SIMULATING MISINFORMATION: ANONYMITY AND ACCOUNTABILITY IN NETWORKS

ABSTRACT

This project presents an interactive tool for simulating and visualizing the spread of misinformation and disinformation in a social network, with a specific focus on the role of anonymity. The goal is to help explore how different types of users, anonymous, verified, and fact-checkers, influence the dynamics of information spreading.

The tool allows users to load or define a network, set behavioral parameters such as trust and tendency to share, and observe how misinformation propagates over time. Instead of producing a single outcome, the simulator is designed to support experimentation with different social settings and assumptions.

By combining visualization with quantitative metrics, the project aims to provide an intuitive way to study how changes in anonymity and trust can affect both the speed and the scale of misinformation spread.

1 Background and Motivation

Online anonymity plays an important role in digital communication. It can protect users, encourage self-expression, and allow people to participate in discussions without fear of personal consequences. At the same time, anonymity also makes it easier to misrepresent identity and spread unverified or false information. When users are not easily held accountable for what they share, the cost of spreading misleading content becomes lower, and harmful information can travel more freely through a network [1, 2].

Misinformation and disinformation are central to this problem. Misinformation refers to false information shared without the intention to deceive, while disinformation is created and shared deliberately to mislead. Both can appear in many forms on social media, including fake news, manipulated images, and misleading advertisements. These forms often mix real and false elements in ways that are difficult for users to distinguish [3].

These issues raise important questions about accountability. Systems that offer strong anonymity often struggle to ensure responsible behavior, while systems that enforce identity may discourage participation or put vulnerable users at risk. This creates a tension between anonymity and controllability. Several research efforts have explored technical solutions that aim to balance anonymity with accountability [4, 5, 6], but a fundamental question remains open: how can the impact of these design choices be measured in practice, especially in the context of information spreading?

This project is motivated by the need for a simple and flexible way to study this trade-off. Instead of proposing a new anonymity system, the goal is to provide a simulation and visualization tool that allows researchers and users to experiment with different assumptions about trust, sharing behavior, and identity. By modeling anonymous users, verified users, and fact-checkers as different types of nodes, the tool makes it possible to observe how misinformation spreads under different social conditions.

Through this approach, the project supports exploration rather than prediction. Users can change parameters, compare scenarios, and examine how small changes in trust or tendency to share can lead to different outcomes over time. In this way, the simulator can be used as a practical framework for reasoning about anonymity, accountability, and misinformation in social networks.

2 System Overview and Parameters

In this tool, the user provides a model of a social network along with with a set of parameters that describe how information spreads through it. The simulation is then executed for a chosen number of timesteps, and the results are shown both visually and as exportable tables. At each timestep, the user can observe which nodes are infected and how the spread evolves over time.

The tool is implemented in Python. NetworkX is used to represent the social network and compute the spread of misinformation, Streamlit provides the interactive interface and PyVis is used for network visualization.

2.1 Node attributes

Each node in the network represents a user and is described by several attributes.

Identity type. *Verified* or *Anonymous*

Credulity. The probability that a node believes and accepts incoming misinformation. A higher credulity means the node is more likely to become infected when exposed to misleading content.

Tendency to share. The probability that a node will propagate misinformation after it has already been infected. It models how actively a user spreads information once they believe it.

Fact checker. A boolean attribute. Fact checker nodes represent entities such as professional verifiers or moderation systems. These nodes never accept misinformation and therefore never become infected or propagate it further.

2.2 Edge attributes

Edges represent communication links between users.

Trust weight. A value between 0 and 1. This value represents how much a node trusts information coming from a specific neighbor. A higher trust weight increases the probability that misinformation transmitted along that edge will be accepted.

2.3 Global anonymity and trust parameters

In addition to node- and edge-specific attributes, users can also set a few global parameters for the analysis. The first group of these parameters describes how anonymity affects behavior in the network: the tendency of a node to share information, and the level of trust between neighboring nodes.

Anonymity is known to increase the probability that a node will spread information. This can happen either with the intention of misleading others, or simply because anonymous users feel less accountable and spend less time verifying what they share.

