# Information Theoretic Adversarial Approach to Rumour Spread

Ahmed Zaheer Dadarkar, Rakesh Kumar
111701002, 111701024

Indian Institute of Technology Palakkad

May 2021

**Abstract**

Rumours have huge negative impacts on the lives of people, hence modelling the spreading of rumours is essential for analysis and development of countermeasures. Hence in this paper, we study the model described in the paper [6] and develop an adversary which targets the spreading of rumours. Then, we modify the original model to better represent the real-life scenario, and develop an adversary for this modified model. Finally, we perform steady-state analysis of the modified model, before and after introducing the adversary as well.

## Contents



## 1 Introduction

Rumours tend to have a huge negative impact on the lives of people, affecting their decisions, beliefs and even their daily life. As described in the paper [6], Rumours lead to distortion of scientific articles and also influence political opinion. A recent example is the spreading of the WhatsApp privacy policy rumours [5], this led to widespread uninstallation of the WhatsApp application. This rumour had spread due to the lack of clarity of information among people, i.e. the communication of information between WhatsApp and it's users was not clear. Another example would be the case of Elon Musk tweeting about using Signal (a messaging application) [3], but due to lack of clarity of understanding the conveyed information, investors had invested in a small medical device company having the same name, this has led to a surge in stock price of this company. A political example would be as follows, Trump supporters spread rumours that coronavirus was a bio-weapon created by China [4].

All of our analysis in this paper would be based on the Information Entropy based model described in the paper [6]. Their model takes into account memory of individuals, conformity, trust between individuals and distortions produced while conveying information. Their paper also describes the existence of earlier rumour models which were

based on Social and Biological Contagion, Magnetization and Phase Change Phenomena, Game Theory, Interacting Markov Chains, and more.

A directed graph describes a network of individuals better than an undirected graph in a rumour spread model, since communicating information (or rumours) is not symmetric. For example, well-known people are able to convey information to a large number people, while very few people are able to communicate information to these well-known people. Hence, we improve upon the earlier model by the use of a directed graph in place of an undirected graph. Further, we describe an adversarial approach which would act against the spread of rumours in the network by diverting the "opinion" of individuals from false rumours to the true information, for both the original model, and the modified model. Formally, following are the contributions of this paper,

1. Modified the original model by using a directed graph instead of an undirected graph.
2. Introducing an adversary to the spread of rumours in order to divert the opinion of individuals from false rumours to the true information.

# 2 The Original Rumour Spread Model

## 2.1 Model Description

In this section we briefly describe the original rumour spread model contributed by the paper [6].

- **Network**: In this model, individuals are modelled as nodes in an undirected graph. Edges in this graph represent the relationship between two individuals, and allow for communication of information (potentially rumours) between them. The undirected graph used for modelling the network is a (BA) Barabási-Albert scale-free network [1]. We believe that a major reason for choosing the BA network was that it's degree distribution approximately follows the power-law, which results in the probability of large degree nodes (well-known individuals) being sufficiently large. Hence, the BA network is able to model the existence of a few well-known individuals, and a large number of common individuals.
- **Information and Memory**: Information which would be communicated between individuals, and which is stored in the memory of individuals is modelled as a fixed-length (say length $s$) binary string. Individuals have a finite-capacity FIFO (First In First Out) memory (say capacity $L$), in which they store these binary strings.
- **Selecting Information for Communication**: The binary string which is to be communicated by an individual is one which has the largest occurrence in it's memory, where ties are broken randomly.
- **Distortion and Acceptance of Information**: Before communicating a binary string, a node may distort this binary string in it's memory and then communicate this distorted binary string to it's neighbours. Also, if a node receives a binary string from it's neighbour, then it may or may not accept this binary string, but if it does, then this binary string is stored in it's memory. Formally, there is a probability $P_n$ that the binary string to be communicated is distorted before being communicated. Note that the first occurrence of the binary string is distorted in the memory. Also, there is a probability $\eta_{mn}$ that the binary string received by $m$ from $n$ is accepted into $m$'s memory. These probabilities are as follows,

$$P_n = \frac{1}{exp(\frac{H_{max}-H_n}{H_{max}}K)+1}$$

where $H_n$ is the information entropy of the distribution of binary strings in $n$'s memory, $H_{max}$ is the maximum possible information entropy a node can have, which is $s$ (achieved by a uniform distribution over all $2^s$ possible $s$ length binary strings), and $K$ is the conservation factor i.e. large $K$ leads to a lower probability of distortion and a small $K$ leads to a higher probability of distortion.

$$\eta_{mn} = \frac{degree(n)^\beta}{\max_{l \in nbhd(m)} degree(l)^\beta}$$

where $\beta$ is the confidence factor i.e. positive $\beta$ leads to acceptance of binary strings from well-known nodes often and rarely from lesser-known nodes, while a negative $\beta$ leads to the reverse. $\beta = 0$ leads to all received binary string being accepted, since in that case $\eta_{mn} = 1$.

The simulation of this model consists of repeatedly performing two steps, each of which are stated below,

1. **Spreading Phase**: During this phase, only nodes which have at least one binary string in their memory can participate. Each node picks a binary string which has the highest occurrence in it's memory (ties broken randomly), and distorts it with probability $P_n$ in it's memory, and then sends it to all it's neighbours.
2. **Acceptance Phase**: During this phase, nodes consider all the binary strings which have been received by them from their neighbours, and accept them into their memory with probability $\eta_{mn}$.

For a more detailed explanation of the model, one can refer to the paper [6], where this model was introduced.

## 2.2 Important Definitions

Here we describe a few definitions which were defined in the original model's paper,

**Definition 1** (Average Entropy)**.** At any timestep $t$ of the simulation, the Average Entropy is computed as the average value of information entropy over all the nodes in the graph, where the information entropy of a node at timestep $t$ is the entropy of the frequency distribution of the binary strings in it's memory. Formally, it is as follows

$$H_{avg} = \frac{1}{N} \sum_{n} H_n$$

Higher value of average entropy indicates the presence of many different rumours in the network, while a lower value indicates the presence of fewer variety of rumours.

