Misinformation in Online Social Networks: Detect Them All with a Limited Budget

HUILING ZHANG, MD ABDUL ALIM, and XIANG LI, University of Florida
MY T. THAI, Ton Duc Thang University and University of Florida
HIEN T. NGUYEN, Ton Duc Thang University

Online social networks have become an effective and important social platform for communication, opinions exchange, and information sharing. However, they also make it possible for rapid and wide misinformation diffusion, which may lead to pernicious influences on individuals or society. Hence, it is extremely important and necessary to detect the misinformation propagation by placing monitors.

In this article, we first define a general misinformation-detection problem for the case where the knowledge about misinformation sources is lacking, and show its equivalence to the influence-maximization problem in the reverse graph. Furthermore, considering node vulnerability, we aim to detect the misinformation reaching to a specific user. Therefore, we study a τ-Monitor Placement problem for cases where partial knowledge of misinformation sources is available and prove its $\#P$ complexity. We formulate a corresponding integer program, tackle exponential constraints, and propose a Minimum Monitor Set Construction (MMSC) algorithm, in which the cut-set$_2$ has been exploited in the estimation of reachability of node pairs. Moreover, we generalize the problem from a single target to multiple central nodes and propose another algorithm based on a Monte Carlo sampling technique. Extensive experiments on real-world networks show the effectiveness of proposed algorithms with respect to minimizing the number of monitors.

Categories and Subject Descriptors: C.2.2 [**Computer-Communication Networks**]: Network Management; G.1.6 [**Optimization**]: Constrained Optimization; G.2.2 [**Graph Theory**]: Network Problems

General Terms: Algorithms, Experimentation, Performance

Additional Key Words and Phrases: Misinformation detection, monitor placement, online social networks

ACM Reference Format:
Huiling Zhang, Md Abdul Alim, Xiang Li, My T. Thai, and Hien T. Nguyen. 2016. Misinformation in online social networks: Detect them all with a limited budget. ACM Trans. Inf. Syst. 34, 3, Article 18 (April 2016), 24 pages.
DOI: http://dx.doi.org/10.1145/2885494

1. INTRODUCTION

Online social networks (OSNs), such as Facebook, Twitter, and Google+, have provided a powerful means of interaction and communication among people. Through network links, everyone from everywhere can create, spread, and acquire contents at the same

This work is supported in part by the NSF grant CCF-1422116 and NSF Career Award 0953284.
Authors' addresses: H. Zhang, M. A. Alim, and X. Li, Department of Computer & Information Science & Engineering, University of Florida, Gainesville, Florida 32611; emails: {huiling, alim, xixiang}@cise.ufl.edu; My T. Thai (corresponding author), Division of Algorithms and Technologies for Networks Analysis, Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Vietnam, and Department of Computer & Information Science & Engineering, University of Florida, Gainesville, Florida 32611; emails thaitramy@tdt.edu.vn and mythai@cise.ufl.edu; H. T. Nguyen, Ton Duc Thang University, Ho Chi Minh City, Vietnam; email: hien@tdt.edu.vn.

DOI: http://dx.doi.org/10.1145/2885494

time. The continuous growth and increasing popularity of OSNs, have brought forth a great deal of attention. A growing number of people integrate popular OSNs into their daily life and regard them as one of the main information sources. More importantly, interactions in OSNs can greatly affect political or social movements and debates. For example, OSNs, like Twitter and Facebook, played an important role in the 2008 U.S. presidential elections [Hughes and Palen 2009] and the 2010 Arab Spring [Wolfsfeld et al. 2013].

While innovations and trustworthy recommendations spread among users in OSNs, the information carrying false or inaccurate claims is also ubiquitous. Online social communities have recently become the target of many attackers. The dissemination of misinformation can result in undesirable social effects and even economic losses [Nguyen et al. 2012; Budak et al. 2011; Tripathy et al. 2010]. For instance, many Twitter tweets with the misinformation about the Chilean earthquake emerged after the disaster in 2010 [Mendoza et al. 2010]. The tweets with false statements about President Obama's injury in April 2013 caused the instabilities in financial markets [Jin et al. 2013]. Also, if a lot of misinformation occupies OSNs, it would be pretty difficult to believe in information in OSNs and we may lose this highly promising medium. As a result, it is of great practical importance to timely detect the dissemination of misinformation in OSNs before it leads to disruptive effects.

Users' activities in OSNs can be captured by their posts, comments, and sharing history. Based on gathered information, the misinformation can be automatically detected. Qazvinian et al. [2011] showed the effectiveness of features, including content-based, network-based, and microblog-specific memes, in correctly identifying misinformation. In Kwon et al. [2013], temporal, structural, and linguistic features are used to identify rumors. We refer to these techniques as "monitors" placed on users. That being said, once a monitor is deployed on a user, it can detect the misinformation by monitoring and analyzing user's activities. Considering the large size of OSNs, it is too expensive to monitor activities of all users. The practical way is to monitor some selected users so as to maximize the successful detection of misinformation. Therefore, the question here is to find the optimal strategy of monitor placement.

We consider two practical scenarios in this article. First of all, it is not uncommon that users in OSNs are exposed to various kinds of misinformation. By placing monitors, we can detect the misinformation that exists and propagates in OSNs. Notice that each node in OSNs could be the source of misinformation. Without proper knowledge to identify from which node misinformation starts, all nodes should be treated equally as the source. In this scenario, we show the equivalence between the monitor placement and the influence maximization in the corresponding reverse graph, which asks for a seed set, such that the expected number of users influenced by this set is the largest possible [Kempe et al. 2003].

Furthermore, the information people acquired from communications and interactions is a powerful and persistent force affecting their behaviors. As usual, there are always some users in OSNs who we want to protect from being influenced by the misinformation. For example, we do not want parents to be influenced by misinformation about vaccination of some diseases, which makes them withhold necessary immunizations from their children [Lewandowsky et al. 2012]. Another example would be keeping underaged teenagers away from the misinformation about violence. However, the timely detection of misinformation is a prerequisite for taking good, protective actions in time. This motivates us to design a monitor placement strategy to detect misinformation before it can reach vulnerable central users. For one single central user, we define a τ-Monitor Placement problem, which aims to minimize the number of required monitors for timely detection of misinformation arriving at the central node. To the best of our knowledge, this is the first attempt to address the above problem in such a scenario. We prove that τ-Monitor Placement is #P-hard, and then design an effective algorithm to solve it. In a more general scenario, there is more than one vulnerable node through which the misinformation may pass. To effectively tackle this scenario, we design a new algorithm using a Monte Carlo sampling technique.

Our contributions in this article are summarized as follows:

—We define a misinformationdetection problem and prove its equivalence to the influencemaximization problem in the reverse graph.

—With a view to finding an optimal monitor set for detecting misinformation in a timely fashion, we define a τ-Monitor Placement problem and prove its #P-hardness.

—In an attempt to obtain an optimal solution for small size instances, we formulate an Integer Program and use a weighted-averaging method to deal with an exponential number of constraints. To tackle large-scale instances, we devise a Minimum Monitor Set Construction (MMSC) algorithm to efficiently identify necessary minimum number of monitors.

—To handle multiple central nodes, we propose a Monitor Set for Multiple Nodes (MSMN) algorithm. We also show the probabilistic guarantee of the reachability obtained through this method.

—The performance of our proposed solutions is validated on real-world social traces of Twitter, Epinion, and Slashdot datasets. In each of the instances, we observe that our proposed scheme outperforms other methods in terms of minimizing the monitors required to protect central nodes.

The rest of this article is organized as follows. We introduce the model of information propagation with monitors in Section 2. Section 3 defines a misinformation-detection problem and shows it is equivalent to the influence-maximization problem in the reverse graph. Narrowing down potential misinformation sources, a τ-Monitor Placement problem is defined and its complexity is proven in Section 4. An IP formulation is proposed and an effective algorithm is proposed to solve the τ-Monitor Placement in Section 5. We propose a new algorithm using sampling technique in Section 6 for misinformation detection when there are multiple vulnerable central nodes. Experimental results are given in Section 7. Section 8 discusses the related work and, finally, Section 9 concludes the article.

2. MODEL

OSNs are becoming a huge dissemination platform, allowing rapid information exchange and influencing a large population in a short period of time. Starting from source nodes, misinformation may spread out along edges of networks. In order to find a strategy to detect misinformation propagating in OSNs, in this section, we first introduce misinformation-propagation models, which specify how misinformation diffuses or propagates from a person to other persons.

