The Grapevine Web: Analysing the Spread of False Information in Social Networks with Corrupted Sources

Jacques Bara
University of Warwick
Coventry, United Kingdom
jack.bara@warwick.ac.uk

Charlie Pilgrim
University of Warwick
Coventry, United Kingdom
charlie.pilgrim@warwick.ac.uk

Paolo Turrini
University of Warwick
Coventry, United Kingdom
p.turrini@warwick.ac.uk

Stanislav Zhydkov
University of Warwick
Coventry, United Kingdom
s.zhydkov@warwick.ac.uk

ABSTRACT

We study the problem of noisy information propagation in networks, where a small number of sources send messages across the network and agents use Bayesian updates to make inferences about the state of the world from the received messages. We provide upper bounds on the total number of sources necessary for learning on a given network and refine the bound in the case of small-world networks. We then extend the model to include an adversarial attacker, who can corrupt some of the information sources. We find that there is an optimal greedy attacking strategy in the case of a single learner, while the multi-learner case is not always solved optimally using greedy approaches. However, despite the influence function not being submodular, we show that the greedy algorithm performs well in practice. We also show that much simpler heuristics, which only look at centrality measures, can also provide a good basis to calculate successful attacking strategies. Finally we analyse the loss of optimality in the case when the attacker has incomplete information about the network and has to estimate the influence of source corruption heuristically. We use real-world social networks, as well as random network models, to empirically evaluate the effectiveness of attacking strategies and suggest a variety of measures to counteract them.

KEYWORDS

Social Networks; Misinformation; Greedy Algorithm; Heuristics; Centrality Measures

ACM Reference Format:
Jacques Bara, Charlie Pilgrim, Paolo Turrini, and Stanislav Zhydkov. 2023. The Grapevine Web: Analysing the Spread of False Information in Social Networks with Corrupted Sources. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 9 pages.

1 INTRODUCTION

Social networks have revolutionised the way we learn about each other and the world. While making the spread of information easier and faster than ever, they also pose a unique set of challenges when it comes to mis-(and dis-)information in society. Each piece of news, even those originating from a reliable source, has the potential to be altered and/or misinterpreted as it travels through the network towards its recipient. Recovering an objective truth about an event can quickly become problematic in such a noisy environment.

A stark recent example of the dangers of misinformation is the rise of mistrust in scientific expertise, in particular, regarding the effects of vaccination [16]. While the reliable sources, such as peer-reviewed medical research publications, give a clear consensus about the benefits of vaccination, this message is often distorted [27], whether deliberately or through ignorance and bias [22], contributing to a significant movement against such consensus.

There is also growing evidence of manipulation of such information systems strategically in order to instil doubt and create divisions within society; for instance, the use of bots on social media can be weaponised by malicious agents to steer the social [25] or political [26] discourse. While deliberate external attacks and random perturbations plant the seed of misleading and harmful information, the social networks themselves, combined with human psychology, have served as a conduit to greatly amplify its spread [1]. These problems call for a systematic study of social networks and information propagation within them.

In a recently proposed model, Jackson, Malladi and McAdams [15] have shown how a Bayesian learner can infer the reliability of information spreading through a tree from sources – a "Grapevine" learning model – showing how the structure of the tree affects learning. In this model a predetermined set of sources, placed at the leaves of the tree, transmit truthful messages, which can however mutate or disappear as they travel through the network. The learner, sitting at the root node, applies Bayesian reasoning to this data to form a posterior belief about the objective state of the world.

The Grapevine model is an important step in understanding how information flows from sources to learners in a Bayesian framework. However, it only considers trees and not networks, and it does not study the effect of information manipulation, which is crucial for the design of trustworthy social networks.

Contribution. In this paper, we extend the Grapevine model [15] to allow a malicious external attacker to corrupt a set of sources and have them transmit false information to multiple Bayesian learners in a social network. The question we are concerned with is whether the attacker can efficiently select the optimal subset of sources to corrupt in order to reduce the probability that the learner assigns to

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom. © 2023 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

the truth. We show that in the single-learner case, even if sources are heterogeneously distant, the set of nearest sources are always the optimal choice for the attacker. In the multiple-learner setting, where both learners and sources exist on a network, the notion of 'nearest' source breaks down at the global level. In this case, the optimal set is difficult to compute, and while a greedy algorithm will perform fairly well, we show that it is not guaranteed to be optimal even in small networks.

Paper Structure. In Section 2, we present previous work on the topics of information propagation, opinion dynamics, manipulation in networks and related problems. In Section 3, we provide the necessary mathematical background as well as some basic facts about the Grapevine model. In Section 4, we present and discuss our extension to networks and formulate the problem we are interested in. Section 5 provides the main results of the paper. Firstly, we prove that, in the one learner case, manipulation of the sources is easy. We then show that, in a network with multiple sources and multiple learners, manipulation is non-trivial and provide empirical evidence to support this claim. Lastly, we discuss the implication of our work and future directions in Section 6.

2 RELATED WORK

A vast amount of literature in multi-agent systems and theoretical computer science deals with the problem of opinion diffusion and its manipulation. Notably the seminal Kempe et al. [18] advances the understanding of influence maximisation from an initial set of triggering nodes. In particular, for innovation propagation on a network, they asked which influential individuals are the best to select in order to trigger a viral spread of innovation or information. However, their sole focus was to analyse whether the initial message has *reached* each node in the network, while we consider whether each node has *learned* the ground truth via a complex Bayesian learning process.

Information cascades can be understood as how a message / opinion / innovation can suddenly take root and quickly spread across a portion of the network [29]. In contrast, we do not necessarily try to model viral phenomena that spread quickly, and instead focus on individuals learning informed opinions. Nevertheless, these models look at the spread of information and highlight the importance of choosing the influential "seeds" - the original sources of information.

Young [30] considers more complex types of diffusion of ideas and innovations, such as contagion, social influence and social learning. Our work, however, assumes no opinion aggregation in the intermediate nodes, instead the complexity in our case arises from the more complex Bayesian aggregation of sources at each learner. This assumption is often avoided in favour of more simplified aggregation ([13] and the models mentioned above) or tackled with rather complex learning models [3]. We primarily focus on a variation on the latter, social learning, where learners make an informed, principled choice given a variety of messages from a variety of sources.