In the same way, trust between nodes is influenced by whether a neighbor is verified or anonymous. A node is expected to trust a verified neighbor more than an anonymous one, since verified users are more likely to be cautious when spreading information under their real identity.

For this reason, instead of forcing the user to manually set these values for every node and edge, the system allows them to define default parameters for all anonymous nodes and all verified nodes. These are given by four parameters: the tendency to share for anonymous nodes, the tendency to share for verified nodes, the trust in anonymous neighbors, and the trust in verified neighbors.

2.4 Initial infected nodes and simulation time

To start the simulation, the user must select a set of initially infected nodes that will introduce the first piece of misinformation into the network. Fact checker nodes are excluded from this selection, since they never accept or spread misinformation. The user may choose the initial nodes manually or request that a given number of them be selected at random from the eligible nodes.

Finally, the user specifies the number of timesteps for which the simulation will run. At each timestep, infected nodes attempt to influence their uninfected neighbors according to the defined probabilities. This process continues until the simulation reaches the chosen time limit, producing a time series of newly infected and total infected nodes.

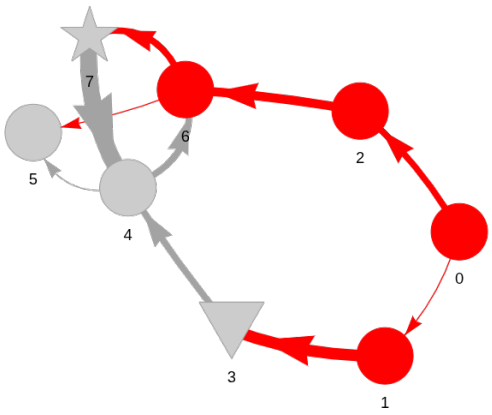

Figure 1: A sample social network used in the simulation.

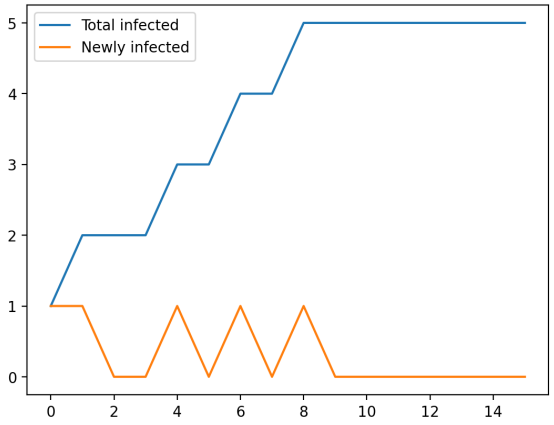

Figure 2: New and total infected nodes over time.

2.5 Outputs

After the simulation is executed, the tool provides several forms of output that allow the user to analyze the spread of misinformation over time.

First, an interactive visualization of the network is displayed for each timestep. A sample visualization is shown in Figure 1. In this view, nodes are colored according to their infection status, and different shapes are used to distinguish verified users, anonymous users, and fact checker nodes. The user can move forward and backward through timesteps in order to examine how the spread evolves and which parts of the network are affected at each stage.

In addition to the network visualization, the tool generates a plot showing the number of newly infected nodes and the total number of infected nodes at every timestep. An example of this plot is provided in Figure 2. This representation makes it easier to observe overall trends such as rapid growth in the spread.

Finally, the simulation results can be exported as tables for further analysis. These include a table listing the final set of infected nodes, a table indicating the timestep at which each node became infected, and a time series table reporting the number of newly infected and total infected nodes at each timestep. These outputs allow the user to study the results outside the visualization environment and compare different simulation scenarios on a large scale.

3 Simulation Logic and Infection Probability Model

At each timestep, the simulation updates the state of the network by determining which uninfected nodes become infected based on their interactions with already infected neighbors.