An OSN is modeled as a directed graph $G = (V, E)$, where V is the set of $|V| = n$-users in this OSN, E represents social interactions among users and the edge orientation indicates the direction of influence. Each edge $e = (u, v) \in E$ is associated with a probability p_{uv} that v gets the misinformation from u. Note that there are two types of social relationships in OSNs, namely, unidirectional and bidirectional. Directed edges can describe unidirectional relationships. For any two users involved in bidirectional relationships, u and v, transmission probabilities p_{uv} and p_{vu} are usually unequal, which can be represented as directed edges with different weights. Moreover, an undirected graph can be transformed into a directed graph by replacing each edge with two directed edges with the same spreading probability as the original edge.

Diffusion Model. To describe how the misinformation propagates, we classify nodes in OSNs into two categories: active and inactive. Nodes are active when they receive misinformation and really accept it; otherwise, they are inactive. Two well-known classes of theoretical diffusion models have been widely explored: the linear threshold (LT) model [Pathak et al. 2010a; Chen et al. 2010] and the independent cascade (IC) model [Kempe et al. 2003; Leskovec et al. 2007]. In LT models, each edge has a weight p_{uv} chosen uniformly at random, and each node u is assigned a threshold θ_u uniformly at random from [0, 1]. At any time-step t, an inactive node u gets active if the weighted sum of its active neighbors exceeds its threshold θ_u.

In the IC model, another widely adopted model, if a node accepts misinformation, it becomes active; otherwise, it remains inactive. Each edge has an influence probability and misinformation is propagated by active nodes independently activating their inactive neighbors. Specifically, a newly active node u gets a single chance to activate every currently inactive outgoing neighbor v, succeeding with a probability p_{uv}. Let S be a set of potential misinformation sources. Starting from the source set S, the process of misinformation propagation proceeds till no more nodes can be activated by misinformation.

Considering the independent influence of misinformation from a node to its neighbors, we adopt the IC model in this article. Note that by modifying the calculation of reachability, solutions proposed in this article can be extended to handle different diffusion models. For example, LT model has been proven to be equivalent to reachability via live-edge paths and the reachability of any pair of nodes can be calculated [Kempe et al. 2003; Chen et al. 2010]. In the research area of viral marketing and misinformation blocking, the input social graph is assumed to be with edges labeled "spreading probabilities" [Kempe et al. 2003; Chen et al. 2010; Budak et al. 2011; Nguyen et al. 2012]. The learning of edge probabilities is not in the scope of this article. Considerable effort has been made in the area of link characterization and learning-influence probabilities [Lin et al. 2013; Goyal et al. 2010; Catanese et al. 2011; Jiang et al. 2013; Meo et al. 2013].

3. MISINFORMATION DETECTION

Without prior knowledge about the sources of misinformation, every node in OSNs is potentially having the chance to propagate misinformation. We want to detect misinformation propagating in OSNs by placing monitors. However, in reality, the resource to place monitors is limited.

Considering the limited budget, our goal is to find k locations to place monitors so as to maximize the detection probability. If there is a path ι from u to v, the probability that misinformation can propagate from u to v through ι is $\prod_{e\in l} p_e$. For a set of paths L, let L^* be the event that all paths of L do not let the information propagate through. Considering the propagation starting from a single node i, the probability of misinformation detection by placing monitors on $A \subset V$ is $D(i, A) = 1 - Pr(L^*_{iA})$, where L_{iA} is the set of paths starting from node i arriving at one of the nodes in A. Therefore, the expected detection probability by placing monitors on A is

$$D(A) = \sum_{i\in V} P(i) D(i, A),$$

where $P(i)$ is the probability that misinformation starts from node i. With the objective to detect the misinformation with the given resources, we define the following problem.

Definition 3.1 (***Misinformation Detection (MD)***). Given a directed graph $G = (V, E)$ representing an OSN, where each edge $e \in E$ is associated with a transmission probability p_e, this problem aims to find a monitor set A of k monitors, such that the expected detection probability is maximized, i.e., $A^* = argmax_{A\subset V:|A|\leq k} D(A)$.

Owing to the lack of knowledge regarding which nodes are potential sources of misinformation, we assume each node having the same probability to be a source. (We will consider later the case when knowledge of misinformation sources is available). In this case, the optimal monitor set is the same as the optimal seeding set of influence maximization on a "reverse" graph (reversing the direction of each edge). We prove the equivalence of the MD problem to the influence-maximization problem, which aims to find a small seeding set in a social network so that their aggregated influence is maximized [Chen et al. 2010].

THEOREM 3.2. *When all nodes are of the same probability to be the misinformation source, the MD problem is equivalent to the influence-maximization problem on a reverse graph.*

PROOF. Considering graph G associated with propagation probabilities, we can generate sample graphs. A sample graph (or a realization) of G is generated by selecting each edge $e \in E$, independently, with probability p_e, to present in this sample. There are $M = 2^{|E|}$ possible sample graphs, denoted as $g_1, g_2, \ldots, g_M$. Let h_l denote the probability that a sample graph g_l is generated. Then, $h_l = Pr[G = g_l]$. For the misinformation flow starting from node i, if there is at least one node in set A that can be reached from i in the sample graph g_l, we say, this misinformation can be detected by A in g_l, which is denoted as $x_{il}(A) = 1$; otherwise, $x_{il}(A) = 0$. Thus, the probability that misinformation from i can be detected by placing monitors on set $A \in V$ is $D(i, A) = \sum_l h_l \times x_{il}(A)$. With respect to the detection probability of misinformation from a single unknown source, the reward of placing monitors on A is

$$D(A) = \sum_i P(i) \sum_l h_l \times x_{il}(A). \tag{1}$$

When $P(i)$ are same for all nodes, changing the order of i and l in the formula, we have

$$A^* = \underset{A \subset V:|A| \le k}{argmax} \sum_l \left(h_l \times \sum_i x_{il}(A) \right).$$

If we reverse all edges in G, $\sum_i x_{il}(A)$ equals the total number of nodes activated by the influence-propagation process when A is the initial target set of sample graph g_l. Thus, optimal locations for monitors are the same as the optimal seeding set. □

In the case of multiple misinformation sources, i.e., each node i can be a source node with a probability $P(i)$ and these probabilities are independent, $D(A)$ is no longer a probability but can be viewed as a reward of placing monitors on node set A. Theorem 3.2 still holds for multiple sources. Based on Theorem 3.2, the MD problem can be solved by reversing all edges in G and applying the solution of the influence-maximization problem. The solution obtained by this approach is within $(1-1/e)$ factor of the optimal one as the influence maximization has a submodularity property [Kempe et al. 2003].

4. MINIMAL NUMBER OF MONITORS TO DETECT MISINFORMATION

In some cases, the knowledge which is helpful for locating the sources of misinformation is available. For example, inaccurate posts extolling the virtues of a new product are most probably from product marketers; political astroturf memes on Twitter are carried out by users or organizations in some political campaigns; and so forth [Mendoza et al. 2010; Ratkiewicz et al. 2011]. There is considerable work on locating the information sources [Prakash et al. 2012; Shah and Zaman 2011; Zhu and Ying 2014; Luo et al. 2013; Zhang et al. 2015]. Given the knowledge about the potential sources of misinformation,

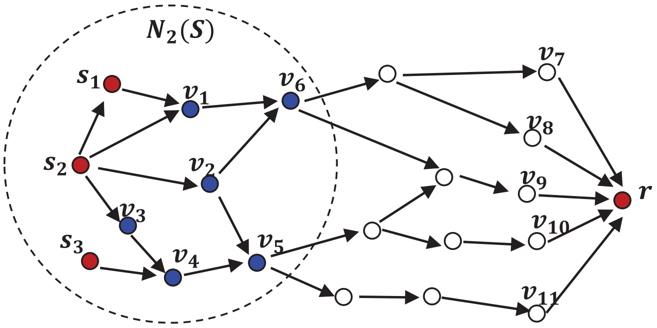


Fig. 1. Here, the source set is $S = \{s_1, s_2, s_3\}$ and node r is a vulnerable central node. When $\delta = 2$, we have the candidate nodes of monitors $N_2(S)$. The best location for monitors is $\{v_5, v_6\}$.

in this section, we aim to address the scenario where we want to timely detect the propagating misinformation before it reaches a subset of users.

4.1. Problem Definition

We first assume that there is a single vulnerable central node we want to protect from the influence of misinformation. This assumption will be relaxed and addressed in a later section. If we only place monitors on nodes incident to this important node, we can detect the misinformation traversing to it but a large number of nodes might have already been influenced by misinformation before we detect it. And it has been suggested that memory for misinformation is more resistant to forgetting than memory for actual details of a real event [Zaragoza et al. 2006]. The difficulty in removing misinformation from infected persons makes the detection of misinformation at its early stage greatly important. Thus, our aim is to detect the misinformation before it travels a long distance.

