Gossips, Gossips

Spreading Rumors in Social Networks

Abstract

In this paper, we consider how to model rumor spreading through social networks, specifically through Facebook. We start with a differential equation model based off of the classic SIR model in order to get an understanding of potential behavior. We then expand that to an agent based model based on actual Facebook friends lists. By testing over hundreds of different combinations of parameters across several Facebook graphs, we were able to determine certain predictable behaviors that depend on the size and density of a given network. Some of these behaviors are expressed in the same way independent of the size and density and others are realized differently in different networks.

Background

All humans are active agents in rumor spreading. While we generally think of rumors as high-school gossip or schoolyard trash talk, rumors can have a large impact. In late 2013, rumors regarding how China would react to the rise of Bitcoin greatly affected the price of the currency days before any official statements were made. During the 2008 presidential campaign, rumors about Senator Obama's place of birth greatly affected the legitimacy of his candidacy. In March 2014, actor Wayne Wright had to publicly declare that he was in-fact not dead after rumors declaring otherwise spread through social networks. However, the spread of rumors is not always harmful. For example, the concept of viral marketing uses the quick spread of rumors through social media to promote goods and services by various companies.

Since the internet boom, the spread of rumors has greatly accelerated. With social products such as Facebook and Twitter, spreading a rumor from a friend halfway around the world is as easy as clicking a "share" button or "retweeting" a status update. Due to their well-defined user relationships, these products allow us to easily analyze the nature of such phenomena in ways we could not before.

Model 1 Equations

We will begin with an SIR model for Rumor Spreading. Consider a population consisting of N individuals which are subdivided into Ignorants (I), Spreaders (S), and Stiflers (R).

Assumptions:

- The rumor spreads by direct contact of the Spreaders with others in the population.
- The population size is constant during the lifetime of a rumor.
- Each person comes into contact with a percentage of the population k.

- Whenever a Spreader contacts an Ignorant, the Ignorant becomes a Spreader at a rate λ.
- When a Spreader contacts another Spreader or a Stifler the initiating Spreader becomes a Stifler at a rate α.

$$N = I + S + R$$

$$\frac{dI}{dt} = -\lambda kIS$$

$$\frac{dS}{dt} = \lambda kIS - \alpha kS(S + R)$$

$$\frac{dR}{dt} = \alpha kS(S + R)$$

We may simplify the above model by reducing it to two equations since we are assuming the population is constant. We may make the substitution $S + R = N - I$, where N is a constant.

$$N - I = S + R$$

$$\frac{dI}{dt} = -\lambda kIS$$

$$\frac{dS}{dt} = \lambda kIS - \alpha kS(N - I)$$

In this model, the only steady state is clearly along the line $S = 0$, and it is a stable steady state. This is because Ignorants depend on the presence of Spreaders for their population to change, and if there are no Spreaders, no additional Stiflers may be created.

Model 2 Equations

In Model 1, we assumed that Spreaders would become Stiflers only if they were themselves stifled by another Stifler or Spreader. However, Spreaders may also spontaneously decide to become Stiflers for a variety of reasons. For example, the Spreader may realize the rumor isn't as exciting as it used to be, or the Spreader may decide the rumor is harmful to a specific individual or group of people and feel guilty about continuing to spread it. To account for this, we may introduce a parameter δ to represent the rate at which a Spreader may spontaneously decide to become a Stifler.

$$N = I + S + R$$

$$\frac{dI}{dt} = -\lambda kIS$$

$$\frac{dS}{dt} = \lambda kIS - \alpha kS(S + R) - \delta S$$

$$\frac{dR}{dt} = \alpha kS(S + R) + \delta S$$

Reducing this to a model of two variables as we did before, we attain the following equations:

$$N - I = S + R$$

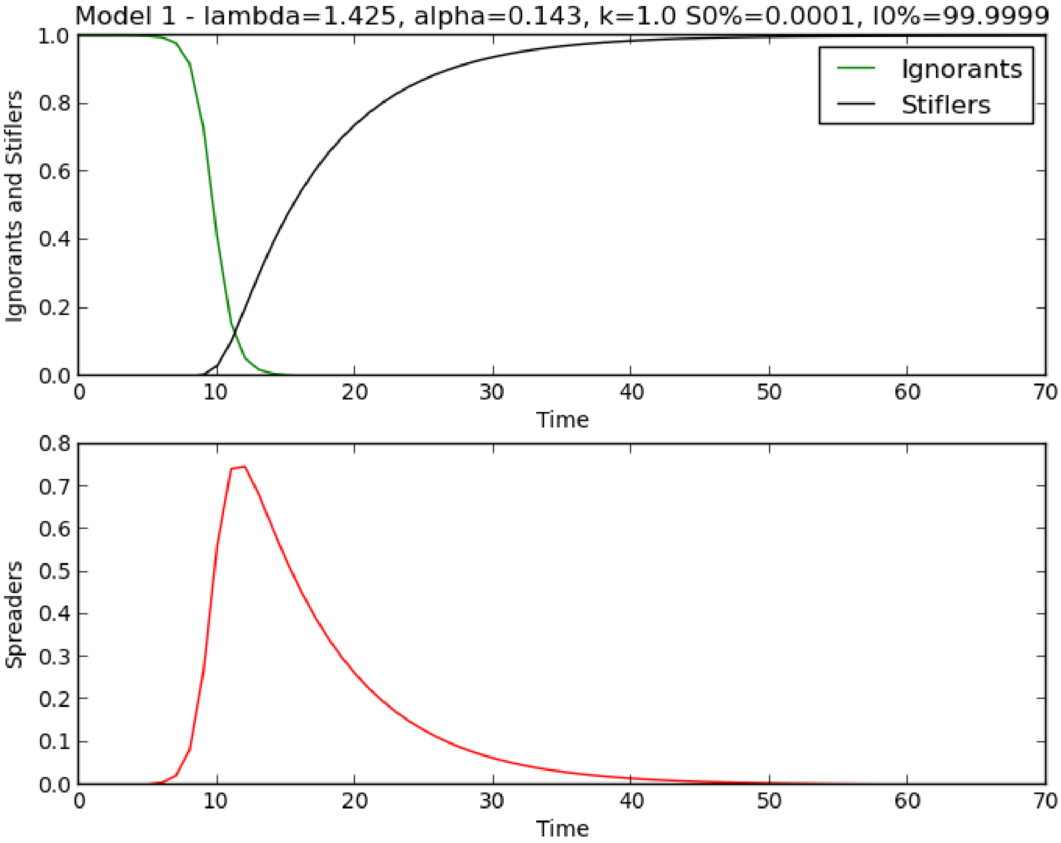
$$\frac{dI}{dt} = -\lambda kIS$$

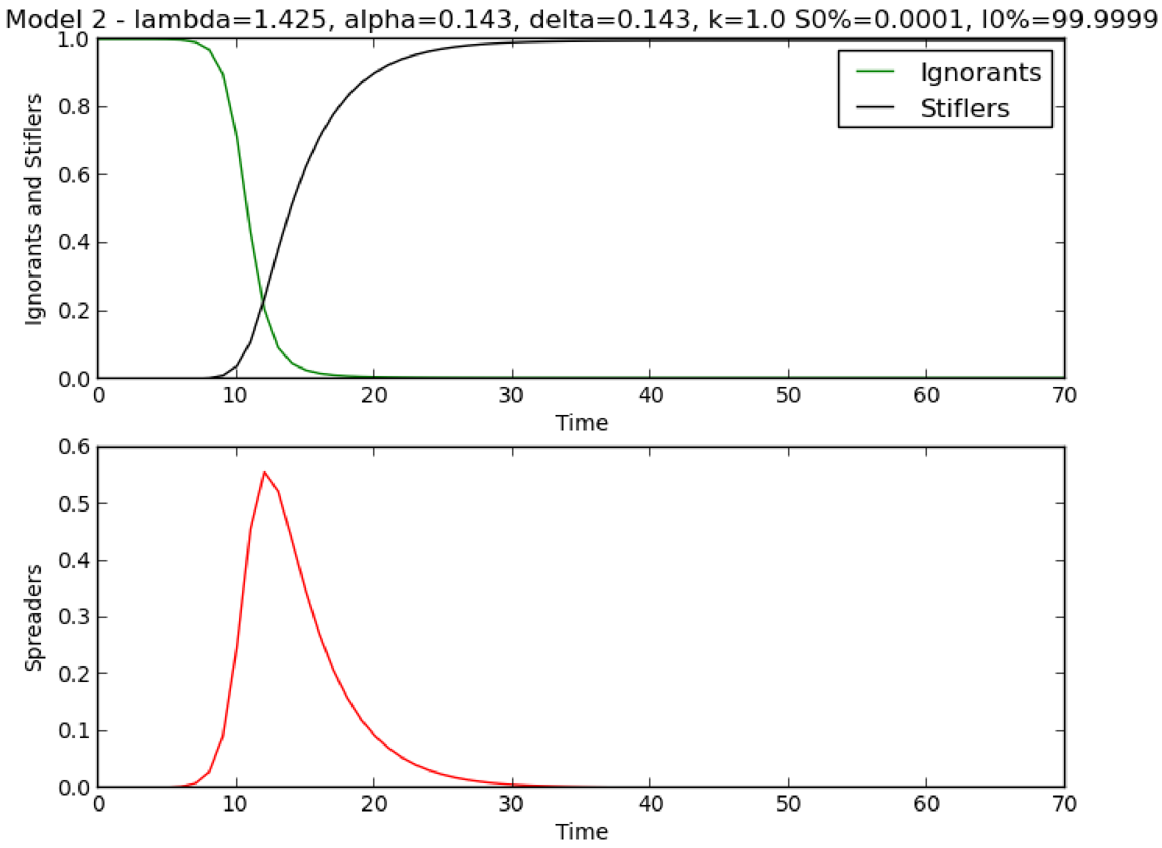
$$\frac{dS}{dt} = \lambda kIS - \alpha kS(N - I) - \delta S$$

Once again, the only steady state occurs along the line S = 0, and it is a stable steady state.

ODE Solution Graphs for Model 1 and 2

In the following two graphs, SO% refers to the initialing percentage of the populations that is Spreaders, and I0% refers to the initial percentage of the population that is Ignorants. Note that the time axis is unitless in this model.