In an attempt to connect information diffusion and decision-making in society, the link between social influence [28] and collective decisions was studied by López-Pintado and Watts [20]. In a similar context, Kameda et al. [17] discussed the importance of finding the most influential nodes since they affect the collective decision-making process the most.

Manipulation of information systems has also been subject to extensive study. Bredereck and Elkind [9] consider several modes of manipulation in a classical majority opinion model, including bribing agents, changing the topology of the network and tempering with the model itself. In this paper, we only consider the "corruption" aspect of manipulation. Auletta et al. [4] provide a time complexity analysis of finding a set of agents that leads to consensus in a simple opinion diffusion model. They show that in general this problem is NP-hard, which foreshadows the hardness of the problem in our case. Borgs et al. [8] study how mistrust propagates through an influence system, in particular, using Google's PageRank algorithm.

Alon et al. [2] consider a very similar setting, where the ground truth is binary and the experts can be corrupted by an adversary in order to disseminate falsehood in the network. Again, the setting differs from ours by using a simple majority opinion diffusion model. Grandi et al. [14] develop the idea of corrupting the reviewers (for example, users of an online review system) by modelling whether it is profitable for the attacker to bribe the reviewers of their service. They study robustness of such review networks according to their topology, as well as under different assumptions of incomplete knowledge. Faliszewski et al. [12] combine manipulation in voting and opinion diffusion on social networks by considering the computational complexity of "campaigning", i.e., influencing specific nodes in the network to affect the outcome of an election. We note that the results in this paper can be similarly extended to the setting of elections by assuming that the state of the world is not objective but subjective, such as a political affiliation. In this case corruption of sources may correspond to a bribing of influential sources from the opposing party.

Our work is connected to adversarial attacks on learning models, with further links to path disruption games [5]. The problem of choosing the best subset of influential sources is a particular case of the general problem of Subset Selection, which arises in many applications. Qian et al. [24] study Subset Selection in the setting where the objective function is noisy. They provide the approximation guarantees for Greedy and POSS in the noisy setting and introduce a new algorithm PONSS which is able to handle the noise.

Greedy has been a popular heuristic in dealing with set functions in general. One of the first theoretical approximation guarantees of Greedy was provided by Nemhauser et al. [23], where the objective function is assumed to be *submodular*, which is the same property used in [18] to show a 63% approximation bound for Greedy in influence maximisation. A recent work by Bian et al. [6] extend this approximation analysis by looking at more general properties of the objective function, submodularity ratio and generalised curvature.

3 PRELIMINARIES

The Grapevine Model. We first describe the information propagation model by [15] which describes how simple messages propagate from information sources to a learner, who then collects the messages and uses them to learn about the state of the world. The messages may not always reach the learner directly, but through the "Grapevine" by passing through other agents. Each transmission has a chance of mutating the message or losing it altogether.

An important feature of this model is that, in contrast to many other opinion dynamics models, the probability of mutation is independent of the agent who is passing the message, i.e., intermediaries do not impose their own bias. In addition, intermediate agents do not aggregate the messages and instead transmit all the received messages.

Let T be a tree, rooted at r, which has m leaves. We call r *the learner* and leaves *the sources*, denoted by $S = \{s_1, \dots, s_m\}$. We then call the unique path from r to each s_i a *chain*. The number of edges in this chain is the *distance* between r and s_i and is denoted $d(r, s_i)$, or simply d_i when it is clear from the context.[1]

Firstly, assume the state of the world is given by a binary variable $\omega \in \{0, 1\}$, which intuitively represents whether some fact is true (e.g., “the new medical treatment is effective"). The prior probability that the state is 1 is given by $\theta := Pr(\omega = 1)$, which is known to the learner.

The process of message propagation proceeds as follows. Each source $s_i \in S$ starts by propagating the truthful message ω. The message then travels through the chain from s_i to r. At each intermediate node in the chain, the message is passed on with probability p_0 if it is currently 0 and p_1 if it is currently 1. If the message is dropped, it enters the null state, which we denote by $\emptyset$ and does not reach the learner. If it is passed, it has probability μ_{10} to change from 1 to 0, and similarly μ_{01} from 0 to 1.

The process of message propagation is an independent Markov chain with the state space $\{0, 1, \emptyset\}$ and the transition matrix M:

$$M = \begin{pmatrix} p_0(1-\mu_{01}) & p_1\mu_{10} & 0 \\ p_0\mu_{01} & p_1(1-\mu_{10}) & 0 \\ p_0 & p_1 & 1 \end{pmatrix}.$$

Markov processes are memory independent, meaning that the probability of going from one state to another does not depend on previous mutations. Hence, for a source $s \in S$ at distance d away from r transmitting a message in state j, the probability that the message ends up in state i at the learner is given by $(M^{d_s})_{ij}$, or $M^{d_s}_{ij}$ for brevity. Thus, the learner receives m messages (including the null ones), one from each source. We denote the vector of received messages by $I \in \{0, 1, \emptyset\}^m$.

Having received the vector of messages I, the Bayesian learner r updates her beliefs about the state of the world. Below, we list explicitly the knowledge available to the learner:

(1) The message vector I and the distance to every associated source $\{d_1, \dots, d_m\}$.
(2) The prior probability of the state of the world $Pr(\omega = 1) = \theta$.
(3) The parameters of the model, $p_0, p_1, \mu_{10}, \mu_{01}$.

We will show later that these assumptions can be relaxed slightly without losing any learning power. For example, the learner does not need to be aware of the null messages or even the probability of losing the message (if $p_0 = p_1 = p$). Jackson et al. [15] also provide an extensive discussion of the effects of incomplete knowledge under this model.

The learner now updates her beliefs by computing the posterior probability of ω being 1:

$$\begin{aligned} b_r(I) &:= Pr(\omega = 1 \mid I; \theta, (p_0, p_1, \mu_{10}, \mu_{01})) \\ &= \frac{Pr[I \mid \omega = 1]\theta}{Pr[I \mid \omega = 1]\theta + Pr[I \mid \omega = 0](1-\theta)}. \end{aligned} \tag{1}$$

We further need the following definitions and notation.

Set Functions. Let X be a set and $f : 2^X \longrightarrow \mathbb{R}$ be a function that assigns a real value to each subset of X. f is *increasing* if for all $A \subset B \subset X$, $f(A) \leq f(B)$. Similarly, f is *decreasing* if the inequality is reversed. If f is either increasing or decreasing, it is called *monotone*. We call f *submodular* if for all $A, B \subset X$, $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$. f is *subadditive* if for all disjoint $A, B \subset X$, $f(A) + f(B) \geq f(A \cup B)$. Clearly, if $f(\emptyset) \geq 0$, then every submodular function is subadditive.