Let us denote S as a set of potential sources of misinformation, and $N_\delta(S)$ as the set of nodes which are within δ distance from any node in S. Specifically, when $\delta = 1$, $N_\delta(S)$ is the set of direct outgoing neighbors of nodes in S. We define a τ-Monitor Placement problem as follows.

Definition 4.1. **τ-Monitor Placement (τ-MP):** Given a graph $G(V, E)$, where V is the set of nodes and E represents the set of edges, transmission probabilities $P_E = \{p_e\}$ associated with each $e \in E$, a subset $S \subset V$ of potential sources of misinformation, and a vulnerable central node r that we want to protect, the problem aims to minimize the number of monitors $A \subseteq N_\delta(S) \setminus S \subset V$, where $\delta > 0$ and $N_\delta(S)$ is δ-distance neighbors of S, such that the mis-detection probability of information at node r is not greater than τ, where $0 < \tau \leq 1$.

For the example shown by Figure 1, if we want to protect the central node r, it is wiser to place monitors on $\{v_5, v_6\}$ than on the neighbor of node r, i.e., $\{v_7, v_8, v_9, v_{10}, v_{11}\}$, or on the direct neighbors of S, i.e., $\{v_1, v_2, v_3, v_4\}$.

4.2. Complexity and Analysis

In this section, we show that the τ-MP problem is #P-hard, by reducing from the $Conn_{s,t}(G)$ problem. Note that #P is the set of counting problems whose corresponding decision problem is in NP. Obviously, a #P problem is at least as hard as the corresponding NP problem.

THEOREM 4.2. *The τ-MP problem is #P-hard.*

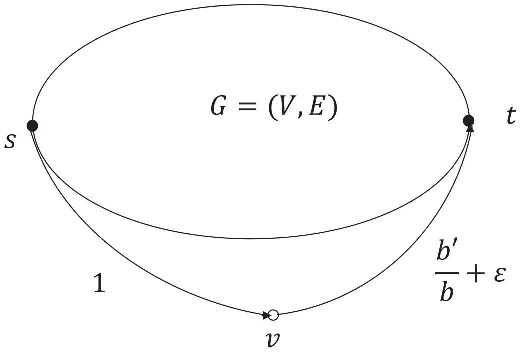


Fig. 2. The constructing instance in the reduction from $Conn_{s,t}$. G, s, t is an instance of the $Conn_{s,t}$ problem. A new node v is added and connected with s and t, with probability 1 and $\frac{b}{b'} + \varepsilon$, respectively.

Proof. To show the $\#P$-hardness, we reduce from the $Conn_{s,t}(G)$ problem, defined in the following.

Computing the connected probability of any two nodes s and t in a probabilistic graph $G = (V, E)$ with edge probability p_e for each $e \in E$ has been proved to be $\#P$-complete [Valiant 1979]. We denote this probability as $Conn_{s,t}(G)$ and also refer the computation of $Conn_{s,t}(G)$ as the $Conn_{s,t}(G)$ problem. The $Conn_{s,t}(G)$ problem can be polynomially solved if we have a procedure to determine that $Conn_{s,t}(G) \leq b'/b$ in a graph G for any integer $b' \leq b$. Since all transmission probabilities p_e are rational numbers represented by a numerator and denominator which are integers, we can let b be the least common multiple of all the denominators, such that a simple binary search from 1 to b can be finished within a polynomial time with respect to the input size.

Consider an arbitrary instance of the $Conn_{s,t}(G)$ problem, where $G = (V, E)$, s, t and all edge probabilities p_e are given. As Figure 2 shows, construct a graph $G' = (V', E')$ by adding a new node v to G and connecting it with s and t, with transmission probability 1 and $b'/b + \varepsilon$, respectively. Let $\varepsilon \leq 1/b$. Note that transmission probabilities for remaining edges in G' are the same with those in G. Let the potential source set $S = s$, the central node $r = t$ with mis-detection threshold $\tau = b'/b$, and $\delta = 1$. This construction can be finished in polynomial time.

Assume that an $\mathcal{A}$ is a polynomial-time, algorithm-solving τ-MP problem. Consider two cases:

—If $\mathcal{A}$ returns the monitor set A whose size is equal to 1, we only need to place the monitor on node v. That is, $Conn_{s,t}(G) \leq b'/b$;
—If $\mathcal{A}$ returns the monitor set A with a size larger than 1, besides v, some nodes are chosen as locations for monitors. Then, we know $Conn_{s,t}(G) > b'/b$.

Therefore, $\mathcal{A}$ can be used to decide if $Conn_{s,t}(G)$ is less than b'/b, implying that our τ-MP problem is at least as hard as $Conn_{s,t}(G)$ problem. □

5. SOLVING THE τ-MP PROBLEM

In this section, we first propose an integer programming (IP) for the τ-MP problem, which will help us obtain optimal solutions for small-size instances. The main challenges to solve this IP is an exponential number of constraints, of which we will propose a novel method, namely weighted-averaging, to overcome this difficulty. Furthermore, we concentrate on the design of an effective heuristic to solve the large-scale τ-MP instances.

5.1. Optimality of the τ-MP Problem

Firstly, an integer-programming problem is formulated in the following.

For each node $u \in V$, we define a variable z_u to indicate whether node u is selected as a monitor location.

$$z_u = \begin{cases} 1, & \text{if node } u \text{ is selected as a monitor location} \\ 0, & \text{otherwise} \end{cases}$$

Once we place a monitor on node u, misinformation passing through it can be detected. Only paths without any monitor can lead to misinformation propagation without detection. Accordingly, we can view paths with at least one monitor as being blocked. After we select a monitor set, all edges incident with any monitors are blocked. We consider the connectivity of nodes with respect to edges not being blocked in misinformation propagation. A variable t_{uv} is defined for each pair of nodes (u, v) to indicate whether node u and node v are disconnected.

$$t_{uv} = \begin{cases} 1, & \text{if node } u \text{ and node } v \text{ are disconnected} \\ 0, & \text{otherwise} \end{cases}$$

The misinformation propagation follows the IC model in which each edge is associated with an influence probability. In an equivalent view of the Independent Cascade process [Kempe et al. 2003], the outcome of the attempt that an active node u influences its inactive neighbor v can be viewed as being determined by flipping a coin bias p_{uv}.

$$\xi_{uv} = \begin{cases} 1, & \text{active node } u \text{ successfully activates neighbor } v \\ 0, & \text{otherwise} \end{cases}$$

Thus, for $(u, v) \in E$, $Pr[\xi_{uv} = 1] = p_{uv}$ and $Pr[\xi_{uv} = 0] = 1 - p_{uv}$.

During the propagation process, the value of ξ_{uv} is either 1 or 0, and thus, we can compute the mis-detection probability of misinformation from sources S to any central node after placing monitors. Only the misinformation propagate through paths without any monitor is mis-detected. The mis-detection probability at a central node r is equal to the ratio between the pairwise connectivity and the total possible connectivity of S and r.

Every time when we flip coins for all pairs of neighbors in G, we generate a sample of G with only edges in G for which the coin-flipindicated attempt will be successful. Denote the probability of the realization G^l of G as $Pr[G^l]$. Considering finite realizations of G, we formulate the following mixed integer programming, denoted by MIP$_F$.

$$\min \sum_{u \in N_\delta(S)} z_u, \tag{2}$$

$$s.t. \quad \sum_{l=1}^{W} Pr[G^l] \sum_{u \in S} \left(1 - t_{ur}^l\right) \le \tau n_S \tag{3}$$

$$t_{uv}^l \le z_u + z_v + 1 - \xi_{uv}^l, (u, v) \in E, l = 1 \cdots W \tag{4}$$

$$t_{uv}^l + t_{vw}^l \ge t_{uw}^l, (u, v) \in E, w \in V \tag{5}$$

$$z_u \in \{0, 1\}, u \in N_\delta(S)$$
$$z_u \in \{0\}, u \notin N_\delta(S)$$
$$t_{uv}^l \in \{0, 1\}, l = 1 \cdots W,$$

where $W = 2^m$ is the number of possible realizations of G, and $n_S = |S|$ is the number of potential sources of misinformation. The objective of Equation (2) is to minimize the number of monitor set sizes. The exponential number of constraints is the major challenge in solving this MIP$_F$. To solve it, we first construct a linear relaxation. We apply a weighted-averaging of all constraints in MIP$_F$. Constraints involving the sample G^l are given the weight $Pr[G^l]$. Thus, we reduce the constraints in (4) to a single constraint

$$t_{uv} \leq z_u + z_v + 1 - \sum_{G^l} Pr[G^l]\xi^l_{uv},$$

which can be further simplified as

$$t_{uv} \leq z_u + z_v + 1 - p_{uv}.$$

Following the same way, we average constraints (3) and (5), and obtain the relaxation of MIP$_F$ as follows, denoted as MIP$_R$.

$$\begin{aligned}
\min \quad & \sum_{u \in N_\delta(S)} z_u, && (6)\\
s.t. \quad & \sum_{u \in S}(1 - t_{ur}) \leq \tau n_S && (7)\\
& t_{uv} \leq z_u + z_v + 1 - p_{uv}, (u, v) \in E && (8)\\
& t_{uv} + t_{vw} \geq t_{uw}, (u, v) \in E, w \in V && (9)\\
& z_u \in \{0, 1\}, u \in N_\delta(S)\\
& z_u \in \{0\}, u \notin N_\delta(S)\\
& t_{uv} \in [0, 1], u, v \in V
\end{aligned}$$

By this weighted-averaging, we can reduce the number of constraints, shown as Lemma 5.1.

