Epidemiological Model Independent Misinformation Source Identification

Kaustav Basu, Arunabha Sen
School of Computing, Informatics and Decision Systems Engineering, Arizona State University
{kaustav.basu, asen}@asu.edu

Abstract

With increased connectivity, ease of use and free access, social networks have become the go-to platform for information interchange. Recently, however, a surge in misinformation dissemination has been witnessed on these platforms. Works exist which assume a particular standardized epidemiological model (SI, SIR, SIRS, etc.) to determine the sources of misinformation dissemination. However, this assumption becomes impractical in real world settings and, little or no works are present which determine the sources of misinformation *without relying heavily on such underlying epidemiological models*. In this paper, we attempt to fill in this gap by presenting a resource optimized strategy of deploying a minimum number of "detection sensors" on a social network, in order to uniquely identify a user, if they were to disseminate misinformation. We show that by monitoring the social media content of a small subset of users, the platform can still uniquely identify a user, if they were to engage in misinformation dissemination. We utilize the mathematical notion of Identifying Codes to solve our problem. As the computation of the optimal solution is NP-Complete, we provide a polynomial time approximation algorithm and two minimal algorithms. Finally, we highlight the significant resource reduction and scalability achieved by our approaches, by utilizing various real world anonymous Facebook datasets.

Introduction

Researchers have studied the dissemination of misinformation on social networks and attempted to identify its source(s) for the better part of the last decade. Numerous studies assumed that misinformation disseminated in a manner similar to that of infectious diseases and, as a result, utilized models such as SI, SIR, SEIS, etc. in order to place sensors and identify the source of the dissemination (Shah and Zaman 2011; Zhu, Chen, and Ying 2016; Zhou et al. 2019). However, these approaches suffer from certain drawbacks. Firstly, these works assume the existence of an underlying spreading model. In the real world, each misinformation may disseminate differently and the selection of an appropriate model may become difficult, thereby hindering the task of successfully identifying the source of the dissemination (Wang et al. 2017; Dong et al. 2019). Secondly, even if we assume the existence of an underlying spreading model, these works further assume that at the end of the spreading process, all the nodes in the network have the capability to declare if they have been infected with misinformation or not (Spinelli, Celis, and Thiran 2017). This is only possible if the nodes themselves are fact checkers, which is not the case on social networks. Finally, even if one may correctly assume a dissemination model, it is hard to estimate the values of the parameters of the model (Wang et al. 2017; Dong et al. 2019). Thus, there is clearly a need for the development of misinformation source detection algorithms which do not rely heavily on an underlying dissemination model.

In this paper, we attempt to fill in this gap by presenting an approach based solely on the topological structure of a social network. We assume that we have complete knowledge of the social network under study and our graph theoretical framework utilizes this in order to place the "sensors" [1] on a *minimum subset of the nodes in the graph*, to accurately identify the users engaging in misinformation dissemination. We show that by following our approach, there is no need to actively monitor each and every user, but only a *subset* of the total users set, which could be as low as $O(\log n)$. *In other words, unlike the epidemiological models, only a small subset of the users have to be fact-checked, as opposed to everyone in the network.* Moreover, we show that by monitoring the users (nodes) in this subset, our approach can still *uniquely identify any user in the social network, who engages in misinformation dissemination*. Our resource optimized approach exploits a common characteristic of social networking platforms: social network posts/comments of an individual are "visible" to his/her immediate friends (assuming that the post does not have custom visibility settings). We show that our approach, based on the mathematical concept of *Identifying Codes*, reduces the resource requirements (fewer number of users to be monitored and as a direct consequence, lesser computational time) significantly for the platforms, with the help of extensive experimentation on anonymous real world Facebook datasets. To that end, the main contributions of this paper are as follows:

- Our framework is an alternative to the existing epidemiological models, in the sense that it does not assume the

[1] We would like to point out here that, in this paper, we focus primarily on the placement of these detection sensors and not on their development.

existence of models such as SI, SIR, SEIR, etc.

- We transform our problem to a variant of the Hitting Set problem in a novel manner and present an ILP, an approximation algorithm and scalable two minimal algorithms.
- Our framework is universal. By simply changing the type of the detection sensor (from misinformation detectors to hate speech detectors, for instance), platforms can effectively identify and take action against users disseminating objectionable content on social networks.

Related Work

In this section, we highlight the motivating studies behind our work in three broad areas - development of the detection sensors, epidemiological model based misinformation source identification and the unique identification capabilities of Identifying Codes.

Detection Sensors

Preliminary research on misinformation detection was primarily unimodal (either textual or visual). Textual approaches, by (Potthast et al. 2017), considered features such as headlines, lexical, syntactic, semantic, writing style etc. of social media posts to determine if the content was information or misinformation. (Gupta et al. 2013) studied visual approaches and tried to identify certain features which can be utilized for the classification of images as information or misinformation. Recently, (Jin et al. 2017; Wang et al. 2018; Khattar et al. 2019; Qi et al. 2019) utilized deep neural networks to classify multi-modal social media posts (textual + visual). Further studies focused on the social context aspect to determine the authenticity of social media posts. (Shu, Wang, and Liu 2018) analyzed user profiles, (Yang et al. 2019) analyzed user opinions in an unsupervised manner and (Jin et al. 2016) studied the user opinions towards social media posts to determine the veracity of the post. The problem of fake news mitigation was mapped to the reinforcement learning framework, with the goal of optimizing the actions for maximal total reward under budget constraints by (Farajtabar et al. 2017). (Shi and Weninger 2016), viewed link-prediction task in a knowledge graph to accurately determine the veracity of a fact. (Shu et al. 2017) presented a survey of detecting fake news on social media. Real-world datasets measuring users'trust level on fake news was constructed by (Shu, Wang, and Liu 2018). (Tacchini et al. 2017) showed that Facebook posts can be classified with high accuracy as hoaxes or non-hoaxes on the basis of the users who "liked" them.

