A Differential and Graph-Based Stochastic Rumour Spread Model for Social Networks

Matthew Reyes
matt.n.reyes@gmail.com

1 Introduction

In the digital age, information can disseminated at an unprecedented pace. Of particular interest is understanding the spread of *rumours*, defined by the Oxford Dictionary as "a circulating story or report of uncertain truth" [1].

For example, when the Associated Press Twitter account was hacked, tweeting *"Breaking: Two Explosions in the White House and Barack Obama is injured."*, many believed it, including the financial markets. The Dow Jones industrial average plunged more than 128 points in seconds after the report [6].

The spreading of rumours across a population bear many similarities to epidemics, an idea first introduced by Daley and Kendall [3]. Rumours can also take the form of "fake news" and viral marketing, where an actor wishes to influence (or "infect") a population with disinformation or an advertising message. Other rumour-like mechanisms include the spread of information across a decentralized network - gossip algorithms are applied in peer-to-peer file-sharing and crypto-currency networks [9].

The objective of this project is to study the dynamics of rumour spread on a social network. We will study the spread of rumours in a population using two approaches: first using an SIR model formulation then modifying it to apply to a social network dataset.

2 An SIR Model

2.1 Methodology

Our model is based off of a modified SIR rumour-spread model introduced by Zhao et al. [10]. In this model, we have a population of size N. With respect to the rumour, there are three distinct groups: *ignorants*, *spreaders*, and *stiflers*. This model uses the following set of rules and assumptions:

- Homogenous mixing of the population. In the context of a complex network, this would mean a fully connected social graph.
- When a spreader contacts an ignorant, the ignorant accepts the rumour with probability λ and becomes a spreader. Otherwise, they become a stifler.
- A spreader can spontaneously lose interest or forget about the rumour with probability δ. This also captures situations where a user doesn't share the post due to inactivity.
- If a spreader contacts a stifler, they become a stifler with probability η. This captures the stifler's influence on their peers to stop spreading a rumour.
- If a spreader contacts another spreader, the initial spreader becomes a stifler with probability γ. This models the possibility that the spreader sees another spreader's take on the rumour that differs from theirs (say, a different article/source/conclusion) and doubts the credibility of it.
- $\eta < \gamma$. That is, it is more likely that a spreader will be convinced by a stifler to stop spreading a rumour versus another spreader (since spreaders may create an 'echo-chamber effect' amongst themselves).

Let $I(t)$, $S(t)$, and $R(t)$ be the *proportion* of ignorants, spreaders and stiflers (respectively) of the population at time t. The above rules yield the following system of equations:

$$\begin{aligned}
\frac{dI}{dt} &= -\kappa I(t)S(t) \\
\frac{dS}{dt} &= \lambda\kappa I(t)S(t) - \kappa S(t)(\gamma S(t) + \eta R(t)) - \delta S(t) \\
\frac{dR}{dt} &= (1-\lambda)\kappa I(t)S(t) + \kappa S(t)(\gamma S(t) + \eta R(t)) + \delta S(t)
\end{aligned} \tag{1}$$

Unlike other epidemic models (such as SIRS), this model assumes that for a given rumor, once someone decides to stifle it, they will never go back to spreading it. A diagram of these equations is visualized in Figure 1.

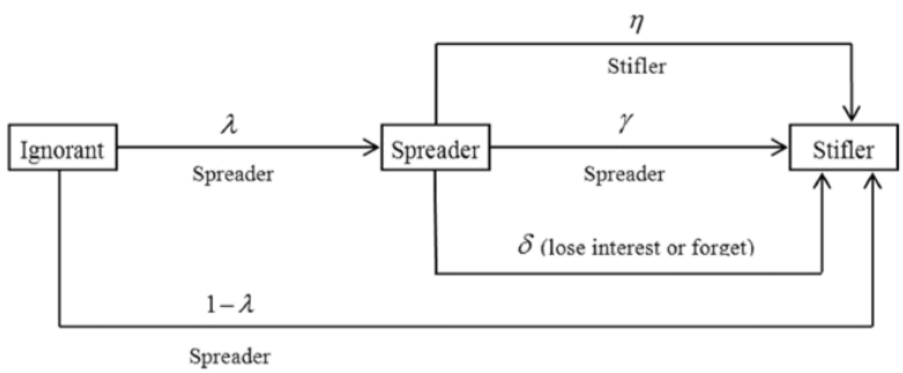


Figure 1: SIR diagram of the model described by Zhao et al. [10]

2.2 Steady State Computation

By construction, the system will always reach a steady state: as $t \to \infty$, $I(t) \to r$, $R(t) \to 1 - r$, and $S(t) \to 0$ for some proportion r. Concretely,

- All spreaders will eventually become a stifler.
- All ignorants remaining were those that were never contacted by a stifler.
- All stiflers were those contacted by a stifler.