Lemma 5.1. *The exponential number of constraints associated with MIP$_F$ can be reduced to $O(n^3)$ constraints.*

Proof. As the constraint (4) in MIP$_F$ shows, we have a constraint for each edge in each realization G^l. Since $W|E| = m2^m$, clearly, constraint (4) corresponds to an exponential number of constraints. We can also easily identify the exponential number of constraints represented by triangle inequality (5). However, MIP$_R$ has $O(n^3)$ constraints owing to the triangle inequality constraints, shown by (9). Thus, the number of constraints in MIP$_F$ is substantially reduced. □

Lemma 5.2. *The optimal objective value of MIP$_R$ is the lower-bound on the optimal objective value of MIP$_F$.*

Proof. We construct a feasible solution $(\tilde{z}, \tilde{t})$ of MIP$_R$ which gives an objective equal to the optimal objective of MIP$_F$. Let $(\hat{z}, \hat{t}^1, \cdots, \hat{t}^W)$ be an optimal solution of the MIP$_F$. Construct a solution $(\tilde{z} = \hat{z}, \tilde{t} = \sum_{l=1}^{W} Pr[G^l]\hat{t}^l)$. The objective value of MIP$_R$ is the same as the optimal objective of MIP$_F$. The equality

$$\sum_{u \in S}(1 - \tilde{t}_{ur}) = \sum_{u \in S}\left(1 - \sum_{l=1}^{W} Pr[G^l]\hat{t}^l_{ur}\right) = \sum_{l=1}^{W} Pr[G^l] \sum_{u \in S}\left(1 - \hat{t}^l_{ur}\right)$$

holds due to the fact that $Pr[G^l]$ adds up to one. Since constraint (3) is satisfied, we know constraint (7) is satisfied, i.e., $\sum_{u \in S}(1 - \tilde{t}_{ur}) \leq \tau n_s$. We can verify that other constraints are also satisfied. □

5.2. Cut-set$_2$-Based Algorithm

As we know, there are usually a large number of users participating in OSNs. However, the existence of polynomial-time algorithms for solving the τ-MP problem in large-scale OSNs is ruled out by #P-hardness. Effective heuristics are needed for large size τ-MP instances.

With the objective to minimize the number of monitors, we propose an efficient, greedy algorithm MMSC, which aims to place monitors on nodes that would help in detecting as many misinformation flows as possible. The basic idea of our strategy is to iteratively select nodes in $N_\delta(S)$ that have the most contribution in misinformation propagation. In this process, we continue to add nodes from $N_\delta(S)$ into the monitor set till no more nodes are needed to ensure the mis-detection probability is no greater than the given risk value τ.

In order to evaluate the contribution of each candidate node u to misinformation detection, obviously, we need to efficiently consider all paths from misinformation sources to the central node r passing through u. Given a path from s to t and an intermediate node v of this path, we can divide this path into two half-paths: one from s to v, and one from v to t. In the evaluation of the contribution of node v in misinformation propagation, we should take both half-paths into consideration. However, the computation of connectedness of two nodes is #P-complete, and thus, it is intractable for OSNs of large size. Therefore, an effective procedure to estimate the connectedness of a pair of nodes (u, v) is needed. In the rest of this section, we define the cut-set$_2$ of our interest and propose an algorithm exploiting cut-set$_2$ in the estimation of connectedness.

A normal cut-set of a graph G is a set of edges, all of whose removal makes G disconnected, but the removal of some of them does not disconnect G. Interested in the connectedness from a node s to a node t, we define a *cut-set$_2$* for s and t as the set of edges whose removal can separate them, i.e., $c_{st}(S, T) = \{(u, v)|(u, v) \in E, s, u \in S \text{ and } t, v \in T\}$, and (S, T) is a partition of V. Any partition that separates s and t is of interest when we discuss the connectedness from node s to node t. We define $C_{st} = \{c_{st}(S, T)\}$ as the set of all possible cut-set$_2$ of node s and node t. Without ambiguity in the context, we simply denote $c_{st}(S, T)$ as c_{st}. Let $\bar{c_{st}}$ be the event that all edges in a cut c_{st} fail, in terms of misinformation propagation. Since the probability of $\bar{c_{st}}$ is $Pr(\bar{c_{st}}) = \prod_{e\in c_{st}}(1 - p_e)$, the probability τ_{st} that the misinformation from s can reach t is $\tau_{st} = 1 - Pr(\cup_{c_{st}\in C_{st}} \bar{c_{st}})$.

The calculation of τ_{st} needs scanning for all cut-set$_2$, which is intractable. We can obtain an upper-bound of τ_{st} by replacing C_{st} with a subset ${C_{st}}'$ of C_{st}. A good selection of ${C_{st}}'$ could give a close estimation of τ_{st}. And the upper-bound can be efficiently computed if ${C_{st}}'$ is the set of disjointed cut-set$_2$. And thus,

$$1 - Pr\left(\cup_{c_{st}\in {C_{st}}'} \bar{c_{st}}\right) \geq \tau_{st}.$$

Note that the number of disjointed cut-set$_2$ separating s and t is equal to the distance from s to t, which is defined as the length of the shortest path from s to t. The details of MMSC are shown in the Algorithm 1 pseudo-code.

In Algorithm 1, we first assess the contribution of each candidate node and select a node with the largest contribution into monitor set A. The contribution of a candidate node i includes two parts; namely, the probability of misinformation transmits from set S to node i, and that from i to the central node r. Note that $\bar{p}_{oi}$ and $\bar{p}_{ir}$ in Line 12 are their upper-bound. The product of these two numbers is the estimation of the contribution of node i in misinformation propagation, i.e., the probability that misinformation flows from sources to central, going through this intermediate node i. To avoid some unnecessary monitors, we perform a refinement procedure $\Gamma()$ (Line 14). Specifically, we check whether each newly added monitor covers some already present monitors in

ALGORITHM 1: Minimum Monitor Set Construction (MMSC)

```
Input: $G(V, E)$, $P_E$, $\delta$, $\tau$, misinformation source set $S$ and a central node $r$.
Output: Monitor set $A$.
 1: $A \leftarrow \emptyset$, $B \leftarrow N_\delta(S)$
 2: while $\tau(A) > \tau$ do
 3:   for each $i \in B$ do
 4:     $F \leftarrow (o, i) \cap (i, r)$
 5:     for each node pair $(s, t) \in F$ do
 6:       for $j = 1$ to $d_{st}$ do
 7:         $C_j \leftarrow \{(u, v) | d_{su} = j - 1 \wedge d_{vt} < \infty\}$
 8:       end for
 9:       $\bar{p}_{st} = \prod_{j=1}^{d_{st}}(1 - \prod_{e \in C_j}(1 - p_e))$
10:     end for
11:   end for
12:   $v \leftarrow argmax_{i \in B} \bar{p}_{oi} \cdot \bar{p}_{ir}$
13:   $A \leftarrow A \cup v$, $B \leftarrow B \setminus v$
14:   Refinement procedure $\Gamma(A)$
15: end while
16: Return $A$
```

the current monitor set A. If so, we remove those monitors from the monitor set. After a node is selected, the misinformation passing through that node can be detected, and thus, information flows from its incoming neighbors or to its outgoing neighbors do not contribute to the mis-detection probability. We safely remove this node from the graph and recalculate the contribution of the remaining candidate nodes. The selection process proceeds till no more nodes are required to satisfy a given detection threshold τ. Finally, a monitor set A is returned at the end of our algorithm.