**Definition 2** (Opinion)**.** At any timestep $t$, $i$ is considered an opinion of a node $n$ if $i$ has the highest frequency of occurrence in the memory of node $n$.

**Definition 3** (Opinion Fragmentation)**.** At any timestep $t$, the opinion fragmentation is measured by computing for each binary string $i$, the proportion of nodes $\delta_i$ which hold opinion $i$, i.e. the fraction of nodes which contain this $i$ as it's most frequent string. Formally, this is computed as follows,

$$\delta_i = D_i/N$$

where $D_i$ is the number of nodes which contain $i$ as it's most frequent string. If for some $i$, $\delta_i$ attains a high value, many individuals believe $i$ to be the true information (with high probability).

**Definition 4** (Range of Information Spread)**.** At any timestep $t$, the range of information spread is measured by computing for each binary string $i$, the proportion of nodes $\mu_i$ which contain $i$ in their memory, this is computed as follows,

$$\mu_i = W_i/N$$

where $W_i$ is the number of nodes which contain $i$ in their memory. For some $i$, higher value $\mu_i$ indicates that $i$ has spread to many individuals in the network.

## 2.3 Introducing an Adversary

Designing a model for rumour-spread allows us to study the process of the spreading of rumours in the network. But, the spreading of rumours is not a desirable phenomenon, hence it is very useful to design an adversary which would make the true information (say binary string $i_{true}$) the most prevalent opinion (Definition 2) in the network, i.e. make most nodes in the graph hold opinion $i_{true}$.

We have designed our adversary as follows - Suppose the adversary is enabled after some timestep $t_0$, then it will pick the top $r$ nodes with the highest degrees, and completely fill their memories with the true opinion $i_{true}$, and the information in the memories of these $r$ nodes would neither be distorted nor any information from their adjacent nodes will be accepted into their memory. Note that information could still be distorted while being communicated from these $r$ nodes, however distortion in the memory will not take place. This adversary could be implemented in the real-world by asking the $r$ most popular individuals to communicate the true information to all those who are in contact with them.

The scale-free property of the underlying network enabled us to come up with this design of the adversary. The intuition behind the design of our adversary is that scale-free networks contain few (but sufficient) number of nodes

with a high degree, since the degree distribution approximately follows the power-law. These nodes which contain a high degree are able to transfer information to a large number of nodes in one timestep.

We now perform experiments by simulating this adversary on the original rumour-spread model. For this we generate the underlying graph by using the BA scale-free method with number of nodes $N = 1000$, $m_0 = 15$ and $m = 10$. A sample generated graph's degree distribution is shown in Figure 1, the graph is connected and it has a diameter of 4. The memory size is kept as 100, and the binary string's length was set to $s = 5$. We perform simulations with multiple values of $K$ and $\beta$. We run each simulation starting with node number 0 having one occurrence 00000 in it's memory, the simulation is run up till 2000 timesteps, and we introduce the adversary after the 500th timestep (since until then, rumours are assumed to be widely spread in the network). Also, note that we only show those simulations in which the majority opinion had been shifted from the true opinion (before the introduction of the adversary), since if the majority opinion was the true opinion itself, then most of the individuals already believe that the true opinion is indeed true. The code for the experiments is present here - `https://github.com/tymefighter/RumourSpread/tree/main/OriginalModel`.

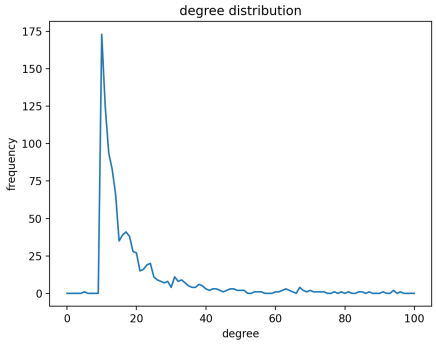


Figure 1: Degree distribution

The plots obtained from the simulation are shown in Figure 2.

