 4

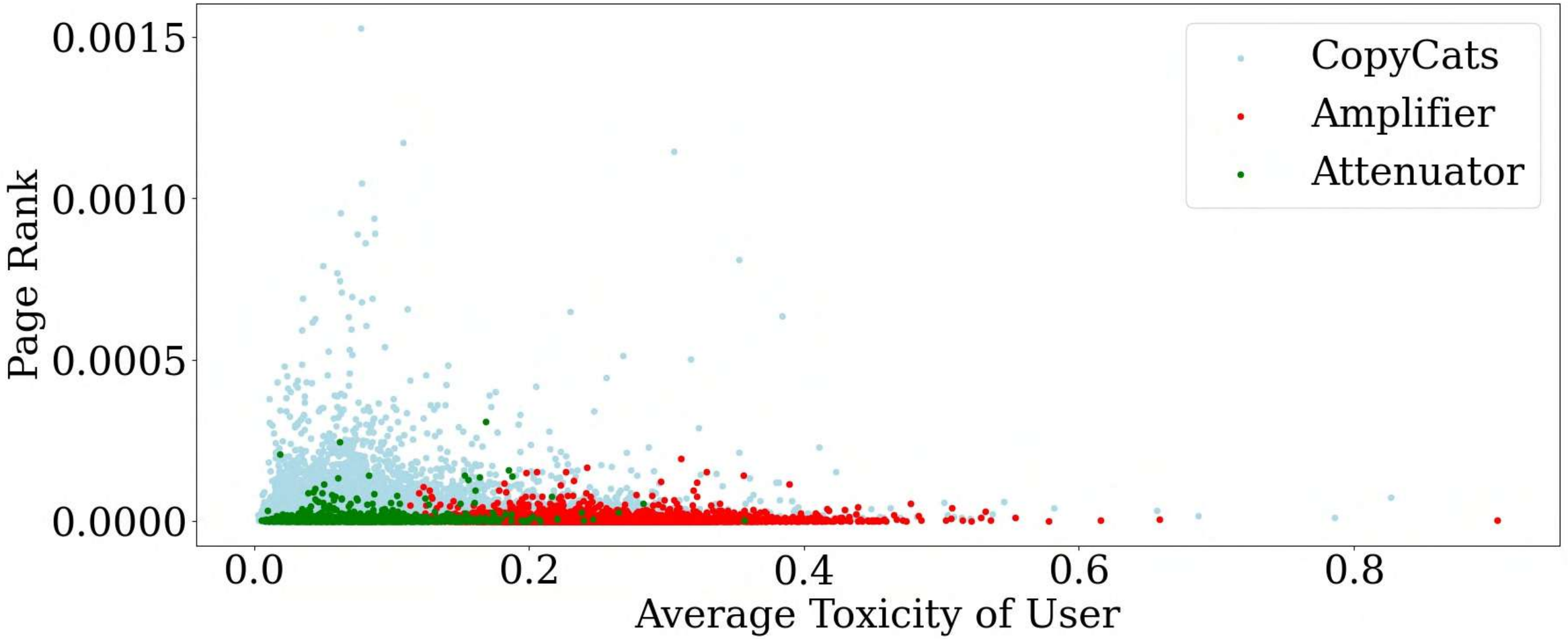


Total Toxicity in the network ⟶ 1.2

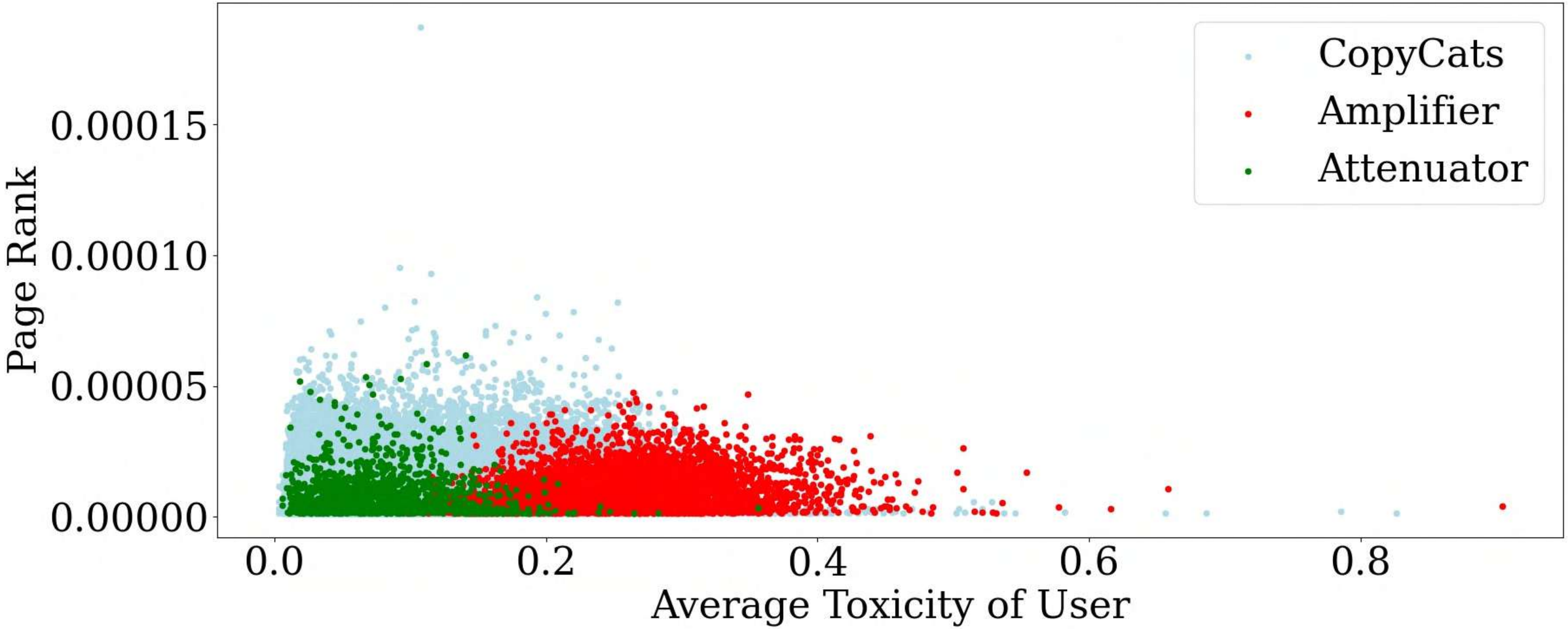
Amplifier, Attenuator and Copycat Characteristics



- Outdegrees of the **CopyCats** are among the highest.



- Hub Values - PageRank on outdegrees
- Highest hubvalues belong to the **CopyCats**.
- This shows the **dominating** role of CopyCats in the spread of toxicity.



- The highest pagerank values belong to the **copycats**, indicating their importance as recipients of links