For every infected node, the model considers its outgoing connections and evaluates the possibility of spreading misinformation to neighboring nodes. The likelihood of transmission depends on three main factors: the credulity of the receiving node, the tendency of the infected node to share information, and the level of trust between the two nodes. These three values are combined to produce a probability that represents the chance of exposure through that specific connection.

A node may receive exposure from multiple infected neighbors during the same timestep. Instead of treating these exposures independently, the model combines them into a single overall probability of infection. In this way, repeated exposure increases the likelihood that a node becomes infected, reflecting the reinforcing effect of seeing the same misinformation from multiple sources.

Once this total probability is computed for each uninfected node, a random sampling is used to decide if the node becomes infected at that timestep. This introduces a stochastic element into the simulation, allowing different runs with the same parameters to produce different outcomes and better reflect the uncertainty present in real social systems.

Nodes that become infected are marked accordingly and added to the set of infected nodes for the next timestep. This process repeats until the specified number of timesteps is reached. Over time, the model produces a history of how misinformation spreads through the network and how quickly different parts of the network are affected.

This approach allows the simulation to capture both local interactions between neighboring users and global patterns of spread, while remaining simple enough to be interpreted and adjusted through user-defined parameters.

4 Code Structure

The implementation of the simulator is divided into two main modules: `main.py`, responsible for the core simulation logic and data handling, and `app.py` for the interactive user interface and visualization.

The module `main.py` represents the social network, runs the misinformation spreading process, and produces numerical results.

The file `app.py` implements the interactive interface of the tool using Streamlit. Its main responsibility is to allow users to configure parameters, run the simulation, and explore the results visually. For visualization, the network is converted into a PyVis graph.

By separating the simulation logic from the user interface, the system becomes easier to read, modify, and extend with new features.

5 Future extensions

Currently, the tool provides the core features needed to analyze how misinformation spreads in a network. There are many directions in which it could be extended. For example, one could model platform-level or government-level policies in more detail than the simple blocker nodes that currently exist. Another extension could be to support automated generation or visualization of networks, making it easier to test different scenarios.

So far, I have focused on small to medium networks that are easy to visualize and experiment with. I have not tested the tool on very large networks, so it is unknown how well it performs in terms of speed or memory usage when the network grows. Evaluating its behavior on bigger networks could be a useful next step for anyone interested in scaling up the analysis.

Because the simulator models the main effects of anonymity, trust, and fact-checking, it gives a starting point for researchers and practitioners to try out interventions and policies. Before this, it was difficult to see how proposed solutions for anonymity and accountability would actually work. Based on this tool, users can modify the model according to their designs and watch how these ideas change the spread of misinformation in the network.

Artifact Availability

The simulation tool, along with example networks used in this study, is available online: Link.

Acknowledgments

Parts of the final text were refined and smoothed with the assistance of an AI language model.

References

[1] Brynn Comes. Blurring the line: Digital anonymity, deception, and the crisis of online authenticity. *Advances in Social Sciences Research Journal*, 12(10):131–135, Oct. 2025.

[2] Jennifer McArthur, Zoë Dunsworth, and Marguerite Ternes. World wide web of lies: Personality and online deception. *Telematics and Informatics Reports*, 11:100075, 2023.

[3] Kai Shu, Amrita Bhattacharjee, Faisal Alatawi, Tahora H. Nazer, Kaize Ding, Mansooreh Karami, and Huan Liu. Combating disinformation in a social media age. *WIREs Data Mining and Knowledge Discovery*, 10(6):e1385, 2020.

[4] Csilla Farkas, Gábor Ziegler, Attila Meretei, and András Lörincz. Anonymity and accountability in self-organizing electronic communities. In *Proceedings of the 2002 ACM workshop on Privacy in the Electronic Society*, pages 81–90. ACM.

[5] Justin Brickell and Vitaly Shmatikov. Efficient anonymity-preserving data collection. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 76–85. ACM.

[6] Vanesa Daza, Abida Haque, Alessandra Scafuro, Alexandros Zacharakis, and Arantxa Zapico. Mutual accountability layer: Accountable anonymity within accountable trust. Publication info: Published elsewhere. Minor revision. CSCML-2022.