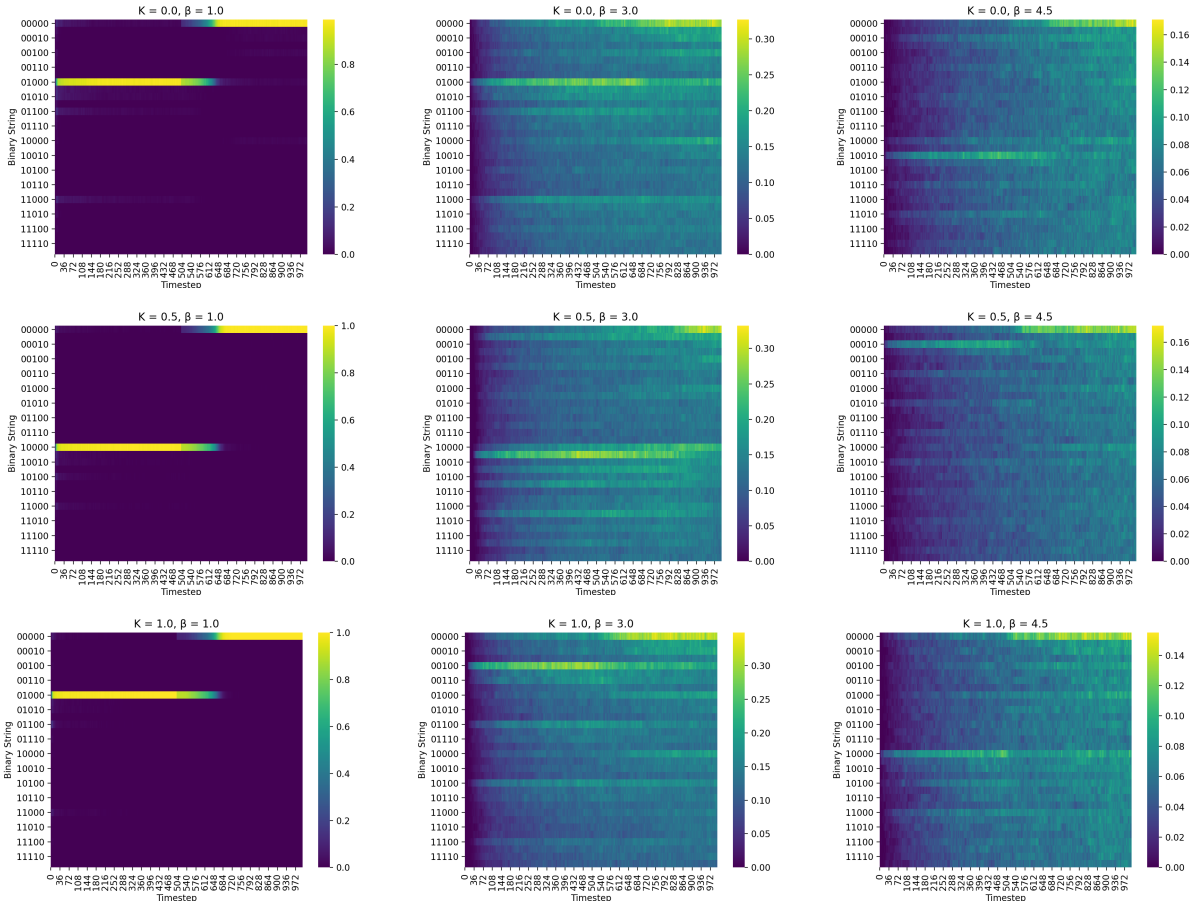


Figure 2: Simulation with Adversary - Opinion Fragmentation Plot for different $K$ and $\beta$ (plot is annotated with the values), the corresponding values of $r$ was set as (i) $r = 100$ (ii) $r = 10$ (iii) $r = 10$ (iv) $r = 100$ (v) $r = 10$ (vi) $r = 10$ (vii) $r = 100$ (viii) $r = 10$ (ix) $r = 10$

From the plots in Figure 2, we can clearly see that for different values of $K$ and $\beta$, we were able to manually set the value of $r$, which led to the adversary being able to divert the opinion of many nodes from a false information to the true one. We notice that only when $k = 0$, we required $r$ to be as high as 100, while for others $r = 10$ works well.

# 3 Directed Graph based Modification

The original model [6], uses an undirected scale-free graph for it's underlying network of individuals. But, it would be more meaningful if the graph was directed, the reason being that in real life, information does not necessarily flow both ways along a edge. For example, Broadcast media, such as television, newspapers, or even social media pages/profiles of famous individuals, are able to communicate information to a large number of individuals, but not all these individuals who receive information from them are to communicate their views back to them. Hence, we modified the original model by using a directed scale-free graph instead of an undirected scale-free graph.

## 3.1 Description

All those components of the original model which have been modified due to the introduction of a directed underlying network are described below,

1. **Network**: We use a directed scale-free graph as the underlying network for the model. For generating a directed scale-free graph we use the algorithm described in the paper [2]. The scale free graph produced by this algorithm has both in-degree and out-degree distributions approximately following a power-law. But, the

algorithm produces a graph which contains self-loops and multi-edges, which are completely counter-intuitive in our context. Hence, we modified the algorithm by accepting the addition of an edge only if it is not already present, and both endpoints of the edge are not the same vertex. Also, the outputted graph need not be connected, but we require that the network of individuals be connected for information to be spread, hence we randomly introduced bidirectional edges between the strongly-connected components in the outputted graph until the graph becomes connected.

2. **Acceptance of Information**: The earlier definition of acceptance probability is now ambiguous since it has not been mentioned whether to use the out-degree or in-degree in place of the degree value. We propose to use the out-degree as information is transferred along an edge, since for a positive $\beta$, individuals who have a higher number of out-edges are trusted more.

$$\eta_{mn} = \frac{\text{out-degree}(n)^\beta}{\max_{l \in \text{in-nbhd}(m)} \text{out-degree}(l)^\beta}$$

## 3.2 Experiments

We now perform simulations with the modified model to better understand the spread of rumours and increase in entropy of the system. The underlying network was generated using the algorithm described in paper [2]. We have set the algorithm parameters as $\alpha = 0.04$, $\beta = 0.9$, $\gamma = 0.06$, $\delta_{in} = 20$, $\delta_{out} = 20.0$, with initial number of nodes being 10, and final graph having 1000 nodes. Note that this $\beta$ is the graph generation algorithm's parameter, and is different from the confidence factor $\beta$. The in-degree and out-degree distributions of a sample generated graph are shown in Figure 3. This sample graph had just one strongly connected component (as we had ensured this property) and a diameter of 15. Further, the memory capacity was set to 100, the binary string's size was chosen to be $s = 5$, and for each simulation 10 nodes were randomly sampled using the out-degree distribution and one instance of the true information 00000 was placed in their memory. The code for the experiments is present here - `https://github.com/tymefighter/RumourSpread/tree/main/DirectedGraph`.