Epidemiological Models for Source Detection

The seminal work of (Shah and Zaman 2011) paved the foundation for the rumor source detection problem. In their work, they assumed that rumors spread according to the SI model on social networks and that they had complete information about all the states of the nodes (including network parameters) in the network. (Zhu and Ying 2014) also assumed that they had complete information about all the node states but assumed that rumors spread according to the SIR model for detecting single rumor sources. In their follow up work, (Zhu, Chen, and Ying 2016) swapped their assumption of the complete network snapshot with that of the partial network snapshot and present two algorithms to detect *multiple* diffusion sources. (Spinelli, Celis, and Thiran 2017) presented static and dynamic sensor placement approaches for rumor source detection, wherein the rumor disseminates following the SI model. (Paluch et al. 2020) go one step further and assume that each node has the capability of reporting which neighboring node sent the virus (misinformation). (Tang 2020) argued that it is difficult to know the topology of the network in advance and utilized network topology inference and epidemiological models for the detection of the sources of misinformation. (Racz and Richey 2020) studied the problem of robust rumor source identification problem in the face of adversaries, who can perturb the original misinformation in order to shield the source. Even though there exists numerous works on epidemiological model based rumor source(s) identification, authors in (Dong et al. 2019; Wang et al. 2017) criticized the epidemiological model assumptions and stated that knowing the underlying model beforehand is infeasible. (Wang et al. 2017) presented an approach to propagate (without knowing the underlying model) the infection label throughout the network and use peaks to identify the source nodes. (Dong et al. 2019) builds on (Wang et al. 2017) and utilizes Graph Convolutional Neural Networks to identify sources based on non-integral node infection labels.

Identifying Codes

Sensor placement optimization for the unique identification of the nodes in a graph was first introduced as Identifying Codes by (Karpovsky, Chakrabarty, and Levitin 1998) and provided results for a class of graphs. (Charon, Hudry, and Lobstein 2003) proved the NP Completeness for the minimum Identifying Code problem based on a reduction from the 3-SAT problem. (Ray et al. 2003) introduced the concept of robust Identifying Codes to deal with faults in sensor networks. The Identifying Code problem was approximated to an approximation factor of $O(\log n)$ in (Gravier, Klasing, and Moncel 2008; Suomela 2007) by utilizing the notions of entropy and disjoint unions. It was shown in (Moncel 2006) that for a particular class of graphs, the cardinality of the Identifying Code solution could be as low as $\log_2\ n + 1$. Integer Linear Programs to compute the Minimum Identifying Code Set (MICS) for a given graph was presented in (Basu et al. 2018a,b; Basu and Sen 2019a,b, 2021; Basu et al. 2019) for monitoring networks obtained from numerous domains such as terrorism, drug and critical infrastructures. A lower bound on the MICS for a k-fault tolerant system was presented in (Sen et al. 2018). It was shown in (Basu 2019) how the same ILPs can be utilized for the computation of the minimum number of users in order to uniquely identify users engaging in misinformation dissemination.

In this work, we show that the *ILP based approaches fail in case of social networks and accordingly present approximation and scalable minimal algorithms*. Additionally, our work is different from the motivating works of (Wang et al. 2017; Dong et al. 2019) in the sense that both works take

as input a set of nodes who have already been infected and make a reasonable assumption that the nodes surrounded by infected nodes are more likely to be the source nodes. In our paper, however, we do not need to keep track of individual labels of each node in the network and can simply identify the sources of misinformation *by triangulating the classification outputs of the detection sensors placed in the network*. Finally and more importantly, our approach *does not take into account any prior information regarding the nodes which have already been infected*.

Preliminaries

In this section, we define the mathematical concept of Identifying Codes as defined in (Basu 2019).

Definition 1. *Given an undirected graph $G = (V, E)$, the subset $V' \subseteq V$, is defined as an Identifying Code Set (ICS) for the vertex set V, if $\forall v \in V, N^+[v] \cap V'$ is unique, where, $N^+[v] = v \cup N(v)$ and $N(v)$ represents the set of nodes adjacent to v in $G = (V, E)$. The Minimum Identifying Code Set (MICS) problem is to find the ICS of smallest cardinality.*

The vertices in the set V' (MICS) can be thought of as *alphabets* of the (identifying) code, and $N^+[v] \cap V'$ is the unique code/signature identifying node v. This is better explained with the help of the following example. Consider the graph $G = (V, E)$, as illustrated in Fig. 1. The MICS of G is $V' = \{u_4, u_6, u_7, u_8\}$. As shown in Table 1, in $G = (V, E)$, $\forall u \in V, N^+[u] \cap V'$ is unique. In other words, all the nodes in the graph receive a unique signature if sensors were to be deployed on nodes in V'. Hence, the node set $V' = \{u_4, u_6, u_7, u_8\}$ is an MICS of G. It might be argued that one could have used a set cover (or vertex cover, by extension) approach to have monitored the network. For the graph in Fig. 1, an optimal set cover placement is $\{u_3, u_7, u_8\}$. For such a placement, if node u_5 were to behave anomalously (post misinformation), it can be seen from Fig. 1, that sensor u_8 ($\in N[u_5]$) would be triggered. But, the triggering of u_8 could have also implied that the node u_8 was behaving anomalously. Therefore, following the optimal set cover placement approach, we could not have distinguished between nodes u_5 and u_8. This drawback is overcome by our Identifying Code approach, as is evident from Table 1.

Definition 2. *Two nodes $u, v \in V$ are said to be "twins" if $N^+[u] = N^+[v]$ in an undirected graph.*

Observation: Identifying Code Set (ICS) of a graph $G = (V, E)$ does not exist, if any two nodes $u, v \in V$ are "twins". In other words, *the necessary and sufficient condition* for an undirected graph to have an Identifying Code is that the graph be "twin-free".

Graph Construction Rules: In our work, a graph representing a social network can be constructed as follows: we can denote each registered user on the social networking platform by a node. If a user u is friends with another user v and posts shared by u shows up on v's timeline and vice versa, then there is an edge between the two. If, for some reason, u and v are friends but u's posts do not show up on v's timeline due to u's custom visibility settings, then, in our work, we do not consider an edge to exist between them.