Zhao et al. [10] derive the steady state equations for this system, with the main finding that if $\frac{\lambda}{\delta} > \frac{1}{\kappa}$, then spreaders can disseminate the rumor and let the system reach an equilibrium.

For this project's implementation, we estimate the time when a steady state is reached by computing the first differences of each solution curve and finding t^* such that

$$t^* = \min_{\frac{dI}{dt} = \frac{dS}{dt} = \frac{dR}{dt} \approx 0} t$$

2.3 Parameter Estimation

We base our parameter estimates on the Stanford Network Analysis Project (SNAP) Facebook egonet [7]:

- The total population of users N is the number of nodes (users) in the network: 4039
- To estimate the contact rate κ, the network's average node degree was used, which was 43.69. It is important to note that this is an *ego network*, meaning the node degree distribution is highly left-skewed and the network's mean node degree is much higher the median.

- The rumour acceptance probability λ was estimated using findings from surveys conducted on the spread of unverified information by Greenhill and Oppenheim [5]. Here, we took the average probability of belief in a rumour which was 0.2.
- The spreader-stifler stifle probability η was estimated by informally surveying friends on Facebook. Approximately 10% of them would stop spreading a rumour if one of their peers tried to stop them.
- The spreader-spreader stifle probability γ was estimated by dividing η by two. The rationale here is that people who believe in the same rumour are likely to face an 'echo chamber' effect, where two spreaders experience confirmation bias rather than convincing each other to stop spreading a rumour.
- The forget probability δ was set to 0.5 as used in Zhao et al's formulation [10].

Parameter	Meaning	Estimated Value
N	Population size	4039
κ	Avg contacts per timestep	43.69
λ	Rumour acceptance probability	0.20
γ	Spreader-spreader stifle probability	0.05
η	Spreader-stifler stifle probability	0.10
δ	Forget probability	0.50

Table 1: Parameter estimates for the modified SIR model

Our parameter estimates are summarized in Table 1. In reality, every rumour will have different properties so parameters will vary depending on its content. Other literature fits model parameters using historical data and parameter grid search.

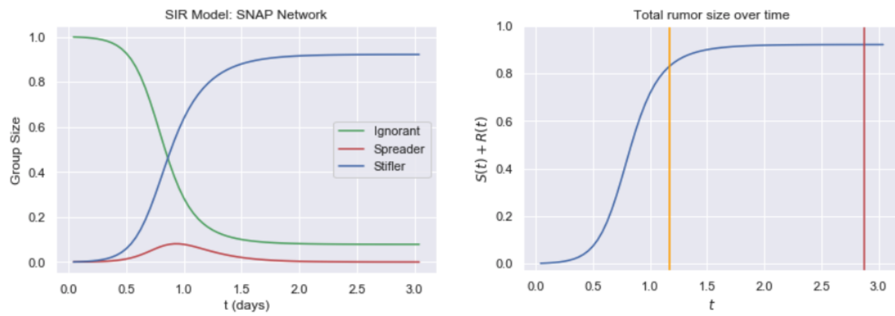
2.4 Solution

Using the above parameters we solve the system of equations using a fourth-order Runge-Kutta method implemented in Python's `scipy.integrate.ode`. The step size used is $h = \frac{1}{24}$ (one hour).

Define the rumour size $w(t)$ as the proportion of users that have come in contact with the rumour at the steady state time t^*:

$$w(t) = S(t) + R(t)$$

Using the parameter estimates from the previous section we solve the system of equations in (1), and compute t^* and $w(t)$. The rumour reaches a final size of 92% at $t^* \approx 2.8$ days, visualized in Figure 2.



(a) (b)

Figure 2: 2a plots the solution curves for (1), and 2b plots $w(t)$. The orange line marks t' such that $w(t') = 0.9w(t^*)$, and the red line marks t^*.