Supermodularity and *superadditivity* are defined analogously, just with the inequalities reversed. Informally, submodularity represents diminishing returns when adding new elements to the set.

Centrality Measures. In this work we consider the following five centrality measures as heuristics to be used in picking influential sources to corrupt. In particular, for a budget of k, we will pick the k sources $s \in S$ with the highest centrality measure.

- **Degree** - Number of nodes connected to a source, Deg(s).
- **PageRank** - Google's PageRank [10] which takes into account quality and number of nodes connected to a source, PR(s).
- **Eigenvector** - Entries of the leading eigenvector of the adjacency matrix A of graph G, Eigv(s).
- **Closeness** - Inverse of the sum of geodesic distances [2] $d(n, s)$ from all other nodes $n \in V \setminus \{s\}$ to the source, $C(s) = (N-1)/\sum_{n \in V \setminus \{s\}} d(n, s)$.
- **Harmonic** - Sum of the inverse of geodesic distances $d(n, s)$ from all other nodes $n \in V \setminus \{s\}$ to the source, $H(s) = \sum_{n \in V \setminus \{s\}} d(n, s)^{-1}$.

The choice of these five centralities are due to a mixture of popularity and intuition. The degree is the simplest centrality and is highly localised. The Eigenvector and PageRank centralities are related to random walks of infinite length, which in turn can characterise epidemic/information spread on networks [21]. Finally the Closeness and Harmonic centralities both consider nodes ‘nearest’ to all others as the most important - these can be seen as heuristics to generalise the single-learner case in which sources nearest to the learner are the most influential.

4 GRAPEVINE NETWORKS

We extend the model presented so far by considering *multiple learners* on a graph. Consider a connected graph $G = (V, E)$ with node set V of size $|V| = n$ and a set of sources $S \subset V$. For each node $v \in V \setminus S$ we can construct a tree T_v, rooted at v such that T_v is a subgraph of G and the set of leaves of T_v is precisely S. Indeed there may be many ways to construct such a tree. We choose to construct T_v by joining shortest paths from v to each source $s \in S$, choosing an arbitrary path if there are multiple. In so doing we may treat each

[1] Jackson et al. [15] consider rooted trees in which all leaves are at the same fixed depth. We formally generalise this here for the sake of introducing notation.

[2] The geodesic distance between two nodes in a graph is defined to be the length of a shortest path between the nodes.

$v \in V \setminus S$ as a Bayesian learner as in the original model with the same message propagation and inference.

For the remainder of this paper, we assume $\mu_{01} < 0.5$ and $\mu_{10} < 0.5$ following [15], so it is more likely for the message to remain unchanged than to flip at any given node. For simplicity we also assume the probability of propagating the message is the same for 0 and 1, i.e., $p = p_0 = p_1$. Hence, an (uncorrupted) instance of the extended Grapevine model is completely characterised by the tuple $\mathcal{G} = (G, S, \mu_{10}, \mu_{01}, p, \theta)$.

4.1 Corrupted Sources

The original work [15] focused on understanding the conditions required for learning to occur. In particular, the results indicate how many sources and how much knowledge of the model is required for the learner to learn the correct information. The extension we introduce allows us to focus on a wider set of problems. We briefly revisit the question regarding the number of sources required for learning to occur in a network, given its size and topology, in Section 4.3. However, the main focus of this paper is to understand how learning can be disrupted by an adversarial attacker who can "corrupt" the sources.

In this paper we assume that the ground truth state, ω is always 1, without loss of generality. We hence define the *corrupted state* to be 0. Recall that each source is originally uncorrupted, i.e., they transmit 1 as the original signal. In order to measure the effect of corruption, we introduce the *influence* metric.

Given an instance of the Grapevine model, recall that $b_r(I)$ is the posterior probability learner r learns after receiving the message vector I. Define $\overline{b}_r := E_I[b_r(I)] = \sum_I b_r(I)Pr(I)$, where $Pr(I)$ the probability of r receiving I in this instance of the model. In other words, $\overline{b}_r$ is the expected learned probability of learner r. Lastly, define $\overline{b} := \frac{1}{n}\sum_{r\in V}\overline{b}_r$ to be the average expected learned probability across the network. $\overline{b}$ represents the expected average belief in the network that $\omega = 1$.

Given an instance $\mathcal{G}$ and a subset of sources $T \subseteq S$, we define a *corrupted* instance $\mathcal{G}_T$, in which every source $s \in T$ propagates the 0 message (while the rest still propagate the ground truth, 1). We call T a *corrupting set*. Analogically to $\overline{b}$, we define $\overline{b}^T$ to be the average expected learned probability in the corrupted case.

We now define the *influence* function $\sigma : 2^S \longrightarrow [0, 1]$ by

$$\sigma(T) = \overline{b} - \overline{b}^T, \tag{2}$$

i.e., by how much the average belief in 1 is expected to drop if sources in T were corrupted to transmit 0.

Finally, we model the attacker's desire to disrupt belief in 1 by maximising the influence, subject to how many sources they are allowed to corrupt. The resulting optimisation problem can be written succinctly as

$$\max_{|T|\leq k} \sigma(T), \tag{3}$$

where k is the *budget* of the attacker.

Note that the above definitions assume that an instance of the model $\mathcal{G}$ is fixed, hence the attacker's aim is to optimise the influence function on a particular instance. However, of course, we want to find algorithms that work on all or a wide selection of instances.

4.2 Simplified Posterior

Under some mild conditions, we can derive a more useful form of the learned posterior from Equation 1. From the equation, we can see that we only need to derive $Pr[I \mid \omega = 1]$ and $Pr[I \mid \omega = 0]$, i.e., the probability that the Markov process given by M produces the message string I from each ground truth state.