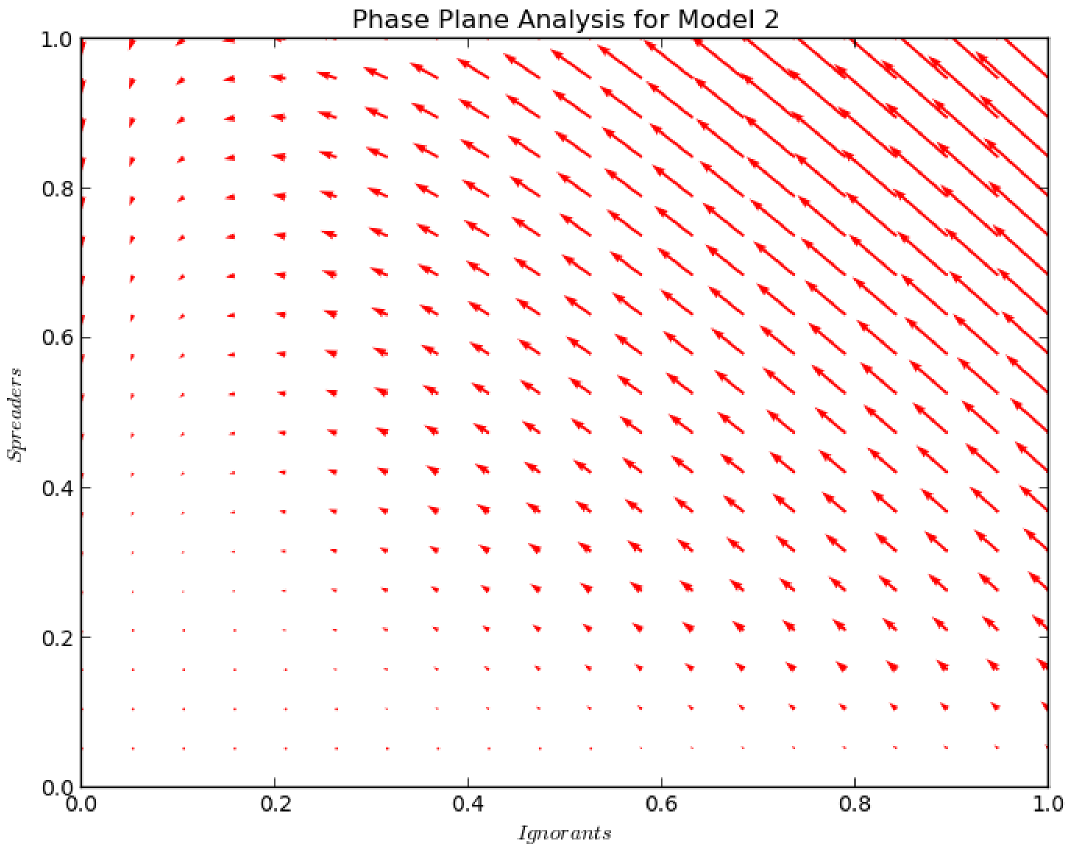




We see from both Model 1 and Model 2 that we expect a quick spike in the number of Spreaders as the rumor grows, followed by a long tail as the Spreaders are converted into Stiflers. In Model 2, the conversion happens quicker, which makes sense given the increased chance for a Spreader to become a Stifler.

Phase Plane Analysis for Model 2

We also conducted phase plane analysis between Stiflers and Ignorants. The analysis confirms our analysis of a single stable point when the number of Stiflers is 0. We see large arrows points towards an increase in Stiflers when there are both a large number of Stiflers and Ignorants, as there are enough people to be converted as well as enough people to perform the conversion.



Agent-Based Model

We decided to transition to an agent based model for several reasons. The foremost of these was that it would allow us to model the spread of a rumor in an actual network of nodes and edges. This is more realistic than our SIR model in which all actors may interact with all other actors. In addition, this fixes the counts of Spreaders, Ignorants, and Stiflers to integers rather than real numbers. This difference will lead to more realistic results on smaller networks, where the quantization of subpopulation sizes is more apparent on a graph. (We consider a small network later.) Lastly, it let us make the activity of our agents a function of the time of day. To do this, we were able to use real data about average Facebook activity from an article on Mashable. This way we could make our contactFraction -- the fraction of an agent's friends that he or she comes in contact with in a given time step -- a function of the time of day.

Input Data

Our Data came from SNAP (Stanford Network Analysis Project). The data consists of Facebook friends lists. It is split up into various graphs that represented different connected components of the collected data. Each graph has an ID, which represents the node whose friend list was used to generate the graph. For a given ID, the graph is formed by connecting the node with the given ID to all nodes in its friends list, and recursing on each of those friends. Statistics of the various graphs are shown below.

Graph	Size	Average Friends Per Node
0	334	17.08
107	1035	53.69
348	225	30.36
414	161	24.41
686	169	21.59
698	62	10.68
1684	787	37.64
1912	748	82.28
3437	535	19.99
3980	53	7.47