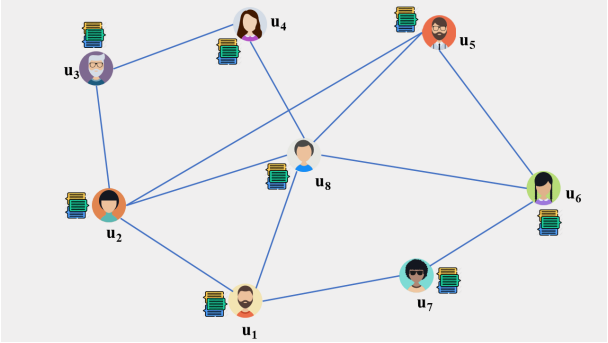


Figure 1: Timeline View of Facebook

$N^+[u_1] \cap V' = \{u_7, u8\}$	$N^+[u_2] \cap V' = \{u_8\}$
$N^+[u_3] \cap V' = \{u_4\}$	$N^+[u_4] \cap V' = \{u_4, u_8\}$
$N^+[u_5] \cap V' = \{u_6, u_8\}$	$N^+[u_6] \cap V' = \{u_6, u_7, u_8\}$
$N^+[u_7] \cap V' = \{u_6, u_7\}$	$N^+[u_8] \cap V' = \{u_4, u_6, u_8\}$

Table 1: $N^+[u] \cap V'$ for Fig. 1

Problem Formulation

In this section, we formalize our problem of finding the minimum number of users, on whom detection sensors must be placed to ensure unique identification of all the users in the network, in case the user represented by v, engages in misinformation dissemination. In this effort, we assume that only *one* node v becomes active at a time step and each and every node $v \in V$ can be monitored.

As evident, our approach does not assume the existence of an underlying epidemiological model for the dissemination of misinformation. We utilize the connections between users on the social network in order to determine the source of misinformation dissemination. Our approach relies on the social network property that posts made by a user's connections will be visible (or available) on the timeline of the user in question. Moreover, we do not assume that the nodes in our network have the capability to reveal their state (either infected with misinformation or not) after misinformation dissemination, as is evidenced by prior approaches. As mentioned previously, this assumption limits the applicability of prior approaches. In our work, we place detection sensors, which *do have the capability to distinguish between real and fake content*, on a small subset of the users in the network. One critical aspect of misinformation detection on social networks is to monitor *all the unique posts* generated by users on the platform. Thus far we have discussed the placement of a minimum number of sensors in order to uniquely monitor all the nodes in the graph. We have to guarantee that by monitoring the content of this subset of the nodes ensures that we monitor all the unique posts generated on the social networking platform. It is trivial to note that, if we place detection sensors on *all* the nodes of the graph, then all the posts generated on the platform will be definitely monitored. We now argue that even with the monitoring of a small subset of users (provided there are no custom post settings which prevent the post from propagating to all the friends), we still retain the capability of monitoring each and every unique post generated on the platform.

Theorem 1. *Given a graph $G = (V, E)$ corresponding to a social network, in which V' is an MICS, by placing detection sensors on every node $v' \in V'$, our approach monitors all unique posts generated by users on the platform for misinformative content.*

Proof. Assume that the detection sensors have been placed on a subset V' of a graph $G = (V, E)$, where V' is an MICS of the graph. It is trivial to note that these sensors can monitor the content generated by the users they have been placed on. By the definition of ICS, at least one detection sensor has been placed in the neighborhood of each and every $v \in V - V'$. If this wasn't the case then we would have nodes in the graph which would not have been monitored. Therefore, if any user u, were to post any content on their timeline, then the post would appear on the timelines of a set of users $\mathcal{V}$ on which the detection sensors have been placed, where $\mathcal{V} \subseteq N^+[u]$ and $\mathcal{V} \subset V'$. Thus we can guarantee that all the *unique* posts on the platform are monitored.

□

Hitting Set (HS) Formulation: We now provide a novel transformation where, we view the MICS problem as a Minimum Hitting Set (MHS) problem. This transformation allows us to utilize the well known greedy algorithm of the MHS problem in order to provide the approximation bound for the MICS problem. Interestingly, previous research efforts which determined the approximation bounds for the MICS problem explored convoluted routes, such as determining entropy, disjoint sets, etc. (Xiao, Hadjicostis, and Thulasiraman 2006; Gravier, Klasing, and Moncel 2008). It may be noted that the greedy heuristic for the HS problem provides a $O(log\ m)$ factor performance bound, where m is the number of elements in the collection set (Vazirani 2013). In the following, we define the minimum HS problem:

Definition 3. *Given a universal set $\mathcal{U} = \{u_1, ..., u_n\}$, and a collection set $\mathcal{S} = \{\mathcal{S}_1, ..., \mathcal{S}_m\}$, where $\mathcal{S}_i \in \mathcal{U}$, find the smallest subset $U' \subseteq \mathcal{U}$, which hits every set $\mathcal{S}_i \in \mathcal{S}$.*

Definition 4. *Closed Neighborhood of $v_i = CN(v_i) = N^+(v_i)$, where $N^+(v_i) = N(v_i) \cup \{v_i\}$, where $N(v_i)$ denotes the neighborhood of the node v_i.*

Definition 5. *Distinguishing Set for v_i and v_j = $DS(v_i, v_j) = CN(v_i) \bigoplus CN(v_j)$. $\bigoplus$ denotes the symmetric difference operation between the closed neighborhood sets $CN(v_i)$ and $CNv(j)$. In other words, picking at least one element from the set $DS(v_i, v_j)$ will distinguish between nodes v_i and v_j.*

Definition 6. *Universal Set $\mathcal{U} = \{v_1, ..., v_n\}$, where each element v_i is a node in the social network graph and Collection Set $\mathcal{S} = \cup_{i=1}^{n}[CN(v_i)\ \cup_{j=1}^{n} \{DS(v_i, v_j)\}]$.*

Our objective is to select the minimum number of elements from $\mathcal{U}$, such that all the elements in $\mathcal{S}$ are hit. Hitting all the elements in $\mathcal{S}$ ensures that, (i) all $CN(v_i)$ sets are hit, which in turn ensures that all nodes in the graph are monitored (a detection sensor has been placed in the closed neighborhood of v_i), and (ii) all $DS(v_i, v_j)$ sets are hit, which in turn ensures that all the nodes in the graph are *uniquely monitored*. Thus the computation of this variant of the minimum HS problem is equivalent to solving the MICS problem.

Problem Solution

Here, we provide (i) optimal solution for the MICS problem utilizing an Integer Linear Program, based on the HS approach, (ii) heuristic solutions for the MICS problem, by relaxing the integrality constraints of the HS ILP, (iii) an approximation algorithm for the MICS problem with guaranteed performance bound, by utilizing the greedy HS approximation algorithm, and (iv) two minimal algorithms.