Firstly, recall that the probability that the original message $j \in \{0, 1\}$ at the source evolves into a message $i \in \{0, 1, \emptyset\}$ as it reaches the learner depends only on the transition matrix M and the distance between the source and the learner, d and is given by M^d_{ij}. Denote $\mu := \mu_{01} + \mu_{10}$. In general, for any μ and p as well as $d \in \mathbb{N}$, we have

$$M^d = \frac{1}{\mu}\begin{pmatrix} p^d(\mu_{10} + \mu_{01}(1-\mu)^d) & \mu_{10}p^d(1-(1-\mu)^d) & 0 \\ \mu_{01}p^d(1-(1-\mu)^d) & p^d(\mu_{01} + \mu_{10}(1-\mu)^d) & 0 \\ \mu(1-p^d) & \mu(1-p^d) & \mu \end{pmatrix}. \tag{4}$$

Since the sources can be grouped by the distance to the learner, let $x_i(d) = |\{s_k \in S \mid d_k = d \cap i_k = i\}|$ be the number of messages $i \in \{0, 1, \emptyset\}$ received by the learner from distance d. Since the Markov chains are independent and only depend on d, we have

$$Pr[I|\omega = j] = \prod_{i\in\{0,1,\emptyset\}}\prod_{d=0}^{D}(M^d_{ij})^{x_i(d)}.$$

We can now combine this with Equation 1 to derive the posterior $b_r(I)$. This can be further simplified into a form $Pr[w = 1|I] = (1 + \chi)^{-1}$, where χ^{-1} is the odds.

$$b_r(I) = Pr[w = 1|I] = \frac{1}{1+\chi}, \text{ where} \tag{5}$$

$$\chi \equiv \frac{1-\theta}{\theta}\prod_{d=0}^{D}\left[\left(\frac{\mu_{10} + \mu_{01}(1-\mu)^d}{\mu_{10}(1-(1-\mu)^d)}\right)^{x_0(d)}\left(\frac{\mu_{01}(1-(1-\mu)^d)}{\mu_{01} + \mu_{10}(1-\mu)^d}\right)^{x_1(d)}\right]. \tag{6}$$

Note that this expression does not depend on the propagation rate, p. This is due to the simplifying assumption of the symmetric propagation rate, which implies that null messages do not carry any information. Of course, low propagation rate still hurts learning as it means that few useful messages reach the learner, implications of which are discussed in Section 4.3.

Full derivation of the posterior is provided in the supplementary materials.

4.3 Learning Threshold

We now consider the implications of one of the main results for the original model:

Lemma 1 ([15]). *Suppose the distance between the sources and the learner is d. Then the learner needs at least $\ell(d) = \frac{1}{(p(1-\mu)^2)^d}$ sources to* learn *the ground truth.*

The expression $\ell(d)$ is called the learning threshold. Informally, *learning* the ground truth means that the learner's posterior tends to 1 as d increases, provided the learning threshold is met.

In our extended model, the learning threshold dictates how many sources a network needs as it grows in size. Since both the learners and the sources live on the network, the distance between them is bounded from above by the diameter of the network, D. Hence, $\ell(D)$

provides a (possibly loose) upper bound for the sufficient number of sources in the network.

Consider the case of small-world networks, where the diameter grows logarithmically with the network, i.e., $D = \alpha \log n$. This results in $\ell(D) = O(n^{\beta})$, where β depends on the parameters of the model, μ and p and the rate of growth of the diameter, α. Hence, the necessary number of sources is polynomial in the number of nodes; however, the order of the polynomial can be quite low if the mutation rates are low. Figure 2 in the supplementary material illustrates the effect of the number of sources on learning as the network grows.

5 COMPUTING OPTIMAL ATTACKS

The main aim of this paper is to analyse how susceptible the extended Grapevine model is to corrupting attacks. To do this, we analyse the complexity of computation of the optimisation problem given in (3), which represents the ability of the attacker to choose the best corrupting set.

The optimisation problem faces two difficulties. Firstly, the attacker might not be able to query the influence function efficiently. The definition of the influence of a set of sources involves computing the expected learner posterior over all possible received messages. Since the number of such messages grows exponentially with the number of sources, a single query $\sigma(T)$ may be infeasible to compute, if the source set is large.

Secondly, even if we assume that the attacker can query σ in polynomial time, there are $\binom{m}{k}$ possible sets of sources to corrupt. For any non-trivial size of parameters m and k, this renders the brute-force approach infeasible.

In this section, we show that, even when finding the exact solution is infeasible, heuristics provide a good approximation of the optimal solutions.

5.1 Single Learner

Theorem 2. *Choosing closest sources is always optimal in the single learner case.*

Proof. While here we describe the main intuitions behind the proof, the formal proof is provided in the supplementary material. Consider the problem of corrupting one of two sources, A or B, where $d_A > d_B$, given a set of existing corrupted and uncorrupted sources.

Without loss of generality, assume the world is in state $\omega = 1$. Maximising the influence function is equivalent to minimising the expected learned probability given the sources:

$$\bar{b}_T = \sum_I p(w = 1 \mid I) p(I \mid T)$$

The received message has a stronger effect on the posterior the closer it is to the learner, as there is less chance of it having been mutated from the true value (from the naive learner's point of view). A 1 increases the posterior probability of the world being in state 1, while a 0 decreases the probability. From the attacker's point of view, a closer distance increases the probability of a 0 being received and decreases the probability of a 1 being received. The attacker prefers a closer source on both counts, as a 0 is more likely to be received, and the learner will give more credence to that 0. Conversely, a 1 is less likely to be received, which would have a positive effect on the posterior. Note that the null messages have no effect on the posterior.

The attacker's preference for a closer source is independent of the existing set of corrupted sources. As such, a greedy attacker strategy of always choosing closest sources is optimal. □

5.2 Multiple Learners

We now use the full extent of our extended model and consider the problem of source corruption on a graph with multiple learners. Figure 1a gives a small example of the model, where the coloured nodes are the sources and the two grey nodes in the middle are the learners. In general, we are also interested in large social networks that can vary from thousands to millions of users. For example, for our experiments we use a subset of the Facebook network introduced by Leskovec and Mcauley [19], which contains over 4000 nodes. As discussed in Section 4.3, the number of sources required to ensure learning in the network is a polynomial fraction of the size of the network, but varies a lot according to the parameters of the model.

The optimisation problem given in (3) is an example of a general problem of subset selection, where the aim is to select the best subset according to an objective function subject to constraints. In the general case, this problem is NP-hard [11].

We first show that the problem is non-trivial only when the mutation rates are not too high so that learning actually occurs in the network.