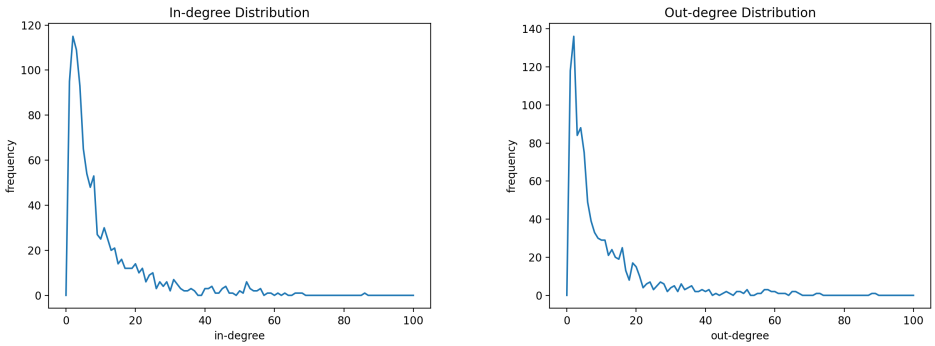


Figure 3: In-degree and Out-degree distributions

We now show simulation results for conservation factor $K \in \{0, 0.5, 1, 3, 6, 10\}$ and confidence factor $\beta \in \{1, 3, 4.5\}$. Note that we have kept $\beta$ as a positive quantity, but in the original model, $\beta$ was allowed to be non-positive as well. We have kept it positive because in real-life scenarios, people do not trust anyone blindly - which corresponds to $\beta = 0$, and do not prefer people with lesser out-neighbours over people with more out-neighbours - which corresponds to $\beta < 0$.

- **Average Entropy**: The plots of average entropy are shown in Figure 4. From these plots we can notice that lower values of $K$, such as $\{0, 0.5, 1.0\}$ lead to average entropy being stabilized at a higher value, and higher values of $K$, such as 3 lead to either a lower value of entropy or for $K \in \{6.0, 10\}$, there is no increase at all. This reason for this is that lower values of $K$ lead to a higher probability of distortion, which lead to multiple different binary strings being distributed in the network, while higher values of $K$ lead to a lower probability of distortion, and hence very few binary strings are circulated in the network. Also, we can notice that on decreasing $\beta$, the average entropy decreases at a higher slope after increasing.

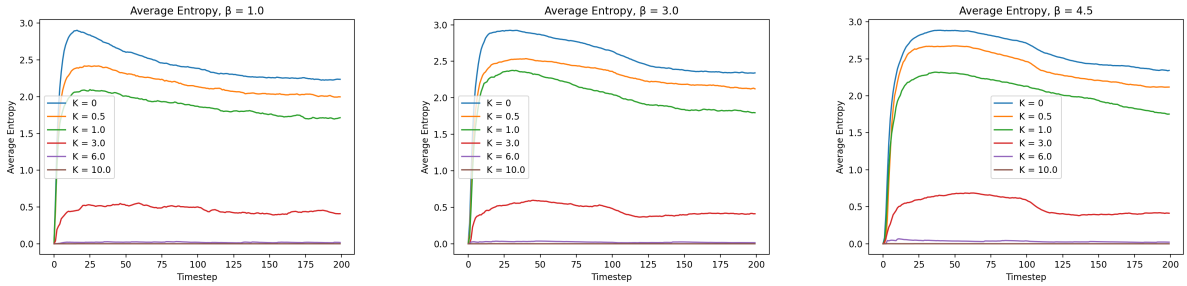


Figure 4: Average Entropy Plots for $K \in \{0, 0.5, 1, 3, 6, 10\}$ with (i) $\beta = 1.0$ (ii) $\beta = 3.0$ (iii) $\beta = 4.5$

- **Opinion Fragmentation**: The plots of opinion fragmentation are shown in Figure 5. In all of the plots we notice that there is a single opinion which dominates the entire network, however, this is not necessarily the true binary string which was initially distributed in the network (which was 00000). Hence, it may be possible that the opinion of individuals may shift to a false information, and this is problematic since most individuals now feel that the false binary string is the true one.

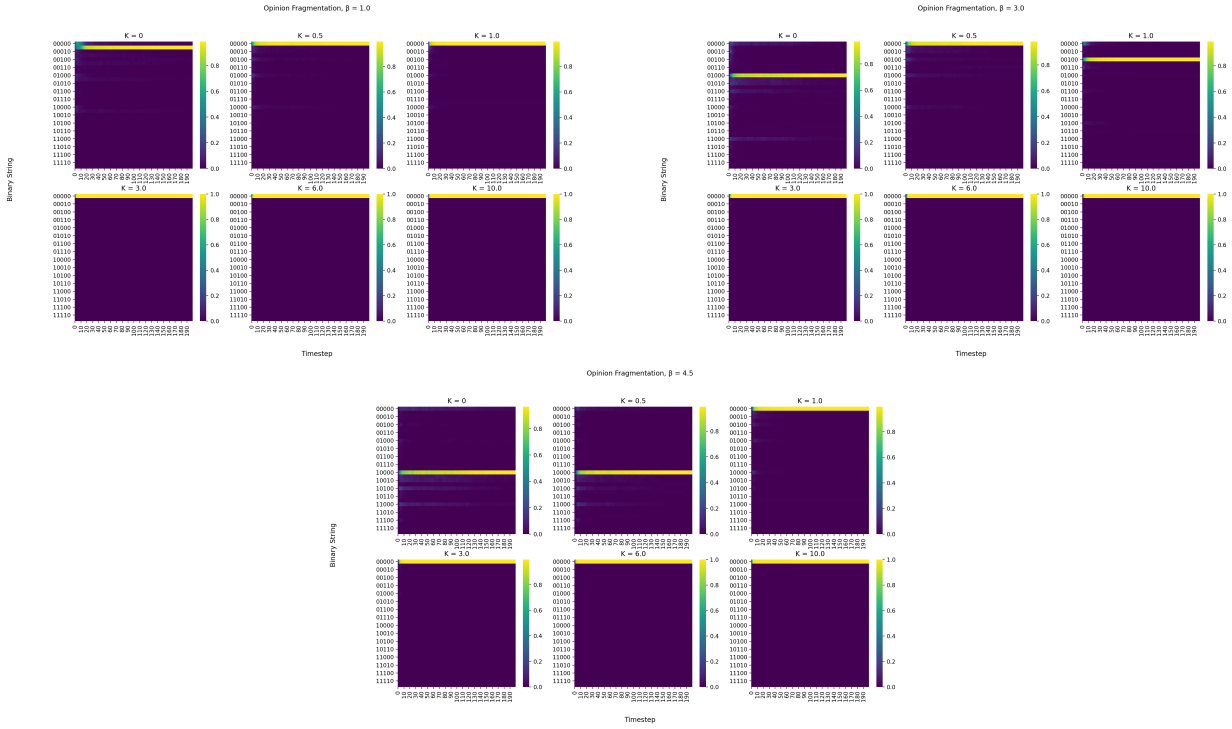


Figure 5: Opinion Fragmentation Plot for $K \in \{0, 0.5, 1, 3, 6, 10\}$ with (i) $\beta = 1.0$ (ii) $\beta = 3.0$ (iii) $\beta = 4.5$