2.5 Sensitivity Analysis

Next, a sensitivity analysis was performed on each parameter. Holding all other variables fixed, one parameter was varied and 3 measures of rumour dynamics were calculated: the final rumour size $w(t^*)$, the time to 90% of the final rumour size t', and the time to final rumour size t^*. Figure 3 visualizes the three most impactful parameters on rumour dynamics.

Plots of the 3 measurements vs. population size N, the forget probability δ, and stifler-stifler stifle probability γ, were omitted as it was found that they had very little impact. These plots and animations can be found in this project's presentation slides: https://goo.gl/vqmpEH

2.6 Conclusions

Based on this SIR formulation, the most important factors contributing to the final size and velocity of the rumour's spread are the

- contact rate κ
- rumour acceptance probability λ
- and spreader-stifler stifle rate η.

That is, if a user wanted to successfully spread a rumour they should

- ensure the social circle they're trying to influence is active and well-connected (high κ).
- ensure the content is engaging and provocative enough to invite spreading of it (high λ). Aside from the content itself, adding a call-to-action such as encouraging friends to tag others in a post and/or share it with friends to spread awareness would also help.

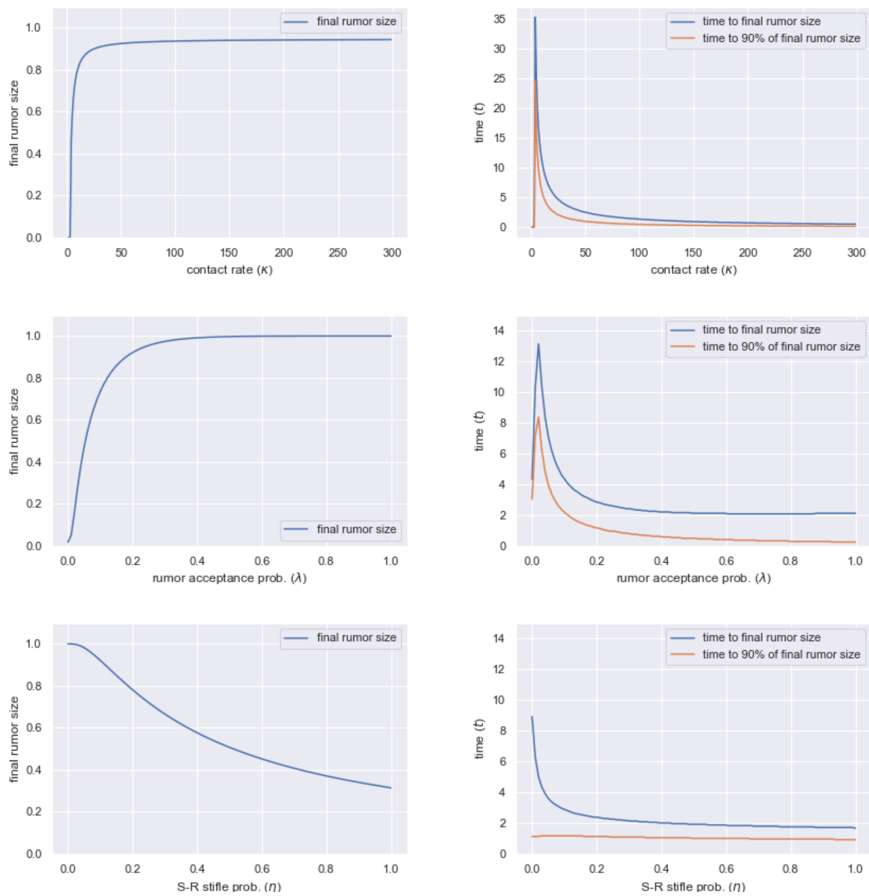


Figure 3: The left column of plots show the final rumour size $w(t^*)$ with respect to 3 different parameters. In the right column, the orange and blue lines represent the value of t' and t^*, respectively.

- ensure the content is believable enough so stiflers don't actively try to debunk the rumour (low η).

3 A Graph-SIR Model

The previous model contained an unrealistic assumption that the social network was fully-connected. To improve the previous model, we modify the SIR model to work over a social network dataset particularly an *ego network*. Ego networks focus on a single actor or "focal" node and their connections called "alters" [4].