Algorithm

We chose to implement our algorithm in Python. From a high level, our model iterates over all possible combinations of our parameters, and runs a “model object” for each combination. A model object is defined by a set of parameters and a graph with nodes and edges. Upon creation, the model randomly chooses which nodes will be the initial Spreaders in accordance with the provided value for the number of initial Spreaders. It then partitions the nodes into three distinct sets of Ignorants, Spreaders, and Stiflers. Initially, all non-Spreaders are Ignorants. We then define a function run, which does all the work for a single time step - we chose each time step to represent one hour. Specifically, run will increment the time value, and for each Spreader already in the Spreader set, it will determine which nodes to spread to, whether it is stifled by an interaction, and whether it spontaneously is stifled. For spreading and stifling, each Spreader considers each friend in a group of “contacted friends” which is random subset of a node’s friends of size determined by the contact fraction parameter. Each model will call run exactly 48 times to represent a two day lifespan of a rumor, and reports the counts of Ignorants, Spreaders, and Stiflers at each time step. Because of the element of randomness, we can choose to run the model several times for a given set of parameters and average the results. For our initial training data, we chose not to do this; however, for the actual simulations that we analyzed, we averaged each model over 10 iterations. (Our training data and actual simulations are discussed later.) Finally, we plot the percentage of each subpopulation over time, and we display the non-Ignorant population (Spreaders and Stiflers who have become savvy to the rumor) over time as well.

Meaning of the Parameters

The following are descriptions of parameters that are used to initialize a model object in our Python script:

- graph: This is a collection of nodes and edges that represent a social network.
- spreadChance: This is the chance that an interaction between a Spreader and an Ignorant will result in the Ignorant becoming a Spreader. This value is highly dependent on the actual rumor in addition to the behaviors of the users in the network; therefore, it is difficult to accurately assign this value for a given rumor.
- stifleChance: This is the chance that an interaction between a Spreader and a non-Ignorant will result in the Spreader becoming a Stifler. This value is also highly dependent on the actual rumor rather than just the users in the network. For example, rumors that are known to be false by some Stiflers will have a high stifleChance.
- numSpreaders: This is the initial number of Spreaders in the network.
- contactFraction: This represents the average fraction of friends that any node has an interaction with in a given hour. This average value transformed into a specific fraction by multiplying it by a factor that depends on the time of day. Certain times of day, Facebook users are either more or less active. The data used for this transformation was acquired from Mashable.
- spontaneousStifleChance: This is the chance that a Spreader will spontaneously become a Stifler in a given hour. The reciprocal of this value is the expected duration of a node's Spreader lifetime if the Spreader has no interactions with other nodes. In Model 1, this value was just zero, so Spreaders would remain Spreaders forever if they had no interactions.

It is worth noting that spreadChance, stifleChance, and spontaneousStifleChance are dependent upon a specific rumor, but contactFraction is an attribute of a given network. Obviously in reality, each of the users in a given network won't have the same contactFraction, but we chose to model it this way for simplicity. Also numSpreaders is neither an attribute of the rumor nor of the network; it is a result of external conditions resulting in rumor introduction in the network.

Results of Training Graph

We started with our training graph, graph 0. This graph had 334 nodes and an average of 17 friends per node. This graph was conveniently around the median for both number of nodes and average friends per node. For each of our 5 parameters, we ran the simulation with 3 different values per parameter, leading to a total of 243 simulations. From there, we pruned our results. First, we eliminated all stagnant graphs – graphs in which the populations did not change, which was generally caused by extreme values in our parameters. We then grouped various simulations together and sampled them, noting their interesting features and recording their parameter values. The parameters we considered for these simulations are shown below.

Parameter	**Spread Chance**	**Stifle Chance**	**Initial Number of Spreaders**	**Contact Fraction**	**Spontaneous Stifle Chance**
Values	0.1, 0.5, 1	0.01, 0.1, 0.5	1, 5, 25	0.01, 0.1, 1.0	0, 0.1, 0.5

Below is a chart of specific observations for selected graphs. These findings aided in our decision on which parameter values to consider in our actual simulations that we would later conduct. The graph numbers refer to the plots attached in Appendix A. The parameters chosen to produce each of these plots may be found in the titles of each plot. The yield percentage refers to the percentage of the population that ends up as non-Ignorants. The peak Spreader percentage refers to the maximum percentage of the population that was Spreaders at any given time. The time value refers to the number of hours for the model to roughly stabilize; this value was measured by eyeball, while the other two values were computed programmatically.