- **Range of Information Spread**: The plots of range of information spread are shown in Figure 6. In each of these plots, we notice that lower values of $K$ lead to many different binary strings being circulated in the network, while higher values of $K$ lead to very few binary strings being circulated in the network. This again can be explained using the fact that decreasing $K$ leads to an increase in distortion probability.

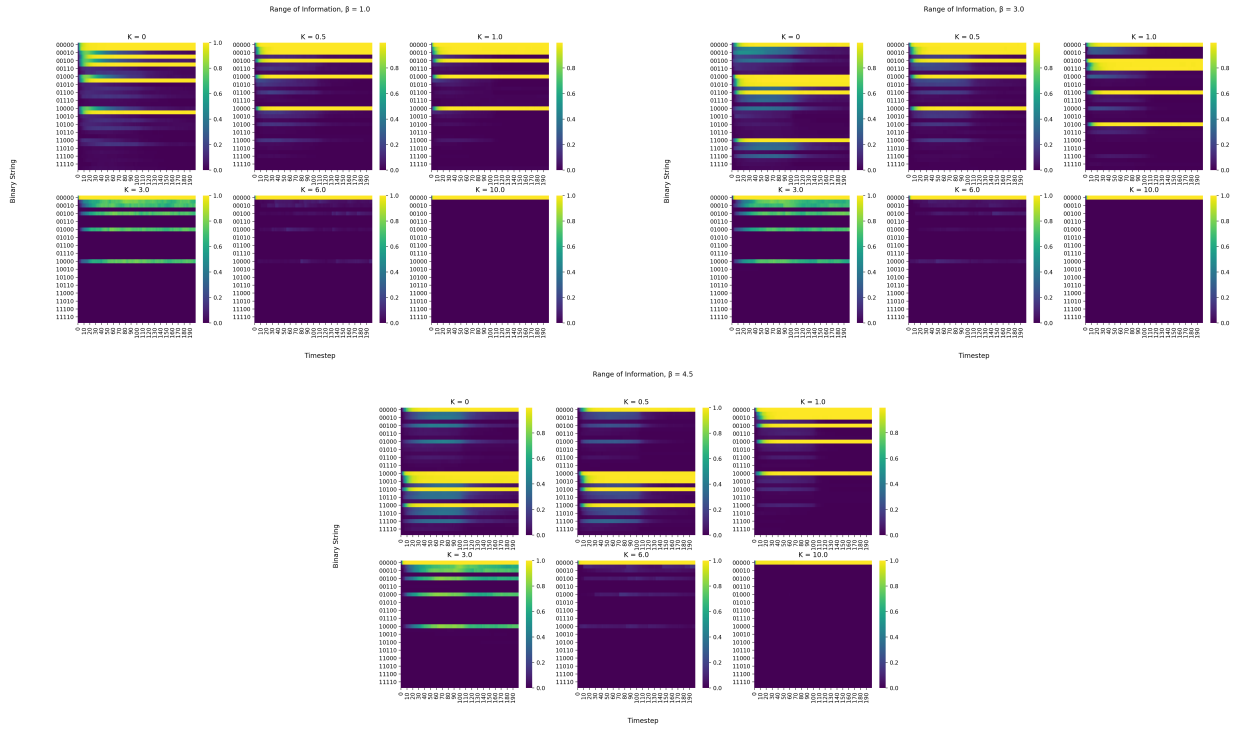


Figure 6: Range of Information Spread Plot for $K \in \{0, 0.5, 1, 3, 6, 10\}$ with (i) $\beta = 1.0$ (ii) $\beta = 3.0$ (iii) $\beta = 4.5$

## 3.3 Introducing an Adversary

The adversary we have designed for the Directed Graph based modified model is similar to adversary we had designed for the Original Rumour Spread model, (see Section 2.3). The only difference is that the top $r$ nodes are selected based on the highest out-degrees (since the definition of degree in the case of a directed graph brings about some ambiguity). The intuition behind the adversary design is also the same as before, since the chosen directed graph also exhibits the scale-free property.

We now perform experiments by simulating an adversary on the modified rumour-spread model. For this we generate the underlying directed graph in the same we did for the experiments in Section 3.2. The memory size is kept as 100, and the binary string's length was set to 5. We perform simulations with multiple values of $K$ and $\beta$. We run each simulation starting with 10 nodes being sampled from the out-degree distribution, and then being given one occurrence 00000 in their memory. The simulation is run up till 2000 timesteps, and we introduce the adversary after the 500th timestep (since until then, rumours are assumed to be widely spread in the network). Also, note that we only show those simulations in which the majority opinion has shifted from the true opinion (before the introduction of the adversary), since if the majority opinion was the true opinion itself, then most of the individuals already believe that the true opinion is indeed true. The code for the experiments is present here - `https://github.com/tymefighter/RumourSpread/tree/main/DirectedGraph`.