Lemma 3. *If $\mu = 1$, no learning can occur. Hence, any algorithm is trivially optimal.*

Proof. Using Equation 6 with $\mu = 1$ gives $\chi = \theta^{-1} - 1$ and $b_r(I) = \theta$ for any message vector I. This means that all learners stay at their prior beliefs and no learning occurs. Full derivation is available in the supplementary material. □

A popular heuristic for achieving an approximate good solution is the Greedy algorithm. Starting with an empty set, at each iteration, an element is added to the set with the highest marginal contribution. The process stops when no additional element has a non-negative marginal contribution or when the size constraint is reached. We will now show that σ is non-decreasing, hence Greedy always outputs a set of size k.

Lemma 4. *σ is a non-decreasing set function.*

Proof. Suppose we have an instance of the Grapevine model $\mathcal{G} = (G, S, \mu_{10}, \mu_{01}, p, \theta)$. By the definition in Section 3, all we need to show is that $\sigma(\Omega \cup \{s\}) \geq \sigma(\Omega)$ for all $\Omega \subset S, s \in S$. Equivalently, by the definition of σ given in (2), we need to show that $\overline{b}_{\Omega \cup \{s\}} \leq \overline{b}_{\Omega}$.

Claim. Corrupting an extra source reduces the expected learned posterior for *every* agent in the network.

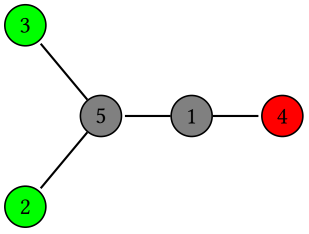
Proof. A full proof is given in the supplementary material, we simply note that it requires two steps: 1) flipping a source from 1 to 0 increases the probability of the learner to receive a 0; 2) receiving a 0 instead of 1 reduces the posterior belief of the learner. Since the $\overline{b}$ is simply the average over all learners, this completes the proof. □

Algorithm 1 Greedy Algorithm

```
Input: Grapevine model instance $\mathcal{G}$, budget $k$
Output: Corrupting set $T \subset S$
Initialise $T \leftarrow \emptyset$
for i in $\{1, \dots, k\}$ do
    $S' \leftarrow S \setminus T$                                ▷ Available sources
    $s^* \leftarrow \text{argmax}_{s \in S'}\ \sigma(T \cup \{s\})$ ▷ Choose source with highest marginal influence
    $T \leftarrow T \cup \{s^*\}$                               ▷ Update corrupted set
end for
return $T$
```

The Greedy algorithm is presented in Algorithm 1. Consider the time complexity of the algorithm. The algorithm performs k iterations, where iteration i requires $m - i$ evaluations of the influence function $\sigma(\cdot)$. In total, this requires $m + m - 1 + \ldots + m - k + 1 = k(m-k) + \frac{k(k+1)}{2} = O(km - k^2)$ evaluations. Note that evaluations of the influence function are often infeasible and even good empirical approximations can be costly. Hence, in most cases, this is a significant improvement on the brute-force search, which requires $O(\binom{n}{k})$ evaluations.

5.2.1 Optimality of Greedy. We now show that, in the case with multiple learners, Greedy does not necessarily output the optimal corrupting set.



(a) Example network.

T	$\sigma(T)$
{3}	0.0533
{2}	0.0533
{4}	0.0646
{4, 3}	0.2332
{2, 3}	0.2588
{4, 2}	0.2332
{4, 2, 3}	0.7134

(b) Influence of the possible sets.

Figure 1: An instance of the Grapevine model, on which Greedy does not find the optimal solution. Nodes 1 and 5 are the learners and nodes 2, 3 and 4 are the sources.

Example 5. Consider an instance of the Grapevine model with the graph given in Figure 1a and the following parameters: $\mu_{10} = 0.1353, \mu_{01} = 0.0386, p = 0.9299, \theta = 0.9297$. Let nodes 2, 3 and 4 be the sources and the attacker's budget be $k = 2$. Table 1b shows the influence of each possible corrupting set.

In the first round, Greedy selects source 4 as it has the highest influence. It then randomly selects between $\{4, 3\}$ and $\{4, 2\}$ as they have the same influence. However, the optimal pair to corrupt is $\{2, 3\}$, which is missed by Greedy.

When Greedy is not optimal, a common approach to providing approximation guarantees is by exploiting properties of the objective function, in particular, submodularity [23]. Unfortunately, the influence function arising from our model is neither submodular nor supermodular, which makes analysing the approximation guarantees of Greedy difficult.

Lemma 6. *Influence σ is neither submodular, nor supermodular.*

Proof. The proof of Lemma 6 is by counterexample. Take the system represented in Figure 1a, in particular, note the influence σ in Figure 1b. Recall the definition of submodularity from Section 3. Setting $A = \{2, 4\}$ and $B = \{3, 4\}$ violates the condition, hence σ is not submodular in this instance.

To show that in some instances σ may also be not supermodular, consider an instance with an odd number of sources, m and parameters $\mu_{01} = \mu_{10} = p = \varepsilon$ for some small ε. Intuitively, there is very little noise in the model, which is known to the learners, hence they trust the messages they receive completely. As $\varepsilon \longrightarrow 0$, we can see that $\sigma(T) \longrightarrow 1$ if $|T| \geq \frac{m}{2}$ and $\sigma(T) \longrightarrow 0$ otherwise. Now choose A and B such that $|A|, |B| \geq \frac{m}{2}$ – this violates the condition of supermodularity. □

5.3 Computing Influence

The ability of the attacker to identify the best sources for corruption is reliant on their ability to calculate the influence of a source set. Consider the definition of influence, $\sigma(T) = \overline{b} - \overline{b}^T$. Computing σ requires computing the average expected learned posterior in both the corrupted and the uncorrupted case. Since $\overline{b} := \frac{1}{n} \sum_{r \in V} \overline{b}_r$, this reduces to n evaluations of individual expected learned posterior, $\overline{b}_r$.

Since $\overline{b}_r = \sum_I b_r(I) Pr(I)$, computing this expected posterior requires iterating over all possible received vectors of messages by the learner. Since each message is either 0, 1 or $\emptyset$, and there are m sources, there are 3^m such vectors. Even if the computation of the posterior, given the message vector, can be done quickly, iterating over this number of possibilities grows infeasibly fast as the number of sources grows.

However, the simplified posterior given by (6) allows us to simplify the sum by grouping together the vectors of messages that produce the same posteriors. Since the posterior only depends on the number of 0- and 1-messages at a particular distance from the learner, all sources at the same distance are indistinguishable from each other.