3.1 The SNAP Network

For this project, the social network dataset used was an ego network from a Facebook user, provided by the Stanford Network Analysis Project (SNAP) [7]. It contains 4039 vertices and 88234 edges, with a graph density of 0.01082. Here, the graph density is defined as the total edges divided by the total number of possible edges. Figure 4, shows that the network is *scale-free* - that is, the degree distribution follows a power law.

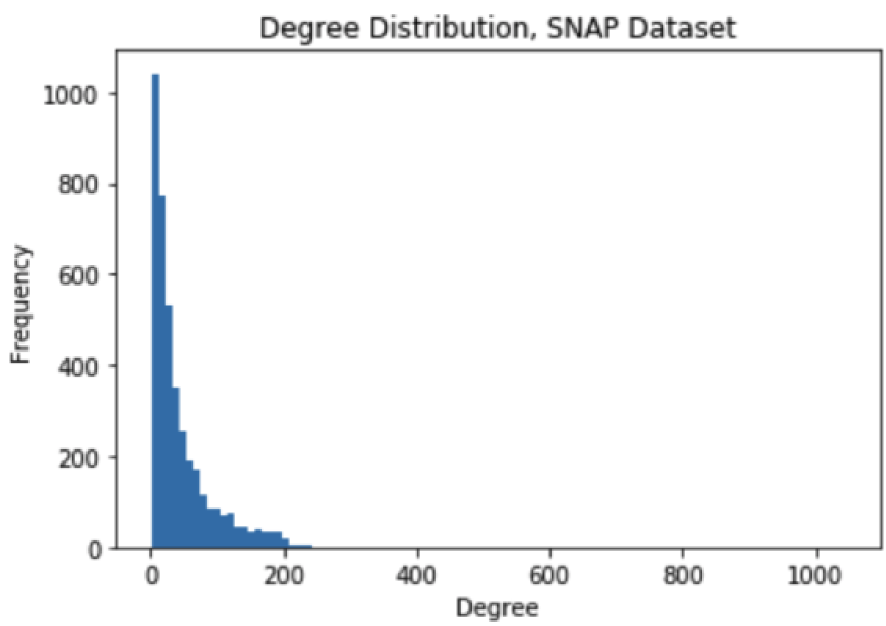
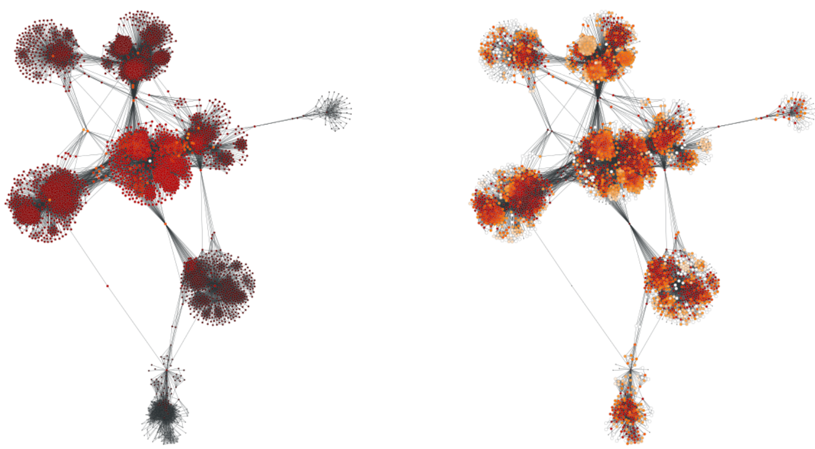


Figure 4: Degree distribution of the SNAP Facebook egonet.

We visualize the network using *graph-tool*, a efficient Python library for network analysis [8] In Figure 5 we plot the network and colour nodes based on each node's *closeness centrality* and *local cluster coefficient*, calculated using their implementations in graph-tool. The closeness centrality of a node was calculated as the reciprocal of the sum of the length of the shortest paths between the node and all other nodes in the graph. Intuitively, the local cluster coefficient describes the embeddedenss of a single node.



(a) Closeness centrality. (b) Local clustering coefficient.

Figure 5: Visualizing the SNAP network and colouring vertices using two different measurements. Nodes with brighter colour indicate larger values.