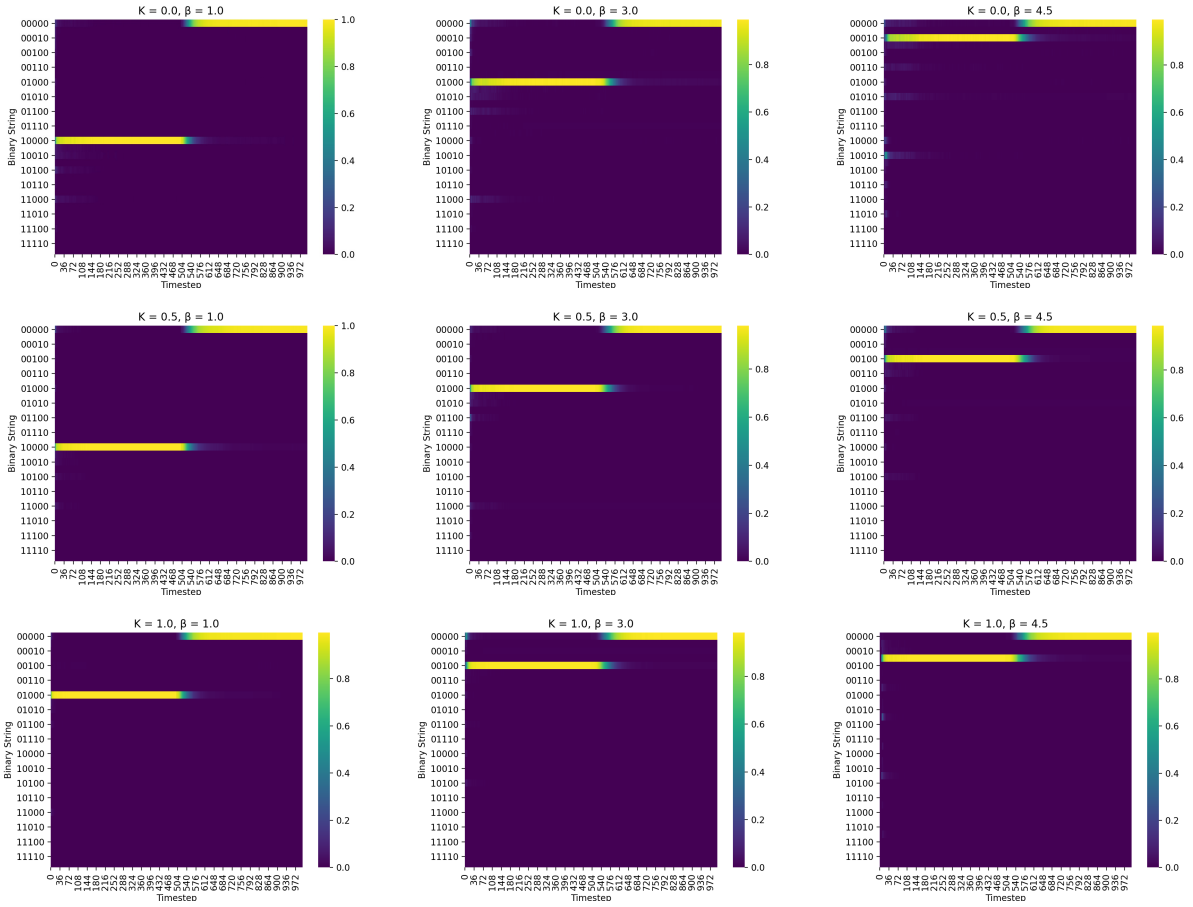


Figure 7: Simulation with Adversary - Opinion Fragmentation Plot for different $K$ and $\beta$ (plot is annotated with the values), the corresponding values of $r$ was set as (i) $r = 50$ (ii) $r = 10$ (iii) $r = 10$ (iv) $r = 50$ (v) $r = 10$ (vi) $r = 10$ (vii) $r = 50$ (viii) $r = 10$ (ix) $r = 10$

From the plots in Figure 7, we can clearly see that for different values of $K$ and $\beta$, we were able to manually set the value of $r$, which led to the adversary being able to divert the opinion of many nodes from a false information to the true one. We notice that only when $k = 0$, we required $r$ to be as high as 50, while for others $r = 10$ works well.

# 4 Analyzing Node and Edge Information Entropy

In this section we perform steady state analysis of the **modified** model (which we may also call as the "**system**") by computing the information entropy distribution of the nodes' memories, and the information entropy distribution of edges. Both of these quantities would be described in detail, and computed ahead. Before that we define the notion of the "steady state",

**Definition 5** (Steady State)**.** The rumour spreading system is said to have achieved a steady state if the average entropy of the system does not change much with time.

## 4.1 Node Information Entropy at Steady State

We present the definitions of the node information entropy and the node information entropy distribution, and their interpretation with respect to the spreading of rumours.

**Definition 6** (Node Information Entropy)**.** The information entropy of a node $n$ at the start of timestep $t$ is the entropy of the frequency-distribution of binary strings present in the memory of node $n$ at the start of timestep $t$.

**Definition 7** (Node Information Entropy Distribution)**.** The node information entropy distribution of the system at the start of a timestep $t$ is the distribution of entropy values achieved by nodes in the graph. It can be estimated by plotting the histogram or KDE (Kernel Density Estimation) of the list of entropy values of all nodes at the start of timestep $t$.

The node information entropy distribution at the steady state provides us with the amount of "randomness" present in the nodes' memories in the steady state. If the entropy distribution has higher density at lower entropy values, then the system has fewer variety of rumours in the node memories, but if it has a higher density at higher entropy values, then the system has larger variety of rumours in the node memories.

## 4.2 Edge Information Entropy at Steady State

We present the definitions of the edge information entropy and the edge information entropy distribution, and their interpretation with respect to the spreading of rumours.