Graph #	**Time**	**Yield %**	**Peak SP %**	**Notes**
6	8	65	23	Stiflers stabilize at 60%, while Ignorants stabilize at 40%.
8	9	78	40	Stiflers stabilize at about 80%, Higher Spread peak of 43
11	3	20	21	Ignorants stabilize around 70%, stay higher than Stiflers throughout
15	6	67	23	Almost identical to graph 6. Only change in parameters is NumSP.
16	11	83	46	Long duration, high yield
32	8	61	21	Very similar to graph 6, but with slightly lower yield. StCh and SSC differ.
35	9	58	19	Nearly identical to graph 32. Only CF differs.
50	5	52	23	Very similar behavior despite initial speaders set to 25 instead of 1 - slightly longer duration, slightly higher peak
86	3	94	80	Observation: Very high peak - spread chance has heavy influence
140	4	100	50	Stifle Chance limits growth of Spread peak.
97	6	100	82	CF lifted to 1 has high effect, brought yield of non-Ignorant to 99% - 100%

191	3	100	100	Reaches 100% very quickly, presumably from the high spread chance
113	4	100	60	Spreaders reach about 60% with low contact fraction
116	5	100	60	Similar peak to graph 113, despite much higher CF

Summary of Training Observations

The data from Graph 0 gave us good insight into the effects of our parameters. We observed that Spread Chance has a very large impact on the peak rumor size and overall shape of the simulation. Number of Spreaders does not affect the simulation very much, as it just puts the simulation a few time steps in the future. Stifle Chance generally does not have much of an effect on the peak rumor size or yield percentage, but it does affect how long it takes for the rumor to spread. Contact Fraction has a mild impact on peak Spreaders, yield percentage, and time. This makes sense as a high contact fraction on a sparse network is somewhat “equivalent” to a low contact fraction on a very dense network. Spontaneous Stifle Chance should not affect peak or yield percentage, except for values very close to 1.0. This is because the reciprocal of the stifle chance represents the expected lifespan of a Spreader that has no interactions; for realistic values, this is not a bottleneck. However, increasing the spontaneous stifle chance does decrease the time it takes for there to be no more Spreaders in a population.

With this insight, we were able to reduce the parameters for our actual simulations, shown below.

Parameter	**Spread Chance**	**Stifle Chance**	**Initial Number of Spreaders**	**Contact Fraction**	**Spontaneous Stifle Chance**
Values	0.10, 0.50	0.001, 0.10	1, 25	0.05, 0.50	0, 0.20

Graphs Considered in Actual Simulations

We wanted to choose graphs with different structure, so we decided we would pick a graph with a small number of nodes (SHD -- small, high density), a graph with a large number of nodes and a large number of friends per node (LHD -- large, high density), and a graph with a large number of nodes and a small number of friends per node (LLD -- large, low density). This led us to choose Graph 698 for SHD (62 nodes, 10.68 friends per node), Graph 3437 for LLD (535 nodes, 19.99 friends per node), and Graph 1912 for LHD (748 nodes, 82.28 friends per node).