3.2 Methodology

To extend our previous SIR model to work on a social graph, we build a stochastic simulation the following rules:

- Initialize the simulation with a single Spreader node, chosen randomly.
- Each timestep (one day), each spreaders s will try to infect each of their neighbouring nodes n_i. For each i,
  - If n_i is ignorant, n_i will become a spreader with probability λ or a stifler with probability $1 - \lambda$.
  - If n_i is a spreader, s becomes a stifler with probability γ.
  - If n_i is a stifler, s becomes a stifler with probability η.
  - s becomes a stifler with probability δ, otherwise s remains a spreader.
  - If all neighbours of s are stiflers, then s becomes a stifler,
- Terminate when only ignorants and stiflers remain.

This model also adds the following assumptions:

- When a spreader posts a rumor, the rumor will show up at the top of their friends' newsfeed.
- All users open Facebook at the same time each day and read all of their friends' posts. This guarantees that they are exposed to a neighbouring spreader.
- All users behave the same, so every node will have the same parameter values.
- Spreaders of a rumour will try to infect all of their neighbours every timestep.

Note that in reality, Facebook newsfeed items are ranked based on various criteria such as relevance and a user's potential engagement with the post [2]. Also, spreaders may not try to spread a rumour every single day - such as if they only use Facebook on weekends.

3.3 Parameter Estimation

We will use identical parameters as in the previous SIR model. However, there will be no contact rate κ as each spreader node contacts all of their neighbours.

3.4 Visualizing a Single Simulation

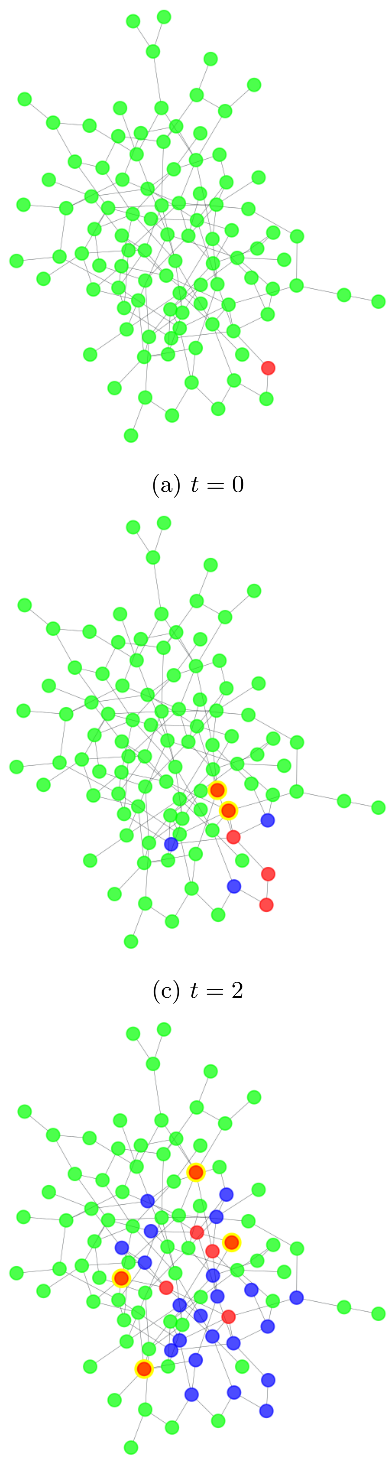
A visualization tool was built to plot the spread of infection at each timestep. When running a simulation, we plot each timestamp, colouring ignorant nodes green, spreader nodes red, and stifler nodes blue. Newly infected spreaders are highlighted in yellow. This series of images can be rendered into animations showing the progression of the rumour as it spreads through the network.

To test this, we generate a random graph with 100 nodes, each assigned a degree between 1 and 6. The first 6 timesteps are visualized in Figure 6.

We run and visualize a simulation on the SNAP network, displayed in Figure 8. Figure 7 shows the growth of the rumour over time for that simulation. In this particular example, the rumour started in the most central cluster of nodes and spread to approximately 70% of the network. At $t = 7$ it manages to penetrate a node the leftmost community and spread to most of its nodes, leading to a 20% jump in the rumour's size.

3.5 Monte Carlo Simulation & Analysis

We run our simulation with the same parameter set many times, randomly selecting an initial spreader node. For this project, the we ran the graph-SIR model 100 times and recorded each simulation's *trace*: the number of spreaders,



(a) $t = 0$

(b) $t = 1$

(c) $t = 2$

(d) $t = 3$

(e) $t = 4$

(f) $t = 5$

Figure 6: Visualizing our graph-SIR model on a small network.