**Definition 8** (Steady State Edge Information Entropy)**.** The edge information entropy of a directed edge $(u, v)$ at steady state is the entropy of the probability distribution of a binary strings which would pass through $(u, v)$. The same definition can be elaborated mathematically as follows - Pick any timestep $t$ after steady state has been reached, let $-1$ denote that no binary string was transferred from $u$ to $v$ (i.e. $v$ does not accept the binary string sent by $u$), let the probability of $i \in \{-1, 0, 1, ..., 2^s - 1\}$ being sent from $u$ to $v$ be $P_{uv}(i)$, then the edge information entropy $H_{uv}$ of $(u, v)$ is defined to be,

$$H_{uv} = H(P_{uv})$$

The Steady State Edge Information Entropy of an edge $(u, v)$ can be estimated by first choosing two timesteps $t_{start}$ and $t_{end}$ such that the average entropy has stopped fluctuating before timestep $t_{start}$ has been reached and, $t_{start}$ and $t_{end}$ are far apart from each other. Now, the entropy of the frequency distribution of a binary string (or -1, i.e. nothing) being transferred from $u$ to $v$ computed from $t_{start}$ to $t_{end}$ is an estimate of the steady state information entropy of the edge $(u, v)$.

**Definition 9** (Steady State Edge Information Entropy Distribution)**.** The edge information entropy distribution of the system at the steady state is the distribution of entropy values achieved by edges in the graph. It can be estimated by plotting the histogram or KDE (Kernel Density Estimation) of the list of entropy values of all edges at timestep $t$.

The edge information entropy distribution at the steady state provides us with the amount of "randomness" due to information being transferred along the edges. If the entropy distribution has higher density at lower entropy values, then fewer variety of rumours are communicated along the edges in the steady state, but if it has a higher density at higher entropy values, then larger variety of rumours are communicated along the edges in the steady state.

## 4.3 Experiments

We now compute the node information distribution and edge information distribution at a steady state which is achieved before the adversary is introduced, and at the steady state which is achieved long after the adversary is introduced. The experimental setup is similar to the one used in Section 3.3, except that the adversary is introduced at the end of timestep 1000. The first time the node entropy distribution is calculated is at the start of timestep 1000 (since steady state is assumed to have been attained), then the second time the entropy distribution is calculated is at the start of timestep 2000 (since steady state is assumed to have been attained again). The first time the edge entropy distribution is estimated by estimating the entropy of each edge from $t_{start} = 500$ to $t_{end} = 1000$, the second time the edge entropy distribution is estimated from $t_{start} = 1500$ to $t_{end} = 2000$. The code for the experiments is present here - `https://github.com/tymefighter/RumourSpread/tree/main/NodeAndEdgeInformation`.

The plots for $K = 0, \beta = 1, r = 50$ are shown in Figure 8, for $K = 0.5, \beta = 3, r = 10$ are shown in Figure 9, and for $K = 1, \beta = 4.5, r = 10$ are shown in Figure 10.

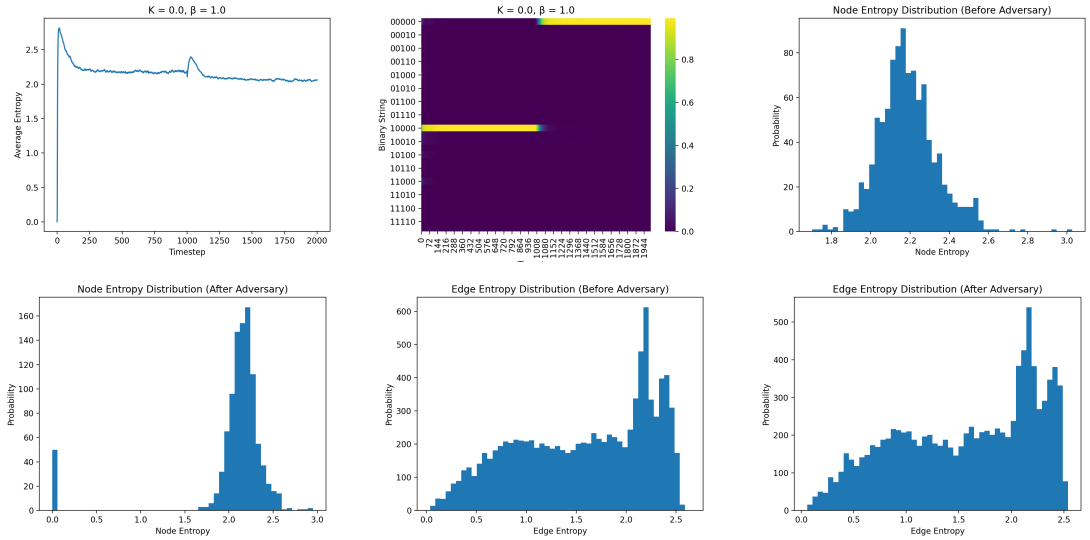


Figure 8: Experiment with $K = 0, \beta = 1, r = 50$, Figures are - (i) Average Entropy (ii) Opinion Fragmentation (iii) Node entropy measured at timestep 1000 (before adv.) (iv) Node entropy measured at timestep 2000 (after adv.) (v) Edge entropy measured from timestep 500 to 1000 (before adv.) (vi) Edge entropy measured from timestep 1000 to 2000 (after adv.)