During the construction of the monitor set, we need to evaluate the contribution of all nodes in $N_\delta(S)$. Additionally, in order to find all cut-set$_2$, we need to run a breadth-first search (BFS). In the worst case, the MMSC algorithm runs in time $O(Kn_1^2 n(n + m))$, where $n_1 = |N_\delta(S)|$ and K is the number of iterations of the main loop (Lines 2∼15).

6. MULTIPLE CENTRAL NODES

In the previous section, we only deal with the τ-Monitor Placement problem with a single central node. Considering that there might be a group of users that we want to protect, we generalize the τ-MP problem to monitoring multiple central nodes and design a new algorithm, accordingly, to address it in this section.

6.1. Monitor Selection for Multiple Central Nodes

Denote the set of vulnerable central nodes as R, and each $i \in R$ is associated with a threshold τ_i for mis-detection probability. When these thresholds are the same, we can simply apply the MMSC algorithm. In general cases, these central nodes have different endurance of misinformation, and thus, we define different mis-detection thresholds to reflect this fact. To tackle this problem, we design a MSMN algorithm, shown as follows.

It is highly possible that a source node is connected with several central nodes, which means that the misinformation from this source node may propagate to these central nodes. Therefore, in the selection of monitors, we need to take into consideration the threat from a candidate node to all central nodes. Considering nodes' different thresholds with respect to misinformation endurance, we will "normalize" the threat in evaluating which node leads to the major threat. The MSMN, shown in Algorithm 2,

ALGORITHM 2: Monitor Set for Multiple Nodes (MSMN)

```
Input: Graph $G$, $P_E$, $\delta$, $\tau$, misinformation source set $S$ and central nodes $R$
Output: Monitor set $A$
$A \leftarrow \emptyset$, $B \leftarrow N_\delta(S)$, $X \leftarrow R$
while $\exists \beta_x(A) > \tau_x$ do
    for $x \in X$ do
        if $\beta_x(A) \leq \tau_r$ then
            $X \leftarrow X \backslash \{x\}$
        end if
    end for
    select $u^* \leftarrow argmax_{u \in B} f_X(u)$
    $A \leftarrow A \cup \{u^*\}$, $B \leftarrow B \setminus u^*$
end while
Return $A$
```

iteratively places a monitor on the candidate node, which maximizes function $f_X(u)$, which is defined as

$$f_X(u) = \min_{x \in X} \frac{\beta_x(A) - \beta_x(A \cup \{u\})}{\tau_x},$$

where X is the subset of central nodes whose mis-detection threshold is not satisfied, and $\beta_x(A)$ is a mis-detection probability at central node x, with respect to monitor set A. When there exists any node x with $\beta_x(A)$ greater than τ_x, we need to add more monitors. As the monitor set A grows, the $\beta_x(A)$ decreases.

6.2. Reachability Estimation with Sampling

The calculation of $\beta_x(A)$ is computationally hard. We need an estimator to estimate the mis-detection probability. The cut-set$_2$ can also be used in this estimation; however, considering the size of the central node set and the gap between exact reachability and its upper bound, it will result in a trivial solution. Here, we estimate reachability by sampling with probabilistic guarantee. First, we introduce an estimator of two-node reachability and show how to use it in the estimation of $\beta_x(A)$ later.

ALGORITHM 3: Connectivity Probability Estimator (CPE)

```
Input: $g$, $P_E$, node $u$, $v$, Node set A, $N$
Output: Connectivity probability $\tilde{p}_{uv}$
$Q \leftarrow 0$
Obtain $g_A$ by removing $A$ from $G$
for $k = 1$ to $N$ do
    Generate a sample graph $g_A(k)$ from $g_A$
    do BFS from node $u$ in graph $g_A(k)$
    if $v$ is connected with $u$ then
        $Q = Q + 1$
    end if
end for
Return $Q/N$
```

Given a monitor set A, we can estimate the reachability of each pair of nodes with a probabilistic guarantee. In Algorithm 3, we propose the Connectivity Probability Estimator (CPE) which returns an estimate, Q/N, of the reachability from u to v in the residual graph G after the removal of A in N simulations. If g is the reversed graph of G, A is the monitor set, and u is one of the central nodes, then $\tilde{p}_{uv}$ is the mis-detection probability of the misinformation from v and propagating to central node u. Considering different sizes of networks, we want to guarantee the quality of the estimation from a probabilistic perspective related with the network size. Shown as Lemma 6.1, we have such a probabilistic guarantee.

LEMMA 6.1. *Given a graph G and a monitor set A, let $\tilde{p}$ outputted by Algorithm 3 be an estimate of true value $p = Pr(monitor\ A does\ not\ detectthe\ misinformation\ from\ u\ to\ v)$. When $N \geq \ln(2n^{\delta})/(2\varepsilon^2)$, we can guarantee an error no more than ε, i.e., $|\tilde{p} - p| \leq \varepsilon$, with a probability of at least $1 - 1/n^{\beta}$.*

PROOF. Define integer variables $\{Q_i\}$ to indicate whether or not the misinformation from u can propagate v in i-th simulation in Algorithm 3, i.e., $Q_i = 1$, if misinformation from u propagates to node v without being detected. Let $Q = \sum_{i=1}^{N} Q_i$. Then, $Q = \tilde{p}N$ and $E[Q] = pN$. By Hoeffding's Inequality, we have $Pr(|\tilde{p} - p| \geq \varepsilon) = Pr(|Q - E[Q]| \geq N\varepsilon) \leq 2\exp(-2N\varepsilon^2)$. When $N \geq \ln(2n^{\beta})/(2\varepsilon^2)$, the inequality $Pr(|\tilde{p} - p| \geq \varepsilon) \leq n^{-\beta})$ holds. □

Therefore, the increase of iteration number N can improve the quality of probability estimation. We first generate N samples and the estimation of reachability is based on this set of samples. Inside the main loop of MSMN (Lines 2∼10), with respect to the X, we compute $f_X(u)$ for each candidate's nodes. In the worst case, the running time is $O(n_1 N K |R|(n + m) + Nm)$, where $n_1 = |N_{\delta}(S)|$ and K is the number of iterations of the main loop.

7. EXPERIMENTAL EVALUATIONS

To evaluate the performance of our proposed algorithms, we conduct experiments on real networks to compare the performance of Algorithm 1 and Algorithm 2 with other methods, which will be described later. In addition, we also aim to address the following questions: What is the importance of detection threshold in determining the monitor set? How do the misinformation sources and candidate nodes of monitors affect the selection? How important is source selection method? What is the impact of placing monitors far from the sources? What is the role of the number of vulnerable targeted nodes, i.e., central nodes, playing in the monitor selection?

Dataset. We use real-world social network data from popular OSNs, including Twitter, Epinion, and Slashdot. In all these datasets, nodes represent users while links are the social interaction between users. The Twitter dataset is extracted by unbiased sampling [Cha et al. 2010]. The Epinion and Slashdot datasets are obtained from [http://snap.stanford.edu/data]. Epinion is a who-trusts-whom online social network of a consumer review site Epinions.com. Members of the site can decide whether to "trust" each other. All the trust relationships interact and form the Web of Trust which is then combined with review ratings. Slashdot is a technology-related news website known for its specific user community. The learning of edge probabilities is not in the scope of this article and, considering difficulties in gathering real action propagation traces, edge transmission probability p_e is randomly chosen from $[0, 1]$. Table I shows the statistics of these datasets.

7.1. Single Vulnerable Central Node

In this section, we evaluate the proposed algorithm for a single vulnerable node from several perspectives. For a comparative analysis of the performance of MMSC, we use two other methods for choosing monitor sets such that the mis-detection probability

Table I. Statistics of Three Real Networks

Dataset	Twitter	Epinion	Slashdot
Nodes	88,484	75,879	77,360
Edges	2,364,322	508,837	905,468
Avg. Degree	26.7	6.7	11.7
Type	Directed	Directed	Directed

is less than τ. To this end, we use a centrality-based heuristic measure as well as a random selection of monitors among the $N_\delta(S)$. The latter one serves as the baseline of our comparison, where monitors are chosen randomly from the candidate set. In all of the subsequent experiments, we run the random selection algorithm 100 times and average it out over them to ensure consistency. For centrality-based measure, we choose the degree centrality, i.e., we keep on adding highest-degree nodes among $N_\delta(S)$ until the mis-detection probability reduces below τ.