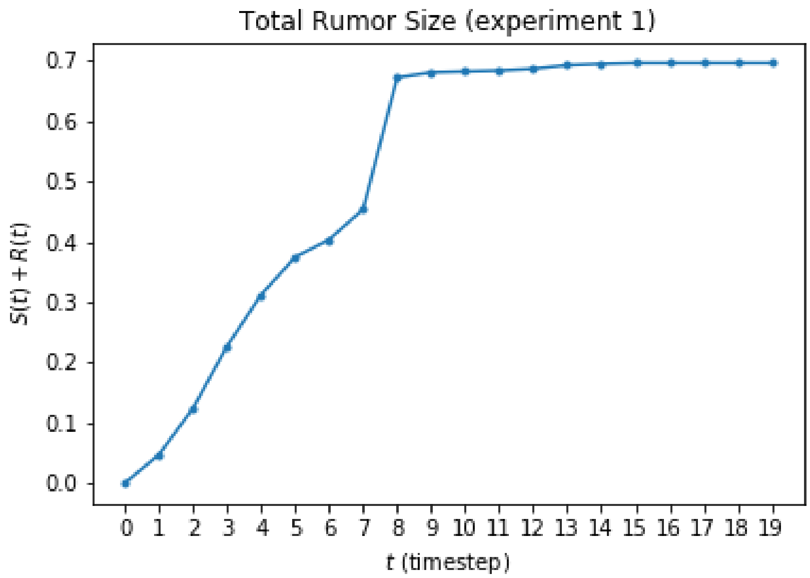


Figure 7: Plotting the SNAP simulation's rumour size $w(t)$ over time.

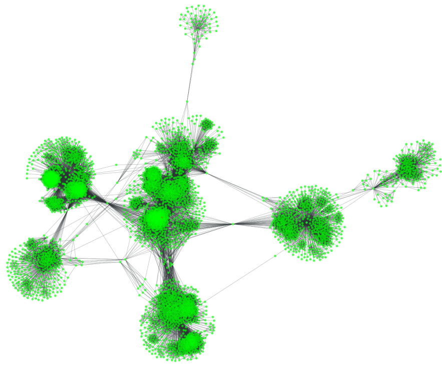
stiflers, and ignorants at each timestep, as well as its duration and final rumour size.

Figure 9 shows a histogram of final rumour sizes. In 50% of the simulations, the rumour spread to over 40% of the network. We also see that for many of the simulations, the rumour did not spread past 5%. These are likely where the infection began in a sparse community and could not 'escape' it. Clearly, the topology of the network, and the starting point of infection is very important in how a rumour propagates.

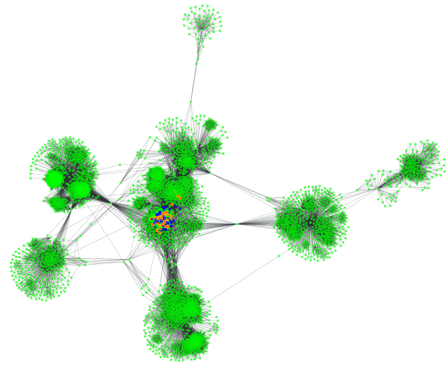
In Figure 10 we plot each simulation's rumour size over time. We note distinct patterns of kinks in the size of the rumour - these occur whenever a rumour reaches a new community and successfully spreads. We also see that, once a rumour has penetrated a community, it has a short window of time to escape it, otherwise it stops spreading and 'dies'.

3.6 Results

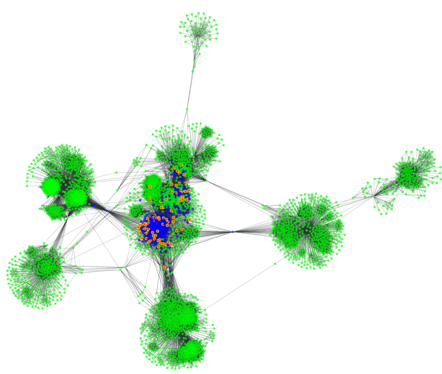
The wide variation in simulation results indicates that, for our model's parameters, the main factors contributing the rumour spread are the initial spreader node, and the rumour's ability to quickly penetrate other neighbouring communities. If a user wanted to successfully spread a rumour they should start with a densely-connected community that has many mutual connections to other