Suppose $m = \sum_{d=1}^{D} m_d$, where m_d is the number of sources at distance d from the learner and D is the diameter of the network (and hence the greatest distance a source can be at). Since, at distance d, any permutation of the m_d messages leads to the same posterior, there are $\frac{1}{2}(m_d + 1)(m_d + 2)$ possible distinct message vectors. Then, the total number of equivalence classes of message vectors is $\prod_{d=1}^{D} \frac{1}{2}(m_d + 1)(m_d + 2)$.

Notice that when all sources have a distinct distance from the learner, each $m_d = 1$ and so the number of equivalence classes is once again 3^m: each message is distinct as it comes from a unique distance. On the other hand, when all sources are at the same distance, the number of classes is reduced to $\frac{1}{2}(m + 1)(m + 2)$: all that matters to the learner is the number of 1s and 0s received.

Hence, in the best case scenario, the computation of the expected posterior of a learner can be polynomial in the number of sources, namely $O(m^2)$. In the worst case, however, the computation is still $O(3^m)$, while a realistic scenario is somewhere in-between: it is exponential but only in the number of distinct distances to sources, which is in turn bounded by the diameter of the network, D.

5.3.1 Approximating Influence with Simulations. When the network and the number of sources are both large, direct computation of the influence becomes infeasible and thus a common approach is to compute an estimate using repeated simulations of the diffusion process given by our model. This process is fast: we only need to precompute M^d as given in 4 and then sample each of the m messages independently. This samples a message vector I with the true probability of r receiving it, so repeated sampling approximates the expected posterior probability, $\overline{b}_r$.

Figure 1 in the supplementary material shows the accuracy of the estimates as a function of the number of samples, N. Even with $N = 10$, most estimates are within 0.3% of the true influence. Larger sample sizes do narrow the spread of the estimates, although with diminishing returns - using more than $N = 100$ samples is unnecessary for most applications. This shows that fast and accurate estimates of the influence function are available to the attacker, provided they know the parameters of the model.

5.4 Empirical Analysis

5.4.1 Experimental Setup. We conducted several simulation-based experiments to test the effectiveness of various attacking strategies. The strategies are all based on the greedy approach. The Greedy algorithm given in Algorithm 1 uses empirical estimates of the influence function at each iteration of the algorithm. All other algorithms are based on a specific centrality measure and simply select a set of k sources with highest centralities. Where possible we compared the heuristics against the optimal solution, computed either analytically or empirically. Where computing the optimum is infeasible, we compared the heuristics against each other.

5.4.2 Experiment on the Facebook Network. In Experiment 1, we used a subset of the Facebook friendship network, provided by Leskovec and Mcauley [19]. It is an undirected graph that has around 4000 nodes, which represent Facebook users, and around 88000 edges, which represent the friendship relation. We generated 667 unique instances on this network as follows. We select between 5 and 9 nodes at random to serve as sources. Then we choose $\mu_{10}, \mu_{01} \in [0, 0.5]$ and $p, \theta \in [0.5, 1]$, each uniformly at random. This characterises an instance of the Grapevine model.

Now we run 100 simulations of the diffusion process to estimate $\sigma(T)$, for each $T \subset S$. Then, for each $k \in \{2, ..., m-1\}$, we record the highest, lowest and average influence for sets of size k. We then run the algorithms with the budget of k and normalise the influence of their output as to be between the optimal and the worst choice of the corrupting set. This produces the score for each algorithm between 0 and 1, where 0 selects the corrupting set with the lowest influence and 1 with the highest.

The empirical distribution of the scores for each algorithm on the Facebook network are shown on the right plot in Figure 2. Each curve illustrates the number of instances where the algorithm performs above a threshold. For example, the Greedy algorithm is only sub-optimal in around 5% of the cases, while using PageRank centrality is only optimal in 10% of the instances.

The results highlight the power of the Greedy algorithm. While there are instances where Greedy only achieves 60% of the optimal value, 95% of the time it outputs the optimal set in only $O(km - k^2)$ evaluations of the influence function.

However, when evaluation of the influence function is infeasible or even impossible due to incomplete knowledge of the model parameters, relying on evaluation-free methods does not jeopardise the performance much. Choosing the nodes with highest harmonic centrality will yield an optimal result in over 80% of the cases, while closeness centrality achieves the optimum 75% of the time.

A more complex method of estimating centrality, PageRank, does not seem to correlate with influence very well. It is only optimal in 10% of the cases, while in 35% of the cases it achieves a below-average score. Eigenvector centrality provides somewhat better results, although it is still significantly worse than the shortest path-based measures.

On the other hand, a simple method of choosing the sources with highest degree, while outperforming PageRank slightly, also falls behind the other heuristics. This suggests that knowing the topology of the network is very important for the attacker, even if the parameters of the model are unknown.

We also run the same experiment on two smaller ($n = 200$) networks, one generated using the Erdős–Rényi model and the other - using Watts-Strogatz [7]. While the results show a similar pattern for the algorithms, an interesting observation is that centrality measure-based algorithms all provide the same results on the Erdős–Rényi graph. This suggests that the topology generated by this random model is not well-described by the centrality measures.

5.4.3 Experiment on a Watts-Strogatz Graph. In Experiment 2, our aim was to test the performance of the algorithms when the number of sources is large. We have used a smaller network to make the simulations computationally feasible. To this extent, we generated a random graph using the Watts-Strogatz model with $n = 200$ nodes, the mean degree of $K = 10$ and the rewiring probability of $\beta = 0.1$.

For each value of $k \in [3, 19]$, we generated 50 unique instances by selecting $m = 20$ sources at random and setting parameters in the same way as in Experiment 1.

With this number of sources, a brute-force search of the optimal solution becomes infeasible. Hence, we compare the Greedy algorithm and its heuristic variants. We also select one set uniformly at random to serve as a benchmark. We measure the performance of the algorithms by normalising the influence of their output by the influence of the random output.

Figure 3 shows the performance comparison of the algorithms. We observe a similar story to Experiment 1. Greedy paired with the empirical evaluations of the influence function performs best, beating the random choice by more than 30%, on average, when selecting $k = 4$ corrupted sources. However, this advantage diminishes when selecting a large number of sources to corrupt, down to around 3% when selecting 18/20 sources. This can be explained by the submodular behaviour of the influence when corrupting significantly more than half of the sources: at some point the attacker has already convinced the learners enough (i.e., reduced their posterior belief close to 0), so that corrupting more sources has little impact.