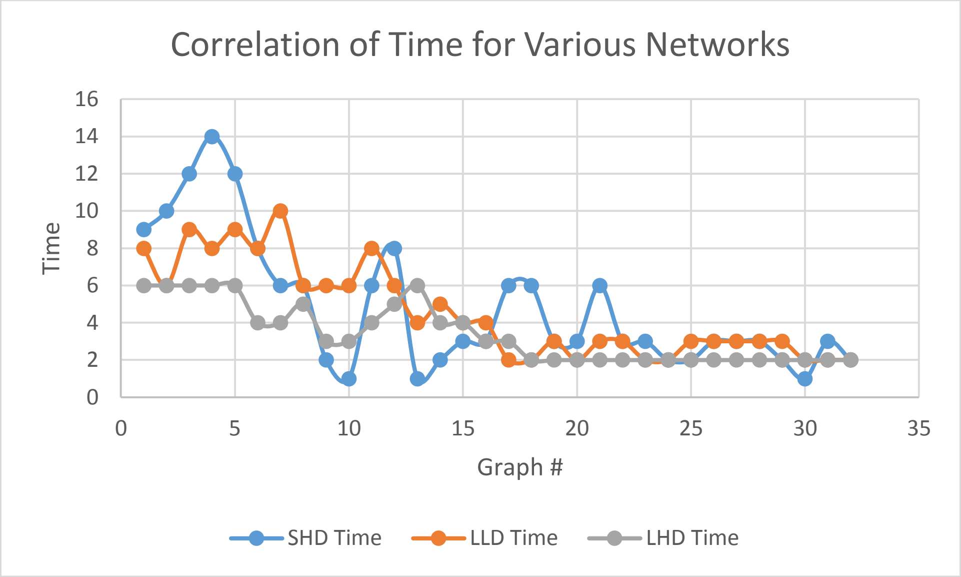
Results of Actual Simulations

In the follow set of results, the graph numbers refer to the graphs in Appendix B. SHD refers to plots where g=698. LLD refers to plots where g=3437. LHD refers to plots where g=1912. The parameters chosen to produce each of these plots may be found in the titles of each plot. The values were each determined in the same way that they were for the training graph data. Time is in hours.

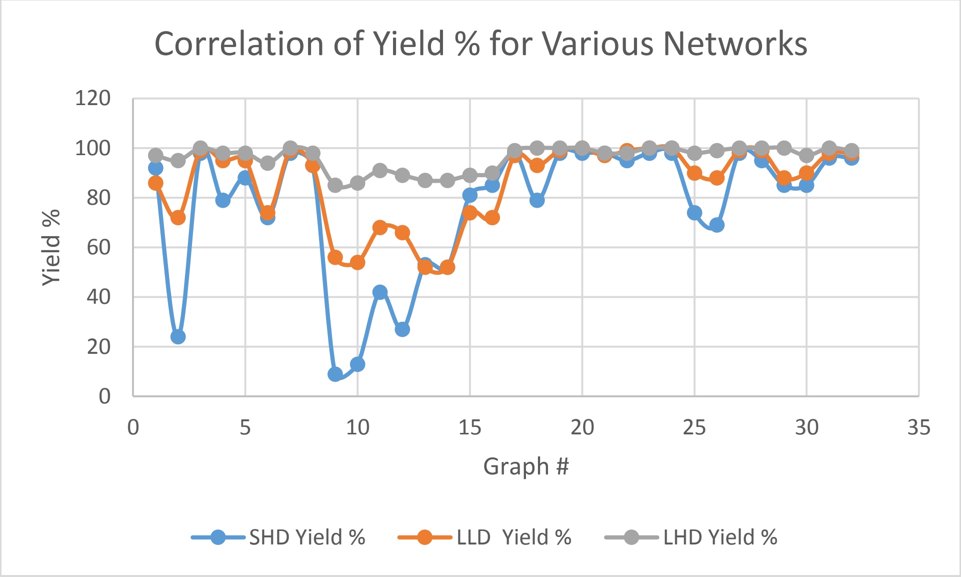
Graph #	SHD Time	LLD Time	LHD Time	SHD Yield %	LLD Yield %	LHD Yield %	SHD Peak SP %	LLD Peak SP %	LHD Peak SP %
1	9	8	6	92	86	97	72	74	96
2	10	6	6	24	72	95	17	42	64
3	12	9	6	98	99	100	84	86	80
4	14	8	6	79	95	98	45	60	66
5	12	9	6	88	95	98	83	74	75
6	8	8	4	72	74	94	41	40	63
7	6	10	4	98	99	100	89	85	78
8	6	6	5	93	93	98	56	53	64
9	2	6	3	9	56	85	10	24	42
10	1	6	3	13	54	86	10	19	42
11	6	8	4	42	68	91	12	23	43
12	8	6	5	27	66	89	14	22	42
13	1	4	6	53	52	87	47	25	38
14	2	5	4	52	52	87	40	22	38
15	3	4	4	81	74	89	40	23	38
16	3	4	3	85	72	90	40	24	39
17	6	2	3	98	97	99	89	93	92
18	6	2	2	79	93	100	64	76	84
19	3	3	2	98	99	100	97	96	95
20	3	2	2	98	100	100	82	85	85
21	6	3	2	98	97	98	96	93	91
22	3	3	2	95	99	98	73	78	76
23	3	2	2	98	100	100	97	96	92
24	2	2	2	98	100	100	83	82	77
25	2	3	2	74	90	98	67	66	58
26	3	3	2	69	88	99	52	60	55