Optimal Solution

Instance: A universal set $\mathcal{U} = \{u_1, ..., u_n\}$ and a collection set $\mathcal{S} = \{\mathcal{S}_1, ..., \mathcal{S}_m\}$, where $\forall i, \mathcal{S}_i \subset \mathcal{U}$.
Problem: Find the smallest subset $\mathcal{U}' \subseteq \mathcal{U}$, which intersects or hits every set $\mathcal{S}_i \in \mathcal{S}$, where $\mathcal{S} = \{\mathcal{S}_1, ..., \mathcal{S}_m\}$.

Corresponding to each $u_i \in \mathcal{U}$, we use a variable x_i,

$$x_i = \begin{cases} 1, & \text{if } x_i \text{ is included in } \mathcal{U}', \\ 0, & \text{otherwise} \end{cases}$$

Objective Function: Minimize $\sum_{u_i \in \mathcal{U}} x_i$

Hitting Constraint: $\sum_{u_i \in \mathcal{S}_i} x_i \geq 1, \forall \mathcal{S}_i \in \mathcal{S}$

The objective function ensures that a minimum number of elements are selected from $\mathcal{U}$. The Hitting Constraint ensures that all the sets in $\mathcal{S}$, are hit. We design two heuristics from the LP relaxations from the above ILP. Heuristic 1 (or $LP1_{HS}$) is the relaxed solution where we select the highest fractional values in the LP solution, in descending order, till the graph is uniquely monitored. Heuristic 2 (or $LP2_{HS}$) is the relaxed solution where we select indicator variables independently at random, via randomized rounding, till the graph is uniquely monitored. $LP2$ has a guaranteed error bound of (1 - 1/e) $\sim$ 63% of the optimal (Vazirani 2013).

Approximate Solution

We can now utilize the well known greedy approximation algorithm for HS as an approximation algorithm for the MICS problem (Vazirani 2013). The performance bound for our HS approximation algorithm is $O(log\ n^2)$, since we have a quadratic number of sets in the collection set as a function of the number of elements in the universe. That being said, however, $O(log\ n^2) = O(log\ n)$. Thus, our HS based approach has a performance bound of $O(log\ n)$ and we can claim that the MICS problem also has an $O(log\ n)$ approximation bound. Prior works have already established the $O(log\ n)$ bound for the MICS approximation, utilizing minimizing entropy and computation of disjoint sets (Gravier, Klasing, and Moncel 2008). However, we believe that our transformation is much simpler and easier to implement.

Minimal Solution

In order to make our approach scalable compared to the ILP and approximation algorithm, we present two minimal Identifying Code approaches. We hypothesize that sacrificing a little on the optimality will lead to greater benefits on the computational side. Our minimal algorithms takes as input the graph and a graph centrality based node sequence. Here, we consider two types of node sequences - (i) nodes

Data: Twin Free Graph, $G = (V, E)$, and a node sequence S
Result: Return the minimal Identifying Code C of G

$C = S$;
Place sensors on all nodes in C;
while $S \neq \emptyset$ **do**
    node $\leftarrow$ Select the first node in S;
    $NewC = C \setminus \{node\}$;
    Place sensors on all nodes in $NewC$;
    if *Each node in G is uniquely identifiable* **then**
        $C = NewC$;
    end
    else
        $C = C$
    end
    Remove node from S;
end
Return C;

Algorithm 1: Minimal Algorithm

arranged in decreasing centrality scores, which we refer to as the MAX approach and, (ii) nodes arranged in increasing centrality scores, which we refer to as the MIN approach. One minimal algorithm takes the graph and MAX node sequence as input and the other takes the graph and MIN node sequence as input. The overall generalized minimal algorithm is presented in Algorithm 1. Further, we have considered four standard graph centrality measures - degree (DC), betweenness (BC), eigenvector (EC) and pagerank (PR) for comparison in the experimental section.

Experimental Results

We implemented our approaches on various real world *undirected social network* datasets obtained from (Leskovec and Krevl 2014; Rossi and Ahmed 2016). For a network to have an Identifying Code, it must be "twin-free" and, one trivial Identifying Code set solution is the set V itself, although, V may not be the Identifying Code set of *minimum/minimal cardinality*. However, our algorithms show that unique identification for all the nodes in the network can be obtained by monitoring a subset $V' \subseteq V$.

Datasets

We utilized various real world, undirected and anonymous Facebook datasets, obtained from (Leskovec and Krevl 2014; Rossi and Ahmed 2016). The instances of the Facebook networks varied from 52 nodes to 32375 nodes, and are denoted as FB1-FB19 in Table 2. However, due to computational limitations of the laptop on which our methods were implemented, we were not able to consider graphs larger than 32000 nodes. Observe that, almost all of the networks initially contained "twins" in Table 2. For instance, FB1 had 52 nodes initially and after a simple "twin" removal procedure, where "twins" were condensed into a single node (super node), was left with 46 "twin" free nodes. Since "twins" were condensed into a single node, if the condensed node were to become active in misinformation dissemination, then additional lower level analysis would be required to distinguish between the "twins". Table 2 presents the various datasets considered for our experimentation and corresponding network statistics with results.

Analyses

Our objective was two-fold - (i) to show that Integer Linear Programs cannot be utilized for unique coverage (unique monitoring) apart from fairly small problem instances, and (ii) the minimal algorithm, which we presented in this paper, not only scales well, but also provides similar quality solutions as that of the ILPs. All the experiments were executed on a 5th generation Intel Core i-5 processor with 2.30 GHz and 64GB RAM. We present the results of our analyses in Table 2 and as plots, illustrated in Figs. 2(a) - 5(b).

We present the number of sensors required to uniquely monitor the corresponding social networks in Table 2[2]. The first column in the table indicates the unique ID of the network. The following two columns report the original number of nodes and edges in the social network (graph), before "twin" removal. The subsequent two columns records the number of nodes and edges post "twin" removal. The next columns indicate the minimum number of sensors required following the various approaches outlined previously. Note that the ILPs and its linear relaxations did not finish computing the number of sensors required for networks FB11 - FB19, within 12 hours of CPU clock time. The HS Approximation algorithm also, could not finish the computation of the number of sensors required for networks FB12 - FB19, within 16 hours. Finally, the MAX BC and MIN BC approaches could not finish the computations for FB17-FB19 and FB16-FB19 respectively, within 16 hours as the worst case computation of the betweenness centrality is $\sim O(n^2)$ time (Brandes 2001), where n denotes the number of nodes. The computation times of 12 and 16 hours were determined based on the computation times of the other approaches on the corresponding graphs. All of the unfinished computations are denoted by $-$ in Table 2.