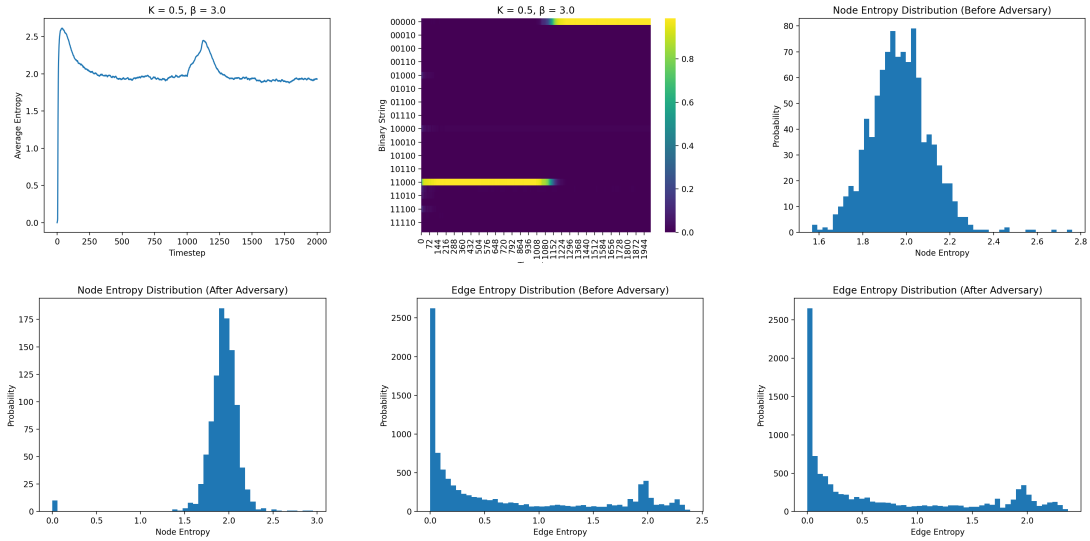


Figure 9: Experiment with $K = 0.5, \beta = 3, r = 10$, Figures are - (i) Average Entropy (ii) Opinion Fragmentation (iii) Node entropy measured at timestep 1000 (before adv.) (iv) Node entropy measured at timestep 2000 (after adv.) (v) Edge entropy measured from timestep 500 to 1000 (before adv.) (vi) Edge entropy measured from timestep 1000 to 2000 (after adv.)

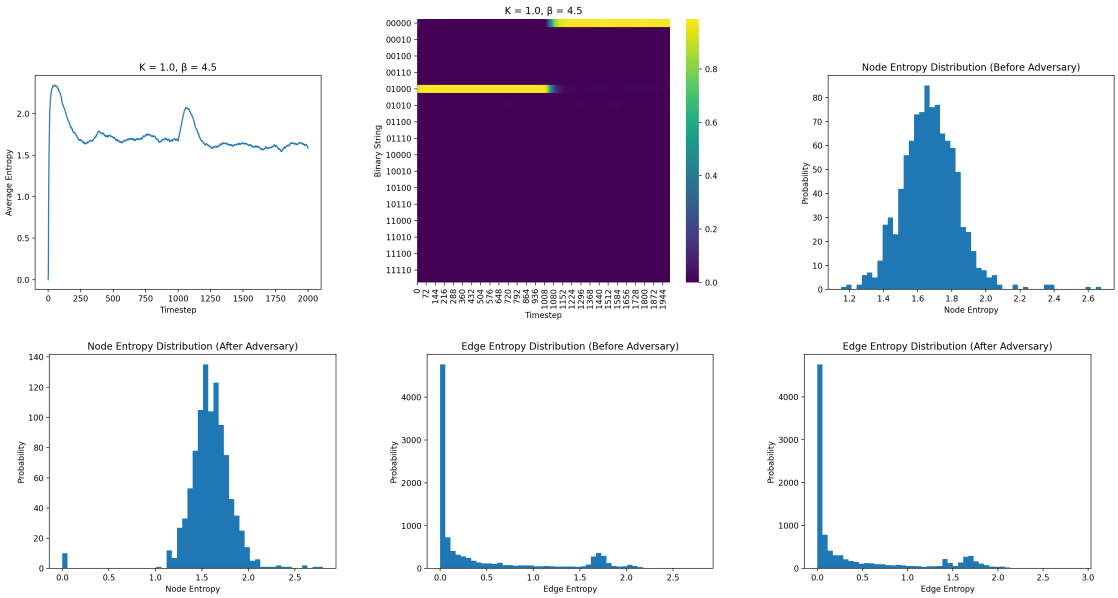


Figure 10: Experiment with $K = 1, \beta = 4.5, r = 10$, Figures are - (i) Average Entropy (ii) Opinion Fragmentation (iii) Node entropy measured at timestep 1000 (before adv.) (iv) Node entropy measured at timestep 2000 (after adv.) (v) Edge entropy measured from timestep 500 to 1000 (before adv.) (vi) Edge entropy measured from timestep 1000 to 2000 (after adv.)

In all these plots we notice that after the introduction of the adversary, the average entropy first increases and then stabilizes at a slightly lower level, this can be explained as follows - the adversary makes the true information to be provided by the selected $r$ nodes, when the false information has already dominated in the network, hence for a small period of time both true and the false information is present in the network, which leads to a higher entropy. Also, in all these plots we notice that the node entropy distribution becomes thinner and centers at a higher entropy after the adversary, when compared with the node entropy distribution before the adversary. We notice in Figures 9 and 10, the edge entropy distribution has a high probability near 0, both before and after the adversary, except that after the adversary, the probability density of higher entropy values has been reduced. But, Figure 8 is unique in the sense that the edge entropy has a well-spread distribution, both before and after the adversary, this is due to the higher distortion probability.

# 5 Conclusion

We have studied the original rumour spread model, and designed an adversary for it such that it would divert the opinion of most of the individuals in the network from false information to the true information. Then, we have described a limitation of using an undirected scale-free graph as the model of the underlying network, and proposed to use a directed scale-free graph instead. Then, we introduced an adversary for this modified model, which was able to divert the opinion of most of the individuals in network towards the true information. Finally, we have performed a steady-state analysis of the modified model, both before introducing the adversary, and after introducing the adversary.

# References

[1] Albert-Laszlo Barabasi and Reka Albert. Albert, R.: Emergence of Scaling in Random Networks. Science 286, 509-512. *Science (New York, N.Y.)*, 286:509–12, 11 1999.

[2] Béla Bollobás, Christian Borgs, Jennifer Chayes, and Oliver Riordan. Directed Scale-Free Graphs. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '03, page 132–139, USA, 2003. Society for Industrial and Applied Mathematics.

[3] Indian Express. Elon Musk - Use Signal, 2021.

[4] South China Morning Post. Trump Supports Spread Coronavirus Rumours, 2020.

[5] India Today. WhatsApp Privacy Policy Update, 2021.

[6] 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, Aug 2017.