27	3	3	2	98	99	100	57	59	52
28	3	3	2	95	99	100	53	56	53
29	2	3	2	85	88	100	75	62	75
30	1	2	2	85	90	97	67	63	76
31	3	2	2	96	98	100	67	40	75
32	2	2	2	96	98	99	65	45	75

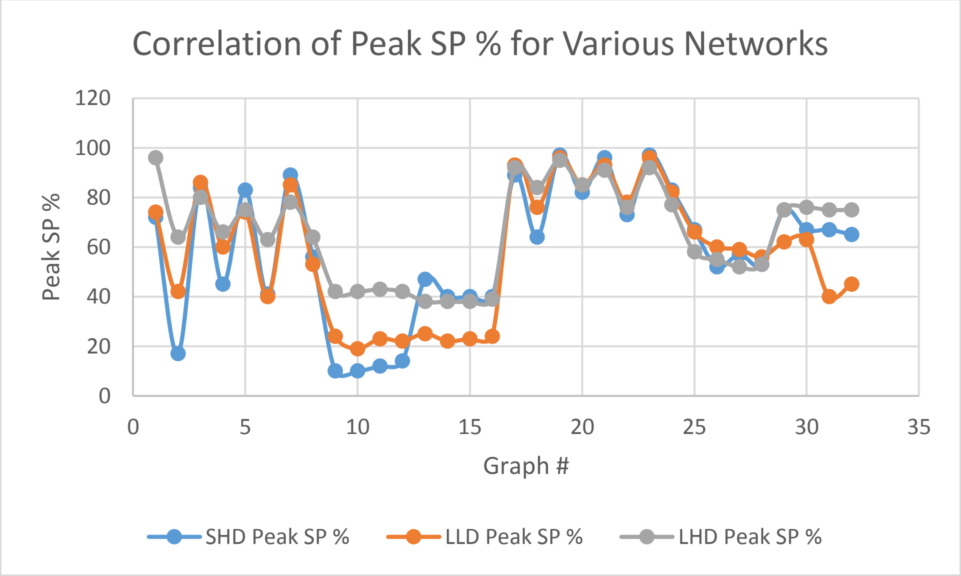
With such a large resulting table of data, it is difficult to see trends that give meaningful insight into the difference in behavior in the various sizes and densities of the networks we considered. We wanted to determine if there is any similar or divergent behavior in the various networks. Using Excel, we were able to produce the following graphs to consider each of our dependent variables individually.



Admittedly, the above graph is the least telling of the three. The trend lines do not seem to follow any patterns, and this is likely due to the fact that there are so few possible discrete values for the time measurement. In addition, since time was human measured rather than calculated, human error may be obscuring a pattern that does exist. Regardless, the following two graphs are more interesting.



This graph is interesting because the three trend lines almost never intersect each other. This means that the SHD Yield % is consistently less than the LLD Yield % which is consistently less than the LHD Yield %. It makes sense that the large high density network would have the greatest yield because nodes would have more opportunities to spread rumors to neighboring nodes in these graphs. It is curious that the SHD network almost consistently has a lower Yield % than the LLD despite being a high density network. A possible explanation might be that density does not necessarily cause greater yield; it only aids in the quick spread of a rumor. Meanwhile large networks allow for more nodes to be Spreaders, allowing the rumors to have longer lifetimes, leading to greater yields.



Except for graphs 9 through 16, these lines almost overlap, which leads us to conclude that peak spread percentage and overall yield do not necessarily correlate, since the trend lines on the previous graph on

Yield % did not overlap. A possible explanation may be that different stifling rates allow the same peaks to be descended at different rates, but along the way, more Spreaders may be introduced. Therefore, even though all networks may show similar peak Spreader percentage behavior, they will still have different over yields of non-Ignorants due to differences in the size and density of the networks.