In our experimentation, we use two notions extensively, *quality* and *cost*. We define *quality* as the "goodness" of a solution, which can be mathematically represented as,

$$Q = (\#\, Nodes - Solution\, Cardinality)/\#\, Nodes \quad (1)$$

The value of Q lies in the interval $[0, 1]$, where 1 represents a high quality solution, or in other words, a solution with high reduction in resources. Lower the value of the solution cardinality, higher the value of Q and higher is the quality of an approach. Next, we define *cost* as the computational time spent in order to attain the respective quality. Fig. 2(a)[3] illustrates the quality of the ILP, LPs and approximation approaches outlined previously, for FB1-FB10. This is because, FB10 is the largest instance which we could provide as input to the ILP. Hence, for initial comparison, we plot the quality and cost results of the ILPs, LPs and HS approximation as obtained from FB1-FB10. As expected, the quality of the ILP is the best. In this figure, we can see that our *Hitting Set (HS) approximation approach produces near*

[2]Best Viewed In Color

[3]Images Best Viewed in Color

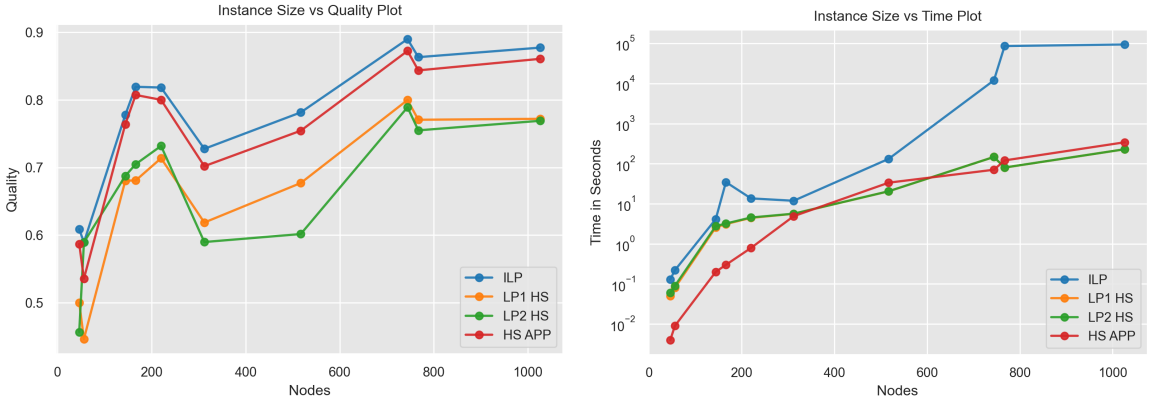
Network ID	# Nodes	# Edges	# Nodes Post Twin Removal	# Edges Post Twin Removal	ILP HS	LP1 HS	LP2 HS	HS APP	MAX DC	MAX BC	MAX EC	MAX PR	MIN DC	MIN BC	MIN EC	MIN PR
FB1	52	198	46	181	18	23	25	19	23	25	23	24	18	18	18	18
FB2	61	331	56	299	18	31	23	26	25	26	25	26	24	24	24	24
FB3	150	1843	144	1731	32	46	45	34	46	47	45	45	38	36	38	38
FB4	168	1824	166	1817	30	53	49	32	60	57	52	57	32	32	33	32
FB5	224	3416	220	3393	40	63	59	44	68	67	62	69	46	44	46	47
FB6	333	2852	312	2730	85	119	128	93	125	124	120	128	88	89	90	88
FB7	534	5347	517	5203	113	167	206	127	174	179	169	177	126	130	134	125
FB8	747	30772	744	30756	82	149	157	95	139	140	125	146	114	102	113	112
FB9	786	14810	767	14702	105	176	188	120	196	204	190	204	112	123	127	120
FB10	1034	27783	1026	27742	126	234	237	143	238	232	217	241	153	148	160	153
FB11	4039	92273	3951	91577	–	–	–	766	1128	1143	1081	1150	788	787	810	789
FB12	6386	224048	6374	224014	–	–	–	–	1560	1593	1471	1580	764	748	779	752
FB13	8600	393126	8590	393102	–	–	–	–	1837	1863	1729	1888	839	811	853	826
FB14	11247	362605	11245	362601	–	–	–	–	2789	2820	2604	2844	1274	1232	1300	1257
FB15	18448	992366	18448	992366	–	–	–	–	3763	3826	3512	3832	1525	1470	1592	1506
FB16	22900	875319	22894	875295	–	–	–	–	5629	5699	5255	5763	2603	–	2648	2569
FB17	27737	1062539	27730	1062518	–	–	–	–	6566	–	6137	6680	3016	–	3108	2968
FB18	29747	1335512	29738	1335487	–	–	–	–	6448	–	5977	5977	2801	–	2884	2757
FB19	32375	1151149	32361	1151102	–	–	–	–	7659	–	6819	7788	3476	–	3565	3397

Table 2: Minimum (Minimal) Detection Sensors Required. – Indicates That the Algorithm Did Not Finish Computation Within A Specific Time Frame. The Best Performing MAX And MIN Approach Has Been Marked In Blue and Purple Respectively.

optimal solution quality. The qualities of the linear relaxations of the ILP are also presented in the figure, but are not as high as the quality of the approximation algorithm. Fig. 2(b) illustrates the cost associated with the approaches for obtaining the corresponding quality, illustrated in Fig. 2(a). It should be observed that the y-axis in Fig. 2(b) is in the log-scale because - (i) the quality of the smaller networks (FB1-FB2) was computed in less than a second, and (ii) to make the plot more visually more understandable. The cost of the ILP approach tends to grow exponentially with an increase in the problem instance. The growth curves of the other approaches, including the approximation algorithm, are not as extreme as that of the ILP. There are a few takeaways from these two plots, (i) the cost of the approximation algorithm, while producing near optimal solution, also tends to grow exponentially with an increase in the size of the problem instance (a linear growth in log-scale implies exponential growth in normal scale), (ii) it is due to such growth in the cost that, we could only implement these approaches on small graphs/networks, more specifically FB1-FB10.