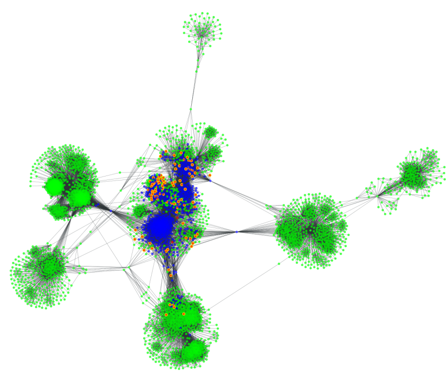
(a) $t = 0$



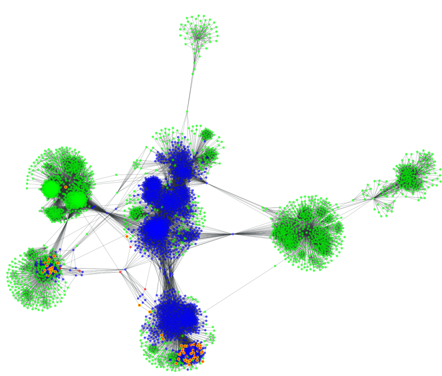
(b) $t = 1$



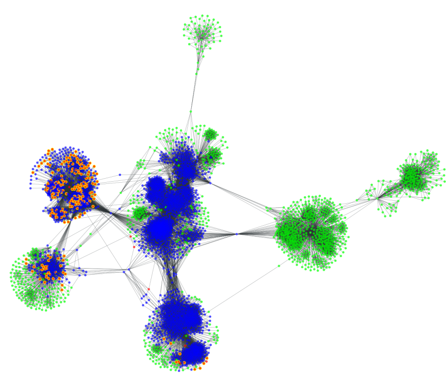
(c) $t = 2$



(d) $t = 3$



(e) $t = 7$



(f) $t = 8$

Figure 8: SNAP network plots of the rumour spread at particular timesteps, starting with a single spreader node. The simulation ended with a final rumour size of 70% after 20 timesteps.

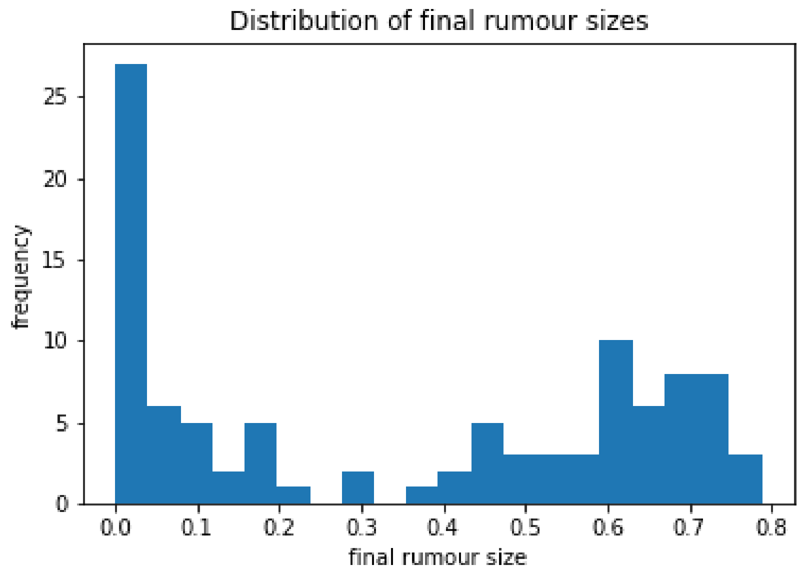


Figure 9: Distribution of final rumour sizes from 100 simulations of the graph-SIR model. In 50% of simulations the rumour's final size was over 40%.

neighbouring communities, to ensure the rumour doesn't get trapped in the first one.

Similar to the previous model, an ideal rumour would be highly engaging for users to spread it (high λ), and should be believable so that other stiflers don't work hard to debunk the rumour and silence other spreaders (low η).

4 Conclusions

We see that the original SIR model overestimates the final rumour size and the speed of rumour spread compared to our graph-SIR model. The highest final rumour size achieved was 80% in the latter whereas the former estimated a final rumour size of 92%. This is due to the fact that the former assumes a fully-connected network and the latter uses a scale-free network, and information requires more time to spread in graphs that are less connected.