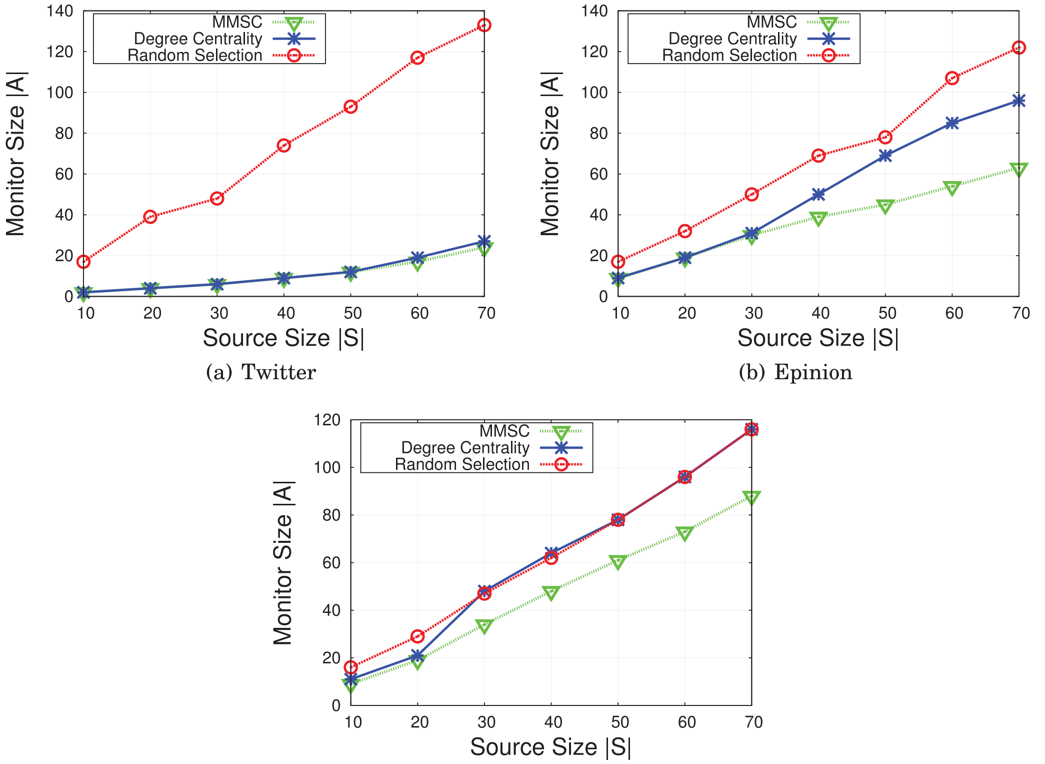
Choice of Sources and Central Node: In practical life, the users who generate misinformation (i.e., sources) tend to have a small number of neighbors (low out-degree), but are typically located in the vicinity of high-profile users, which means they are adjacent to (friend/follower of) high-degree nodes [Nguyen et al. 2012]. We have taken this important observation into consideration while selecting the source nodes. On the other hand our aim is to monitor those from being influenced by misinformation who often do not have many neighbors and, hence, cannot verify the authenticity of any wrong news/information through online friends. As a result, these users are more vulnerable to be influenced by misinformation. Accordingly, we choose the target nodes in our experiments who have low in-degree.

We constructed a set consisting of only the low out-degree neighbors of top $|S|$ high-degree nodes in the network. Source nodes were chosen randomly from this set. We used the average degree of nodes in a network as cut-off to decide whether a degree is low or not. As for the central nodes, they were chosen randomly from the set of low in-degree nodes.

Impact of Sources: Figure 3 shows the performance of MMSC, degree-centrality-based monitor selection and random-monitor selection when $\delta = 1$, i.e., the candidate-monitor set constitutes all the direct neighbors of the sources. Usually, misinformation is initiated by a small number of people [Seo et al. 2012], so we start with $|S| = 10$ and keep on increasing the source count by 10 till $|S| = 70$. In all of the sub-figures $\tau = 0.10$. Clearly, MMSC beats all other methods in terms of minimizing the number of monitors for making the mis-detection probability less than τ. As the number of sources generating misinformation increases, the gap between MMSC and other methods keep on getting larger. This implies that when there are more sources of misinformation, our proposed approach can detect misinformation more efficiently, as can be seen from Figures 3(a), 3(b), and 3(c).

From Figure 4(a) we can see that MMSC outperforms the other two methods in minimizing the monitor size when more and more source nodes of misinformation are being considered in the Twitter network. The same goes true for the Epinion and Slashdot networks as evident from Figures 4(b) and 4(c).

Impact of δ: In modern day OSNs, people are interconnected more than ever before, resulting in a high average degree. Increasingly, these networks are getting denser, and as a consequence, network diameter is reducing gradually. Information mostly propagates within 2 to 5 hops [Cha et al. 2009] nowadays, and we want to focus on placing monitors within a small distance from the sources so that misinformation can be detected at an early stage. Subsequently, we observe the behavior of different methods in reducing mis-detection probability below $\tau = 0.10$ by finding the necessary monitors in different networks when $\delta = 2$, in addition to $\delta = 1$. Figure 4 depicts



(a) Twitter

(b) Epinion

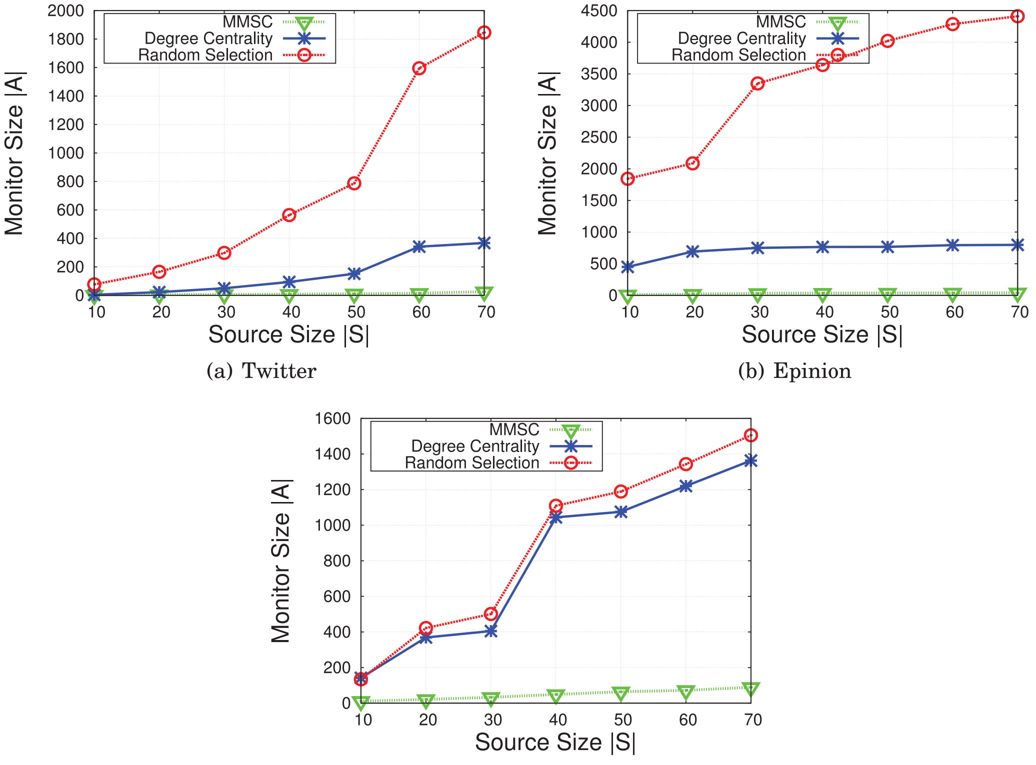
(c) Slashdot

Fig. 3. Monitor size with a different number of source nodes, $|S|$, when $\delta = 1$.

the performance of the methods on different datasets when $\delta = 2$. In this case, the candidate monitor set consists of all the direct neighbors of the sources and their direct neighbors, opening up the possibility of exploring more options. We consider the same sources as in the case of $\delta = 1$ (Figure 3).

As we take into account larger δ values, no significant increase in monitors are observed for MMSC as Figures 3(a) and 4(a) show. This is intuitive in the sense that unless we encounter any bottleneck within δ hops of the sources, the monitor size will not be changed dramatically. The same observation can be made for the Epinion network, where for $\delta = 1$, degree-based centrality was performing well compared to MMSC till $|S| = 30$, whereas for $\delta = 2$, MMSC beats the former one, even when the source node count is 10. Figures 3(b) and 4(b) justify these observations.