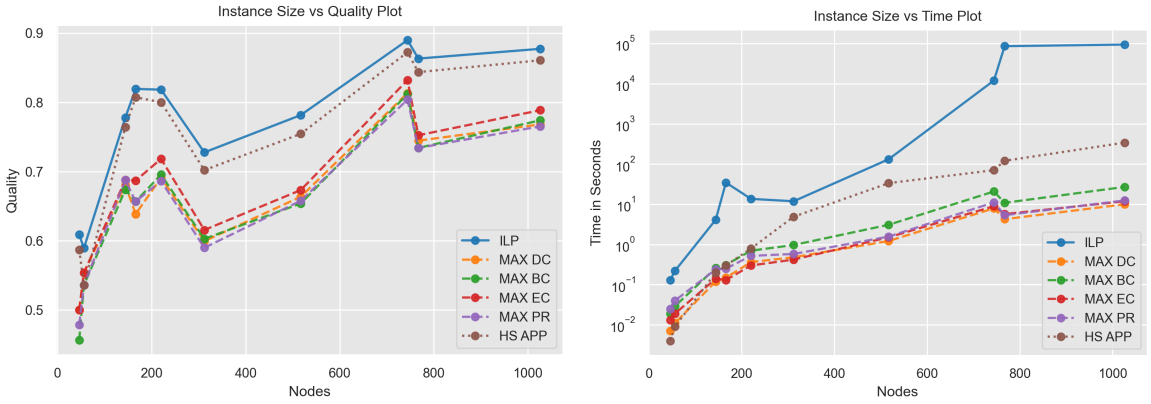
Having established the fact that the optimal approach and our approximation approach do not scale well, we sacrifice on the objective of attaining the minimum (or even approximate) solution and settle for the *minimal solution*, and present the results of our analyses utilizing the minimal algorithm (with its two node sequence orderings). It may be recalled that we considered two groups of node sequence orderings based on common centrality measures, the maximum ordering with MAX DC, MAX BC, MAX EC and MAX PR, and the minimum ordering with MIN DC, MIN BC, MIN EC and MIN PR. *To show the efficacy of our minimal approach, we compare the quality and cost of each variation with the ILPs and approximation algorithm for smaller networks, before moving on to larger networks.* Figs. 3(a) and 3(b) illustrate the comparison of the qualities and costs of MAX DC, MAX BC, MAX EC and MAX PR with the quality and cost of the ILPs and approximation algorithm respectively, for FB1- FB10. Fig. 3(a) illustrates that the maximum ordering attains comparatively high quality (greater than 0.65 on average) with MAX EC providing the best quality. Fig. 3(b) illustrates the fact that our minimal approach, based on MAX ordering, approximately attains at least 10x scalability, when compared to the approximation algorithm, as the sizes of the networks keep increasing. It can also be observed that the MAX BC approach is the most expensive approach, among the four maximum orderings, whereas the MAX DC approach is the least expensive. Overall, in terms of quality, the approaches MAX EC and MAX DC do not vary much, but in terms of cost, the MAX DC is more efficient than the MAX EC approach. Figs. 4(a), 4(b) illustrate the comparison of the qualities and costs of MIN DC, MIN BC, MIN EC and MIN PR with the quality and cost of the ILPs and approximation algorithm respectively, for FB1 - FB10. Fig. 4(a) illustrates that the MIN ordering attains almost the same quality, if not better, when compared to the approximation algorithm (averaging greater than 0.75). Among the minimum orderings, MIN BC provides the best quality, whereas the MIN PR and MIN DC are the most efficient. Overall, the difference in cost between the MIN BC and MIN PR / MIN DC is significantly larger than difference in quality between the two, and hence, one may opt for the more efficient MIN PR/ MIN DC approach.

Now, we compare the performances of the minimal algorithm following the MAX ordering sequence with those following the MIN ordering sequence. Note that, MAX BC and MIN BC failed to execute on larger graphs, and hence, we do not consider these two approaches going forward. We include the HS Approximation algorithm for instances it managed to finish its execution (FB1-FB11). Figs. 5(a) and 5(b) illustrate the quality and cost performances of the minimal and approximation algorithms. In Fig. 5(a), it can be seen that the quality of the minimal algorithms following the MIN ordering sequence is almost identical to that of the HS Approximation algorithm and the quality of the minimal algorithms following the MAX ordering sequence is slightly lower. In Fig. 5(b), we see that both the minimal approaches



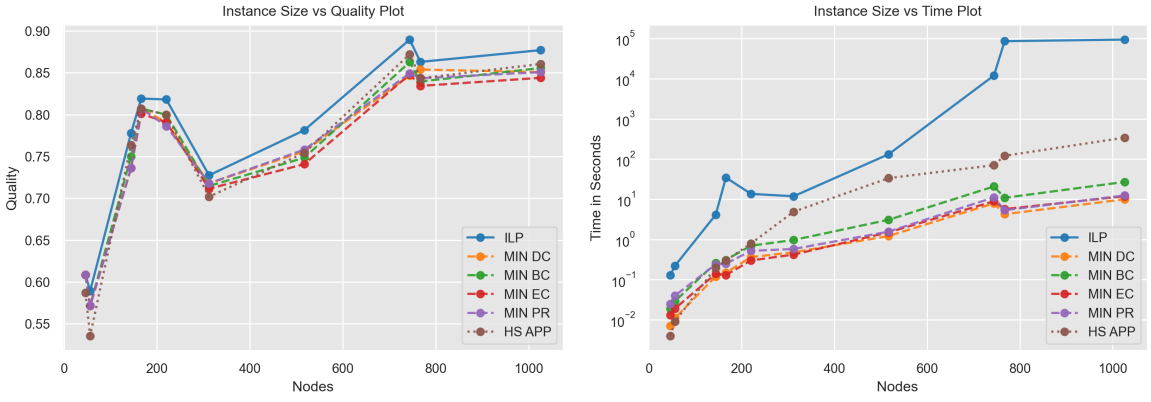
(a) Quality Analysis (b) Cost Analysis

Figure 2: Visual Analysis of The ILP, LPs and Approximation Performances For FB1-FB10



(a) Quality Analysis (b) Cost Analysis

Figure 3: Visual Analysis of The ILP, Approximation and Minimal Algorithm (MAX) Performances For FB1-FB10