According to our analysis of the (hopefully) more realistic graph-SIR model, the starting point of infection and the social network's topology are the two main factors contributing to the a rumour's final size.

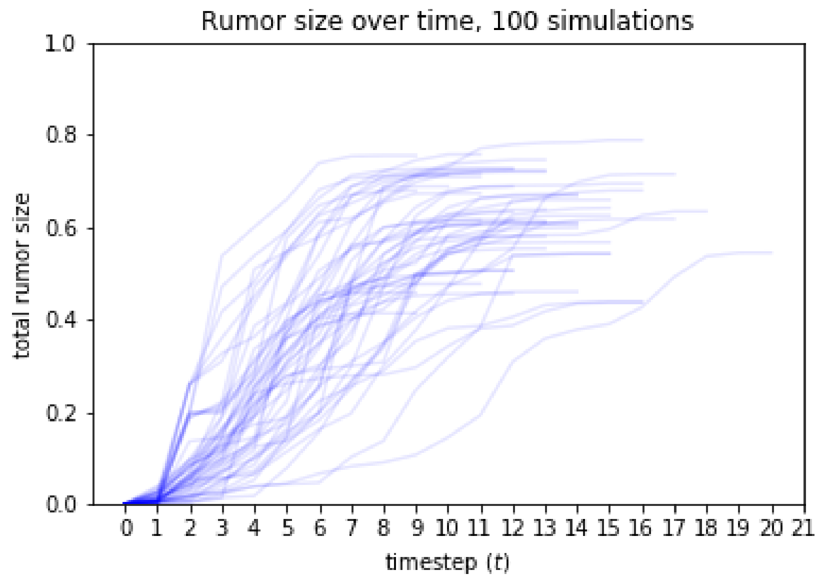


Figure 10: Plotting each simulation's rumour size over time. The highest final rumour size achieved was 80%.

5 Future Work

This project's analysis only scratches the surface of possible future work. Future questions for research include:

- If we used a fully-connected graph in the graph-SIR model and compared its results to the deterministic SIR model, do we get similar results? (when using the same set of parameters)
- Can we fit our model parameters to a real dataset?
- What specific conditions does a rumour 'blow up'? Does it have to do with a spreader node's centrality or betweenness? Certain values of the model's parameters? Particular network structures?
- What if we perform a sensitivity analysis on the graph-SIR model?
- Use more than one initial spreader - how do our results change?
- How can we improve the efficiency of our simulation's implementation? (100 simulations took 1.4 hours)

References

[1] Rumour. Available at https://en.oxforddictionaries.com/definition/rumour.

[2] Josh Constine. How facebook news feed works, Sep 2016. Available at https://techcrunch.com/2016/09/06/ultimate-guide-to-the-news-feed/.

[3] Daryl J Daley and David G Kendall. Stochastic rumours. *IMA Journal of Applied Mathematics*, 1(1):42–55, 1965.

[4] Rich DeJordy and Dan Halgin. Introduction to ego network analysis. 2008.

[5] Kelly M Greenhill and Ben Oppenheim. Rumor has it: The adoption of unverified information in conflict zones. *International Studies Quarterly*, 61(3):660–676, 2017.

[6] David Jackson. Ap twitter feed hacked; no attack at white house, Apr 2013. Available at https://www.usatoday.com/story/theoval/2013/04/23/obama-carney-associated-press-hack-white-house/2106757/.

[7] Jure Leskovec and Julian J Mcauley. Learning to discover social circles in ego networks. In *Advances in neural information processing systems*, pages 539–547, 2012.

[8] Tiago P. Peixoto. The graph-tool python library. *figshare*, 2014. Available at http://figshare.com/articles/graph$_{t}ool$/1164194.

[9] Chao Wang, Zong Xuan Tan, Ye Ye, Lu Wang, Kang Hao Cheong, and Neng-gang Xie. A rumor spreading model based on information entropy. *Scientific Reports*, 7(1):9615, 2017.

[10] Laijun Zhao, Hongxin Cui, Xiaoyan Qiu, Xiaoli Wang, and Jiajia Wang. Sir rumor spreading model in the new media age. *Physica A: Statistical Mechanics and its Applications*, 392(4):995–1003, 2013.