In the cases where selecting a good corrupting set matters, similarly to Experiment 1, using closeness and harmonic centrality works best when considering evaluation-free algorithms. Once again, PageRank and degree centrality lag behind in performance, but nevertheless outperform the random choice on average. This is an important consideration for the degree-based algorithm as it

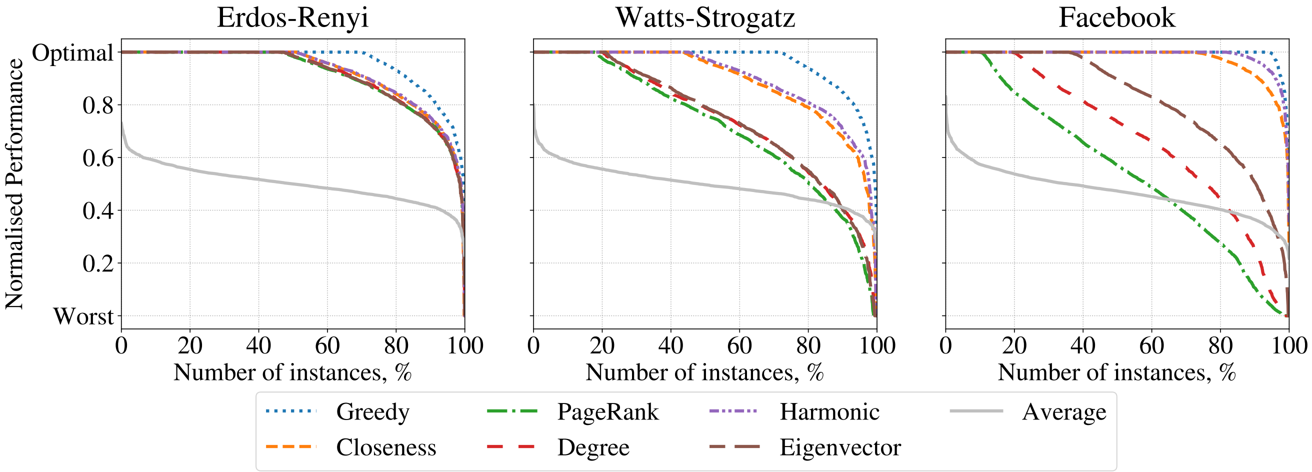


Figure 2: Empirical distribution function of the algorithms' performances on each corresponding network.

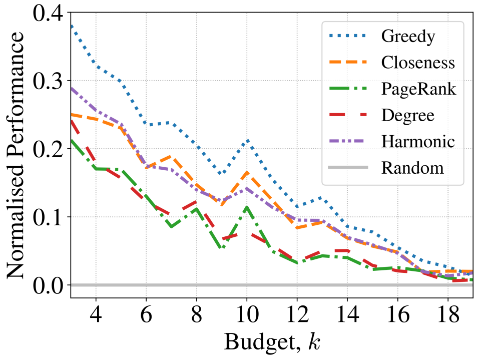


Figure 3: Mean normalised performance of the algorithms on a Watts-Strogatz network with $m = 20$ sources as a function of the budget, k.

assumes very little knowledge about the network, while still giving an edge over a random choice.

6 DISCUSSION

In this paper, we extended Jackson, Malladi and McAdam's Grapevine model [15], originally proposed for a single Bayesian learner on a tree, to a multi-learner model on a network. In Section 4.3 we generalised their main result by deriving the number of sources necessary for learning as a function of the diameter of the network, moreover providing an analytic form for small-world networks.

We then focused on the problem of a coordinated attack on the sources, which corrupts them to share false information. We show in Theorem 2 that, in the single learner case, the attacker has a simple optimal strategy: pick the sources closest to the learner. In contrast, the general case, where multiple learners form a network, cannot always be solved optimally by the Greedy algorithm, even if the attacker can evaluate the influence function exactly (Example 5).

We then analyse the effectiveness of Greedy attacking strategies in Section 5.4: one that uses empirical evaluations of influence, and several that only rely on centrality measures as a heurisitic. We test these approaches against the optimal (where feasible) or random choice on a subset of the Facebook network as well as two random graph models: Erdős–Rényi and Watts-Strogats.

All experiments show that Greedy approaches work well in practice and their effectiveness depends on the information available to the attacker. If evaluations of influence are available, Greedy is almost always optimal on the Facebook network. Selecting nodes based on the Closeness or Harmonic centrality can also be an effective strategy, while being ignorant of model parameters. Even selecting sources based only on their degree provides an edge over random selection.

This work paves way for many future directions. The Bayesian approach to learning is a realistic extension of many information diffusion models, yet it clearly introduces complexity in its analysis. For instance, it gives rise to a complex influence function, which characterises the effect of a set of sources on the learning process. While we provided justification for the complexity of computation of the influence, a theoretical hardness guarantee would provide security by making efficient attacks more difficult.

Similarly, the complexity of influence makes it harder to provide theoretical guarantees of the Greedy algorithm, while empirical results suggest that such guarantees should be possible. This calls for further investigation of Greedy approximations for a more general class of functions, which exhibit submodularity and supermodularity features at the same time. In our case, a good starting point would be applying the work of Bian et al. [6] to the influence function arising from a specific instance of the Grapevine model.

There is also more work to be done in examining knowledge assumptions in the model. For instance, assuming heterogeneous probability of message propagation (i.e., $p_0 \neq p_1$) introduces an aspect of learning purely from the number of received messages . While assuming that the learners (or the attacker) lack the knowledge of model parameters can lead to a simpler learning process, but may also make it more difficult for the attacker to create a sophisticated strategy.

ACKNOWLEDGMENTS

This work was supported by Engineering and Physical Sciences Research Council grant EP/S022244/1 for the MathSys CDT.

REFERENCES

[1] Daron Acemoglu, Asuman Ozdaglar, and Ali ParandehGheibi. 2010. Spread of (mis)information in social networks. *Games and Economic Behaviour* 70, 2 (2010), 194 – 227.

[2] Noga Alon, Michal Feldman, Omer Lev, and Moshe Tennenholtz. 2015. How Robust is the Wisdom of the Crowds?. In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI)* (Buenos Aires, Argentina) *(IJCAI'15)*. AAAI Press, 2055–2061.