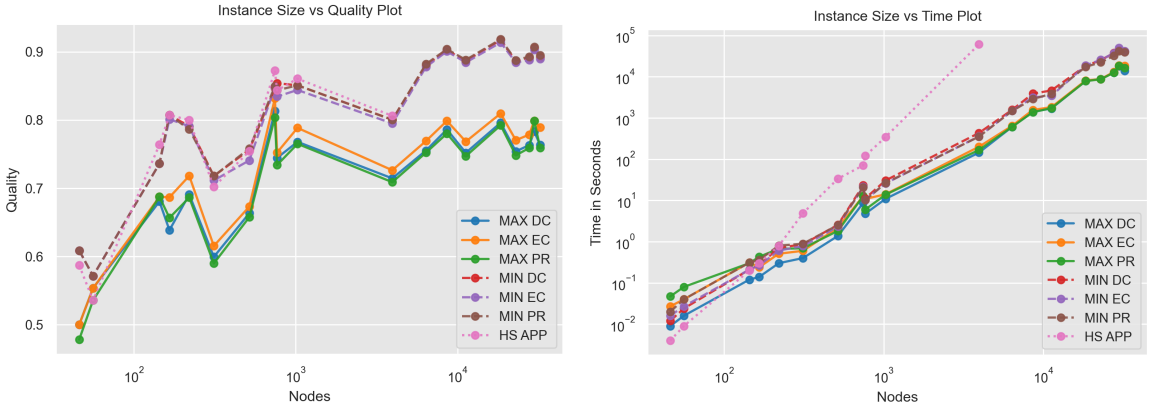


(a) Quality Analysis (b) Cost Analysis

Figure 4: Visual Analysis of The ILP, Approximation and Minimal Algorithm (MIN) Performances For FB1-FB10

grow at a smaller rate as compared to the HS Approximation algorithm. In fact, both the minimal approaches are at least 10x faster than the approximation algorithm. As the size of the instances increases, this gap widens to almost 100x. The minimal algorithms following the MAX ordering are computationally more efficient than those of the MIN ordering.

It can be observed that the quality of the minimal approach following the MIN ordering is as good as the quality of the approximation algorithm, if not better, while bearing only a fraction of the cost. Moreover, the quality achieved by the minimal algorithm following MIN ordering is far superior to that achieved by MAX ordering as, in the case of MAX ordering, nodes removed from consideration (i.e., step 5 of Algorithm 1) are the nodes with higher centrality scores.



(a) Quality Analysis

(b) Cost Analysis

Figure 5: Visual Analysis of The Approximation and The Two Minimal Approaches For FB1-FB19

If such central or important nodes are not considered for sensor placement, then the *effect* of reach of that highly central node is lost, in the sense that, placing a monitor at a more central node would monitor *more* nodes than that of placing a monitor at a lesser central node. Thus, removing highly central nodes from consideration at every iteration would result in the selection of additional nodes to make up for this loss, resulting in the higher solution cardinality. This is not the case for the MIN ordering, where nodes with low scores are removed from consideration first, thus resulting in a smaller loss in effect as opposed to the MAX ordering based approach. Thus, the cost associated with the MIN ordering is higher as a larger set of nodes are being removed from consideration when compared to the MAX ordering.

Following our experimentation, it can be seen that MIN DC, MIN EC and MIN PR all provide near identical quality and cost. MAX EC provides the best quality and has equivalent cost to the other MAX approaches. A key point which we want to make here is that, all of these experiments were conducted on a laptop. We believe that with better computing resources, our MIN and MAX ordering based minimal algorithms could be scaled even further.

Conclusion

In this paper, we have presented a graph theoretic framework for monitoring the minimum number of users in a social network, to uniquely identify any user engaged in misinformation dissemination. Our framework does not assume any underlying epidemiological model for dissemination and does not know in advance, the set of users (nodes) who have been already infected with misinformation. Another major advantage of our framework is the universality - by changing the objective of the detection sensor, one can monitor any objectionable content. An approximation algorithm with guaranteed performance bound was utilized, along with a minimal algorithm with varying input node sequences. We compared our approaches with ILPs and showed that the ILPs as well as their linear relaxations and the approximation algorithm, do not scale particularly well. However, we showed that our minimal algorithms not only scaled well, but also provided solution quality almost as good as the approximation algorithm, if not better, while only being fractionally costly. It may be argued that our approach needs to be implemented every time the graph changes. This is not necessarily true as one might incorporate a simple augmenting approach to handle cases where new users join the platform. Since the locations of the sensor deployment are known, we can easily check if the new node(s) will be uniquely monitored and accordingly, we can monitor the appropriate node(s). Since the number of new nodes joining the platform will be significantly lesser than those who are already present, this augmenting task will not be computationally expensive. Finally, it may be noted that our approach is not robust (unique identification is lost due to sensor failures or adversarial attacks), and it is one aspect that we are currently investigating.

References

Basu, K. 2019. Identification of the Source (s) of Misinformation Propagation Utilizing Identifying Codes. In *Companion Proceedings of The 2019 World Wide Web Conference*, 7–11.

Basu, K.; Dey, S.; Nandy, S.; and Sen, A. 2019. Sensor Networks for Structural Health Monitoring of Critical Infrastructures Using Identifying Codes. In *2019 15th International Conference on the Design of Reliable Communication Networks (DRCN)*, 43–50. IEEE.

Basu, K.; Padhee, M.; Roy, S.; Pal, A.; Sen, A.; Rhodes, M.; and Keel, B. 2018a. Health Monitoring of Critical Power System Equipments Using Identifying Codes. In *International Conference on Critical Information Infrastructures Security*, 29–41. Springer.

Basu, K.; and Sen, A. 2019a. Monitoring individuals in drug trafficking organizations: a social network analysis. In *2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 480–483. IEEE.

Basu, K.; and Sen, A. 2019b. On augmented identifying codes for monitoring drug trafficking organizations. In *2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 1136–1139. IEEE.

Basu, K.; and Sen, A. 2021. Identifying individuals associated with organized criminal networks: a social network analysis. *Social Networks* 64: 42–54.

Basu, K.; Zhou, C.; Sen, A.; and Goliber, V. H. 2018b. A Novel

Graph Analytic Approach to Monitor Terrorist Networks. In *2018 IEEE SocialCom*, 1159–1166. IEEE.

Brandes, U. 2001. A faster algorithm for betweenness centrality. *Journal of mathematical sociology* 25(2): 163–177.