We can draw similar conclusions for the Slashdot dataset. As $|S|$ increases, MMSC outperforms both the degree-based heuristic and the random selection of monitors, as observed in both Figures 3(c) and 4(c). Similar to the observation found in the case of $\delta = 1$, although degree-based centrality measure competes with MMSC when $|S| \leq 20$ for smaller δ—it is clearly beaten by the latter when δ is increased. One interesting observation from both of these figures that draws our attention is the performance of degree-based centrality, which is very much similar to the random selection when $\delta = 1$, as we keep increasing the source counts. This indicates that, when it comes to identifying misinformation and thus stopping it from being disseminated throughout the network, degree-centrality-based heuristic's performance is no better than the



(a) Twitter

(b) Epinion

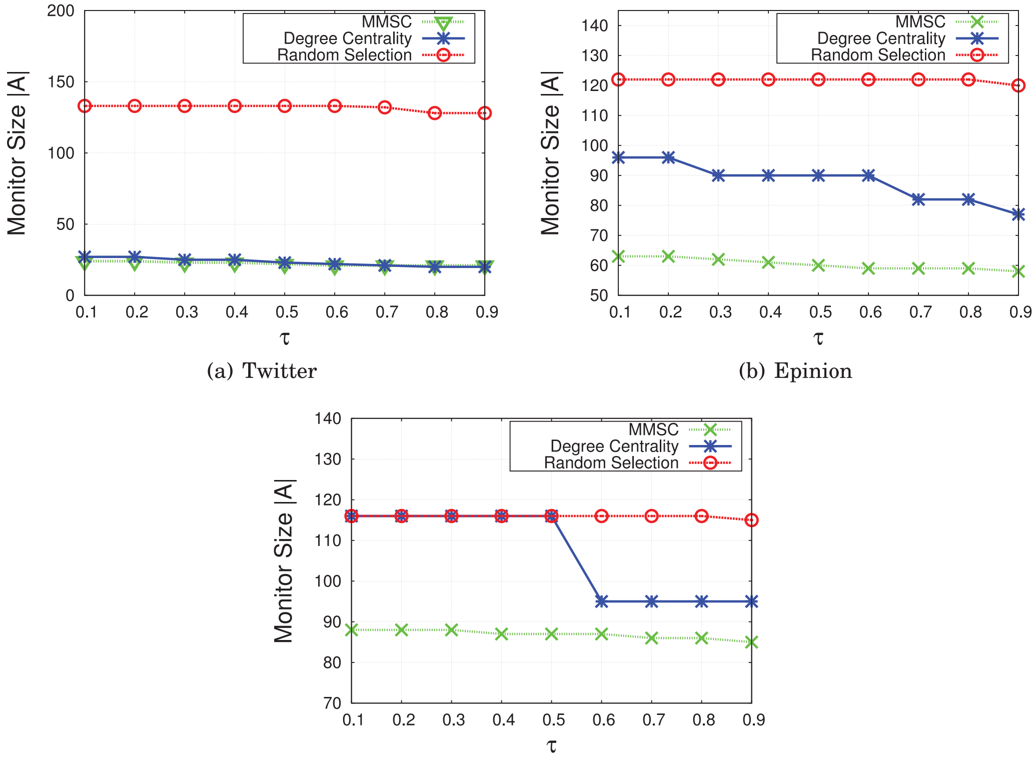
(c) Slashdot

Fig. 4. Monitor size with different numbers of source nodes, $|S|$, when $\delta = 2$.

random selection. This renders the centrality-based measure practically inapplicable in large social networks for misinformation detection.

Impact of τ: We also explore the behavior of different methods when the mis-detection threshold τ is varied from 0.10 to 0.90 for different δ values. The results are shown in Figure 5 for $\delta = 1$ (the result for other δ values are consistent with that of $\delta = 1$). In all the sub-figures, we consider $|S| = 70$; however, the result is quite similar for other $|S|$ values. For all of the three networks, more and more monitors are required when τ continues decreasing. This is coherent with the fact that the larger the τ, the lesser the number of monitors are required to ensure that the mis-detection probability is below the higher threshold τ. More monitors are needed in place to make sure that the mis-detection probability is kept below the decreasing value of τ.

From Figure 5(a) we can see that for Twitter data, MMSC requires fewer monitors compared to the other two methods when τ is varied from 0.10 to 0.90. This is, again, in conformity with the earlier results of Figures 3 and 4 and supports the superiority of the proposed Algorithm 1. We can draw the same conclusion for the other two Datasets of Epinion and Slashdot from Figures 5(b) and 5(c) respectively, regarding the better performance of MMSC. And as the τ increases from 0.10 to 0.90, each of the methods require fewer monitors to detect and prevent the misinformation from reaching the target. This is due to the relaxed requirement of misinformation-mis-detection probability which must be lower than τ and as described already, this relaxed requirement comes as a result of the increased value of τ.



(a) Twitter

(b) Epinion

(c) Slashdot

Fig. 5. Monitor size with different τ, when $\delta = 1$.

7.2. Multiple Vulnerable Central Nodes

In this section, we show the performance of Algorithm 2 (MSMN) on three datasets, which we also used for showing the superiority of Algorithm 1 (MMSC). However, our goal is to show the impact of multiple, vulnerable central nodes on monitor selection. To this end, we perform experiments on real-world datasets to find out the number of monitors required to track the set of vulnerable central nodes. We compare the result of MSMN with that of the high-degree-based centrality method and randomly chosen method.

In order to explore the implication of multiple central nodes, we choose central nodes by setting $|R| = 1\%$ to $|R| = 5\%$ of total nodes in the network, according to two different criteria, which we will explain shortly. As the choice of central node influences the monitor selection, at the same time, we also anticipate the mis-detection threshold $\tau_i, \forall i \in 1, \ldots, |R|$ for each central node. Accordingly, to evaluate their impact on monitor placement, we assign (1) the same threshold (0.50) to all of the central nodes and (2) different thresholds chosen uniformly between [0,1] from random distribution.

Since, the aim is to observe the role of different central nodes in altering the monitor placement requirement, we fix the δ to 1 and source size $|S|$ to 10. This, however, in any way, does not limit the findings we extract from the experimental results that we report. For larger values of S, the results show similar trends to that of what we present in this article, except for the fact that they require a greater number of monitors to be chosen from $N_\delta(S)$. This observation is intuitive and goes along with the behavior when $|R| = 1$, as already shown in Figures 3 and 4.

Criteria for source-central node selection: In real life, the users who generate misinformation (i.e., sources) tend to have a small number of neighbors (low out-degree) but are typically located in the vicinity of high-profile users which means they are adjacent to (friend/follower of) high-degree nodes. Users who do not have many neighbors cannot often verify the authenticity of any wrong news/information through online friends. As a result, these users are more vulnerable to be influenced by misinformation. On the contrary, misinformation propagators sometimes want to target high-degree nodes so that these high-degree nodes, once influenced through misinformation, can act as the source of misinformation and further disseminate misinformation.

To take such distinct scenarios into account, we perform experiments on two different combinations of source-central node selection, which we name as Combination 1 and Combination 2.

Combination 1: Sources are chosen randomly from the neighbors of high-degree nodes, and central nodes are chosen to have low in-degree.