[3] Marco Amoruso, Daniele Anello, Vincenzo Auletta, Raffaele Cerulli, Diodato Ferraioli, and Andrea Raiconi. 2020. Contrasting the Spread of Misinformation in Online Social Networks. *J. Artif. Intell. Res.* 69 (2020), 847–879. https://doi.org/10.1613/jair.1.11509

[4] Vincenzo Auletta, Diodato Ferraioli, and Gianluigi Greco. 2020. On the complexity of reasoning about opinion diffusion under majority dynamics. *Artif. Intell.* 284 (2020), 103288. https://doi.org/10.1016/j.artint.2020.103288

[5] Yoram Bachrach and Ely Porat. 2010. Path disruption games. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. Toronto, Canada, 1123–1130.

[6] Andrew An Bian, Joachim M Buhmann, Andreas Krause, and Sebastian Tschiatschek. 2017. Guarantees for greedy maximization of non-submodular functions with applications. In *International conference on machine learning*. PMLR, 498–507.

[7] Béla Bollobás and Oliver Riordan. 2008. *Random Graphs and Branching Processes*. Springer Berlin Heidelberg, Berlin, Heidelberg, 15–115. https://doi.org/10.1007/978-3-540-69395-6_1

[8] Christian Borgs, Jennifer Chayes, Adam Tauman Kalai, Azarakhsh Malekian, and Moshe Tennenholtz. 2010. A novel approach to propagating distrust. In *Proceedings of the 6th Workshop on Internet and Network Economics (WINE)*. Stanford, California, 87–105.

[9] Robert Bredereck and Edith Elkind. 2017. Manipulating Opinion Diffusion in Social Networks. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, Carles Sierra (Ed.). ijcai.org, 894–900.

[10] Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems* 30, 1 (1998), 107–117. https://doi.org/10.1016/S0169-7552(98)00110-X Proceedings of the Seventh International World Wide Web Conference.

[11] Geoff Davis, Stephane Mallat, and Marco Avellaneda. 1997. Adaptive greedy approximations. *Constructive approximation* 13, 1 (1997), 57–98.

[12] Piotr Faliszewski, Rica Gonen, Martin Koutecký, and Nimrod Talmon. 2018. Opinion Diffusion and Campaigning on Society Graphs. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 219–225. https://doi.org/10.24963/ijcai.2018/30

[13] Michal Feldman, Nicole Immorlica, Brendan Lucier, and S. Matthew Weinberg. 2014. Reaching Consensus via non-Bayesian Asynchronous Learning in Social Networks. In *Proceedings of the 17th. International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*.

[14] Umberto Grandi, James Stewart, and Paolo Turrini. 2020. Personalised rating. *Auton. Agents Multi Agent Syst.* 34, 2 (2020), 55.

[15] Matthew O. Jackson, Suraj Malladi, and David McAdams. 2020. Learning through the Grapevine: The Impact of Message Mutation, Transmission Failure, and Deliberate Bias. In *EC*. ACM, 645.

[16] N.F. Johnson, N. Velásquez, N.J. Restrepo, and et al. 2020. The online competition between pro- and anti-vaccination views. *Nature* 582 (2020), 230–233.

[17] Tatsuya Kameda, Yohsuke Ohtsubo, and Masanori Takezawa. 1997. Centrality in sociocognitive networks and social influence: An illustration in a group decision-making context. *Journal of Personality and Social Psychology* 73, 2 (1997), 296–309.

[18] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. 137–146.

[19] Jure Leskovec and Julian Mcauley. 2012. Learning to discover social circles in ego networks. *Advances in neural information processing systems* 25 (2012).

[20] Dunia López-Pintado and Duncan J. Watts. 2008. Social Influence, Binary Decisions and Collective Dynamics. *Rationality and Society* 20, 4 (November 2008), 399–443.

[21] Naoki Masuda, Mason A. Porter, and Renaud Lambiotte. 2017. Random walks and diffusion on networks. *Physics Reports* 716-717 (2017), 1–58. https://doi.org/10.1016/j.physrep.2017.07.007 Random walks and diffusion on networks.

[22] Sachin Modgil, Rohit Kumar Singh, Shivam Gupta, and Denis Dennehy. 2021. A confirmation bias view on social media induced polarisation during Covid-19. *Inf. Syst. Front.* (Nov. 2021), 1–25.

[23] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. 1978. An analysis of approximations for maximizing submodular set functions—I. *Mathematical programming* 14, 1 (1978), 265–294.

[24] Chao Qian, Jing-Cheng Shi, Yang Yu, Ke Tang, and Zhi-Hua Zhou. 2017. Subset selection under noise. *Advances in neural information processing systems* 30 (2017).

[25] Chengcheng Shao, Giovanni Luca Ciampaglia, Onur Varol, Kai-Cheng Yang, Alessandro Flammini, and Filippo Menczer. 2018. The spread of low-credibility content by social bots. *Nature Communications* 9, 1 (2018), 4787.

[26] Alexander J. Stewart, Mohsen Mosleh, Marina Diakonova, Antonio A. Arechar, David G. Rand, and Joshua B. Plotkin. 2019. Information gerrymandering and undemocratic decisions. *Nature* 573, 7772 (2019), 117–121.

[27] Faezeh Taghipour, Hasan Ashrafi-rizi, and Mohammad Reza Soleymani. 2021. Dissemination and Acceptance of COVID-19 Misinformation in Iran: A Qualitative Study. *International Quarterly of Community Health Education* 0, 0 (2021), 0272684X211022155. https://doi.org/10.1177/0272684X211022155 arXiv:https://doi.org/10.1177/0272684X211022155 PMID: 34098804.

[28] Jie Tangand, Jimeng Sun, Chi Wang, and Zi Yang. 2009. Social influence analysis in large-scale networks. In *International conference on Knowledge discovery and data mining (KDD)*. Paris, France, 807–816.

[29] Duncan J. Watts. 2002. A Simple Model of Global Cascades on Random Networks. *Proceedings of the National Academy of Sciences of the United States of America* 99, 9 (April 2002), 5766–5771.

[30] H. Peyton Young. 2009. Innovation Diffusion in Heterogeneous Populations: Contagion, Social Influence, and Social Learning. *American Economic Review* 99, 5 (2009), 1899–1924.