Homophily

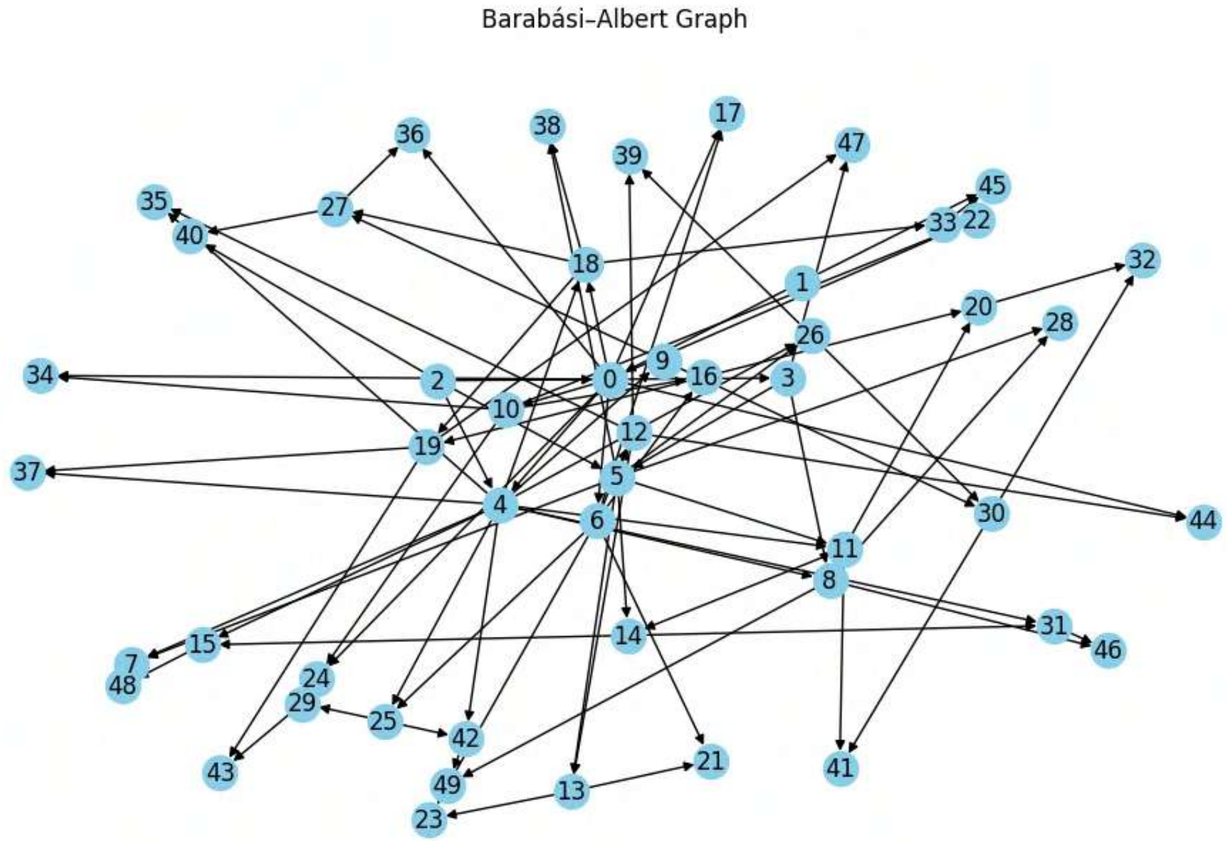
User Category	Attenuator	Amplifier	CopyCats
Attenuator	7.26e-18	0.02068	0.02844
Amplifier	0.02068	0	0.10937
CopyCats	0.02844	0.10937	0

Principle that we tend to be similar to our friends

This suggests that the amplifiers, attenuators and copycats, are almost randomly connected, without any preference or prejudice among each other

Experiments

- To validate our proposed model → we use Barabási–Albert Graphs.
- BA Graphs
  - resemble real-world networks,
  - rich-gets-richer phenomenon



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?

Results

- We wanted to understand the impact of the placement of these user types in the network.
- We devise five scenarios

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.

Model when simulated on BA Graphs

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	$\mathbf{3.87 \times 10^8}$	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	$\mathbf{2.52 \times 10^9}$	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	$\mathbf{3.25 \times 10^{10}}$	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}	$\mathbf{2.15 \times 10^{11}}$	4.86×10^{10}	1.5×10^{10}

Model when simulated on BA Graphs

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}

- The total toxicity increases with time and graph size.
- Case 3 has the highest total toxicity of them all, and Case 2 the lowest (even lower than the baseline)
- We see the role of Amplifiers and Attenuators in Case 2 and 3

Model when simulated on Subgraphs

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}

- We see a rise in Total Toxicity over time

We observe the following in summary

- Total toxicity rises in our model with random assignments of nodes to categories.
- The placement of the amplifiers, attenuators and copycats matter.
- Total toxicity of the subgraphs also rises.

Conclusion

- We propose a new model for capturing the spread of toxicity
- Toxicity exists on a spectrum (in the range from [0-1]). We do not label users as being hateful or non-hateful, either statically or dynamically.
- 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.
- our model successfully reproduces the increase in total toxicity and average toxicity observed in the empirical data.