Charon, I.; Hudry, O.; and Lobstein, A. 2003. Minimizing the size of an identifying or locating-dominating code in a graph is NP-hard. *Theoretical Computer Science* 290(3): 2109–2120.

Dong, M.; Zheng, B.; Quoc Viet Hung, N.; Su, H.; and Li, G. 2019. Multiple Rumor Source Detection with Graph Convolutional Networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 569–578.

Farajtabar, M.; Yang, J.; Ye, X.; Xu, H.; Trivedi, R.; Khalil, E.; Li, S.; Song, L.; and Zha, H. 2017. Fake news mitigation via point process based intervention. *arXiv preprint arXiv:1703.07823* .

Gravier, S.; Klasing, R.; and Moncel, J. 2008. Hardness results and approximation algorithms for identifying codes and locating-dominating codes in graphs. *Algorithmic Op. Research* 3(1).

Gupta, A.; Lamba, H.; Kumaraguru, P.; and Joshi, A. 2013. Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy. In *Proceedings of the 22nd international conference on World Wide Web*, 729–736. ACM.

Jin, Z.; Cao, J.; Guo, H.; Zhang, Y.; and Luo, J. 2017. Multimodal fusion with recurrent neural networks for rumor detection on microblogs. In *Proceedings of the 25th ACM international conference on Multimedia*, 795–816. ACM.

Jin, Z.; Cao, J.; Zhang, Y.; and Luo, J. 2016. News verification by exploiting conflicting social viewpoints in microblogs. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Karpovsky, M. G.; Chakrabarty, K.; and Levitin, L. B. 1998. On a new class of codes for identifying vertices in graphs. *IEEE Transactions on Information Theory* 44(2): 599–611.

Khattar, D.; Goud, J. S.; Gupta, M.; and Varma, V. 2019. MVAE: Multimodal Variational Autoencoder for Fake News Detection. In *The World Wide Web Conference*, 2915–2921. ACM.

Leskovec, J.; and Krevl, A. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data.

Moncel, J. 2006. On graphs on n vertices having an identifying code of cardinality log2 (n+ 1). *Discrete Applied Mathematics* 154(14): 2032–2039.

Paluch, R.; Gajewski, Ł. G.; Hołyst, J. A.; and Szymanski, B. K. 2020. Optimizing sensors placement in complex networks for localization of hidden signal source: A review. *Future Generation Computer Systems* 112: 1070–1092.

Potthast, M.; Kiesel, J.; Reinartz, K.; Bevendorff, J.; and Stein, B. 2017. A stylometric inquiry into hyperpartisan and fake news. *arXiv preprint arXiv:1702.05638* .

Qi, P.; Cao, J.; Yang, T.; Guo, J.; and Li, J. 2019. Exploiting Multi-domain Visual Information for Fake News Detection. *arXiv preprint arXiv:1908.04472* .

Racz, M. Z.; and Richey, J. 2020. Rumor source detection with multiple observations under adaptive diffusions. *IEEE Transactions on Network Science and Engineering* .

Ray, S.; Ungrangsi, R.; Pellegrini, D.; Trachtenberg, A.; and Starobinski, D. 2003. Robust location detection in emergency sensor networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 2, 1044–1053. IEEE.

Rossi, R. A.; and Ahmed, N. K. 2016. An Interactive Data Repository with Visual Analytics. *SIGKDD Explor.* 17(2): 37–41. URL http://networkrepository.com.

Sen, A.; Goliber, V. H.; Zhou, C.; and Basu, K. 2018. Terrorist Network Monitoring with Identifying Code. In *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*, 329–339. Springer.

Shah, D.; and Zaman, T. 2011. Rumors in a network: Who's the culprit? *IEEE Trans. on information theory* 57(8): 5163–5181.

Shi, B.; and Weninger, T. 2016. Fact checking in heterogeneous information networks. In *Proceedings of the 25th International Conference Companion on World Wide Web*, 101–102. International World Wide Web Conferences Steering Committee.

Shu, K.; Sliva, A.; Wang, S.; Tang, J.; and Liu, H. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter* 19(1): 22–36.

Shu, K.; Wang, S.; and Liu, H. 2018. Understanding user profiles on social media for fake news detection. In *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 430–435. IEEE.

Spinelli, B.; Celis, L. E.; and Thiran, P. 2017. A general framework for sensor placement in source localization. *IEEE Transactions on Network Science and Engineering* 6(2): 86–102.

Suomela, J. 2007. Approximability of identifying codes and locating-dominating codes. *Inf. Processing Letters* 103(1): 28–33.

Tacchini, E.; Ballarin, G.; Della Vedova, M. L.; Moret, S.; and de Alfaro, L. 2017. Some like it hoax: Automated fake news detection in social networks. *arXiv preprint arXiv:1704.07506* .

Tang, W. 2020. Identifying misinformation and their sources in social networks .

Vazirani, V. V. 2013. *Approximation algorithms*. Springer Science & Business Media.

Wang, Y.; Ma, F.; Jin, Z.; Yuan, Y.; Xun, G.; Jha, K.; Su, L.; and Gao, J. 2018. Eann: Event adversarial neural networks for multi-modal fake news detection. In *Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining*, 849–857. ACM.

Wang, Z.; Wang, C.; Pei, J.; and Ye, X. 2017. Multiple source detection without knowing the underlying propagation model. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Xiao, Y.; Hadjicostis, C.; and Thulasiraman, K. 2006. The d-identifying codes problem for vertex identification in graphs: probabilistic analysis and an approximation algorithm. In *International Computing and Combinatorics Conference*, 284–298. Springer.

Yang, S.; Shu, K.; Wang, S.; Gu, R.; Wu, F.; and Liu, H. 2019. Unsupervised fake news detection on social media: A generative approach. In *Proceedings of 33rd AAAI Conference on Artificial Intelligence*.

Zhou, Y.; Wu, C.; Zhu, Q.; Xiang, Y.; and Loke, S. W. 2019. Rumor source detection in networks based on the SEIR model. *IEEE access* 7: 45240–45258.

Zhu, K.; Chen, Z.; and Ying, L. 2016. Catch'em all: Locating multiple diffusion sources in networks with partial observations. *arXiv preprint arXiv:1611.06963* .

Zhu, K.; and Ying, L. 2014. Information source detection in the SIR model: A sample-path-based approach. *IEEE/ACM Transactions on Networking* 24(1): 408–421.