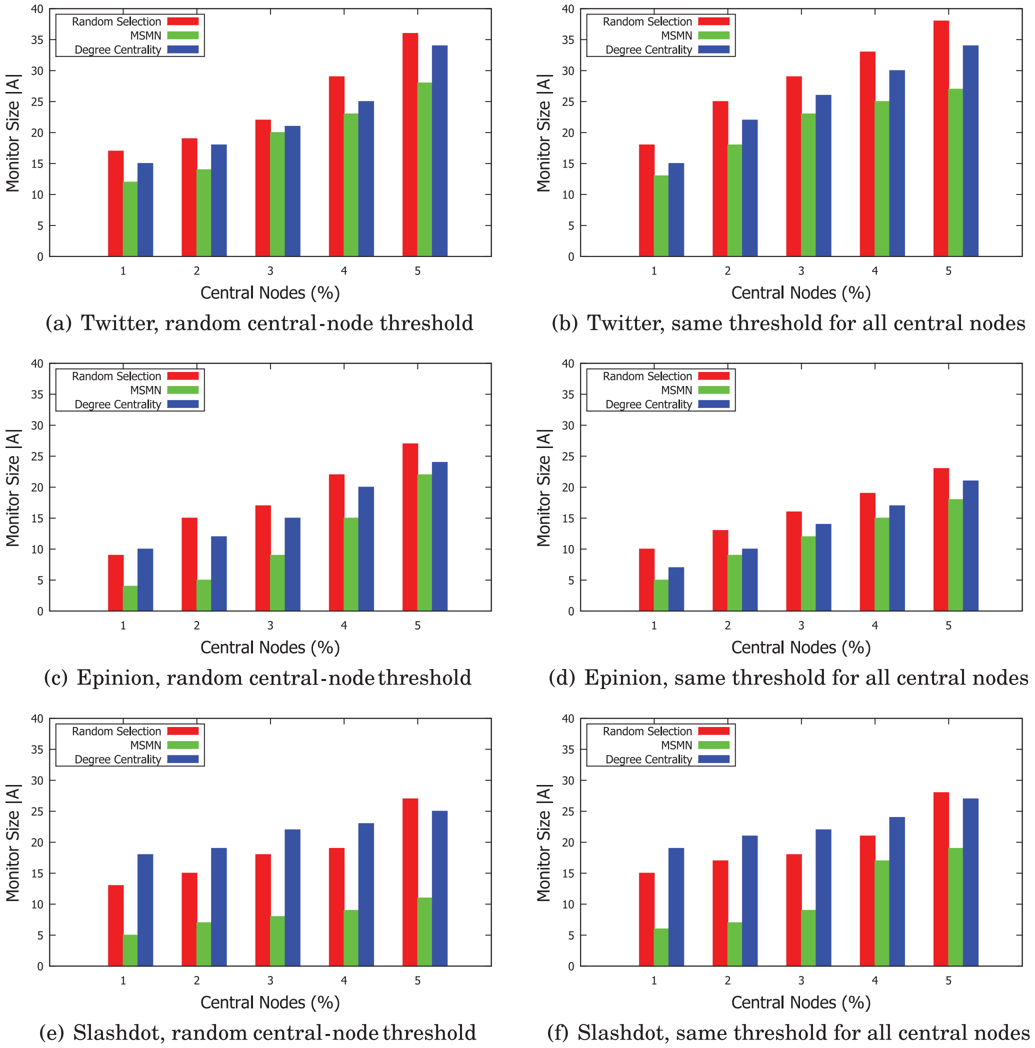
Combination 2: Both sources and the central nodes are chosen randomly from among all of the nodes in the network.

In all of the subsequent experimental results, we present the monitor count $|A|$ required for handling the misinformation for $|R| = 1\%, 2\%, 3\%, 4\%$ and 5% of the total nodes. In the reachability estimation, for each data set, we choose $\beta = 1$ and $\epsilon = 0.05$ and calculate the required number of simulations.

Performance of MSMN: From both Figures 6 and 7, we can see that MSMN requires a much lower number of monitors, compared to high-degree and random-selection methods, to handle the same number of central nodes. For Twitter data, as the number of central nodes increases, the number of monitors required to stop the misinformation for all of the methods increases, as can be seen from Figures 6(a) and 6(b). However, MSMN requires a comparatively small number of monitors to be placed for protecting the central nodes. Only 12 monitors are needed for MSMN to stop misinformation reaching the central nodes when they are 1% of the total nodes, whereas high-degree and random-selection methods require comparatively more monitors, as shown in Figure 6(a). The gap between MSMN and other methods increases even further as more central nodes are required to be monitored. This clearly demonstrates the efficiency of MSMN.

As for the difference between 6(a) and 6(b), we observe that when all the central nodes have a similar threshold, a little more monitors are required for all the methods, compared to the scenario when central nodes have different thresholds. The monitor requirement for a central node count of $|R| = 4\%$ is 23(25), 25(30) and 29(33) for MSMN, high degree, and random selection, respectively, for random thresholds (same threshold). This is intuitive and interesting at the same time; in particular, when one wants to save a single central node with a small threshold, one will need more monitors to be added in order to monitor the small threshold, compared to the one that has a larger misinformation threshold. This same trend is observed for the other two datasets of Epinion and Slashdot, as can be seen between Figures 6(c)–6(d) and 6(e)–6(f), respectively.

Impact of source-central node combination: From Figure 7, which corresponds to Combination 2, we observe that each of the methods in all the datasets require very large numbers of monitors, compared to Figure 6, which corresponds to source-central node selection, Combination 1. This implies, as the source and central nodes are chosen randomly, it takes more $N_\delta(S)$ nodes to be placed around the sources to stop misinformation from propagating toward the central nodes. Likewise, as with Combination 1, as the number of central nodes increases, more monitors are needed in



(a) Twitter, random central-node threshold

(b) Twitter, same threshold for all central nodes

(c) Epinion, random central-node threshold

(d) Epinion, same threshold for all central nodes

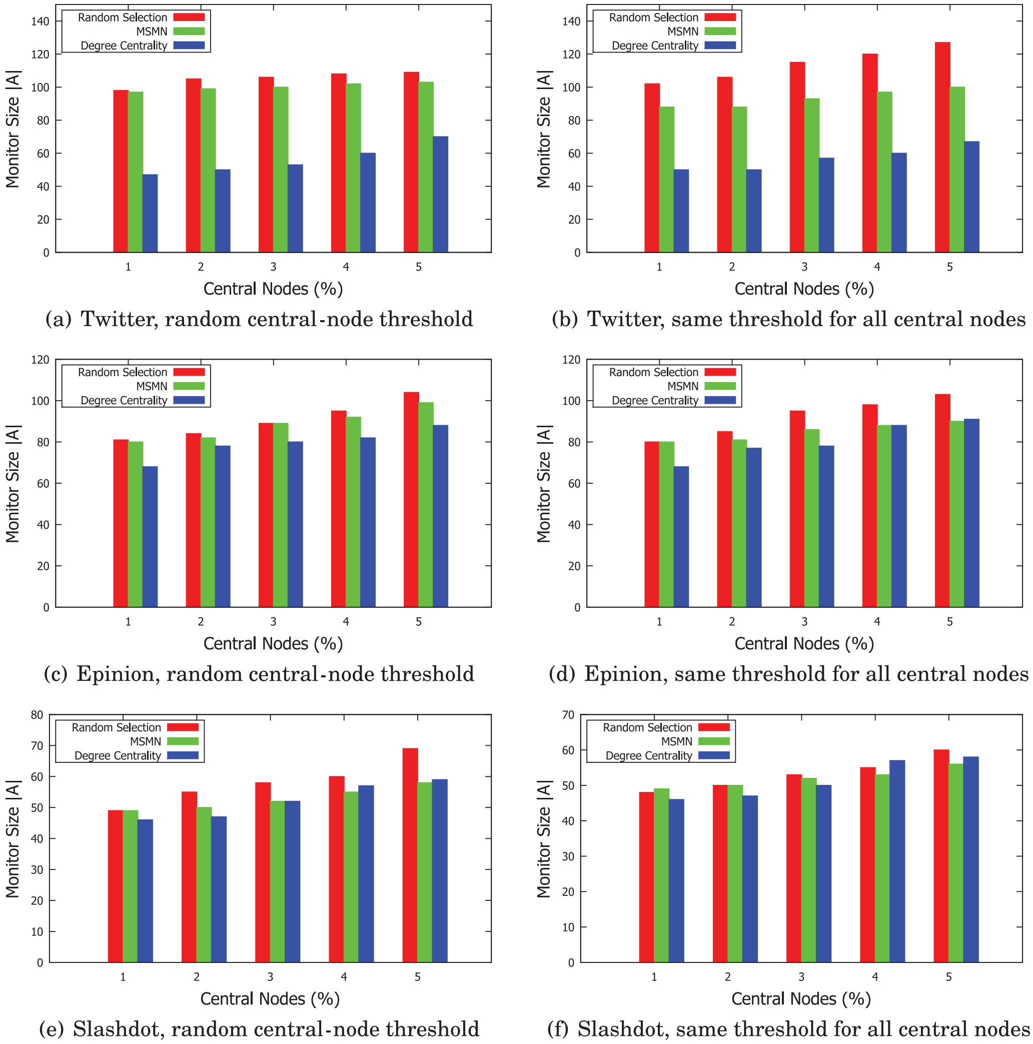
(e) Slashdot, random central-node threshold

(f) Slashdot, same threshold for all central nodes

Fig. 6. Monitor size for multiple, vulnerable central nodes for source-central node Combination 1. Left column for different random thresholds for each of the central nodes; right column for same threshold for all central nodes.

Combination 2 for all of the methods. As we can see from Figure 6(c) and Figure 7(c), for the same number of central nodes, MSMN requires more monitors for Combination 2 as compared to Combination 1. For instance, when the central-node count is 5% of the total nodes, for the latter combination, MSMN needs 99 monitors; whereas for Combination 1, only 22 monitors are needed in Epinion when the thresholds are assigned randomly. For the same dataset, when the thresholds are all equal for the central nodes (like the case where thresholds are chosen randomly), Combination 1 requires less monitors compared to Combination 2, as can be seen from Figures 6(d) and 7(d). We observe similar results for the Twitter and Slashdot datasets, as well.

However, unlike Combination 1, the gap between MSMN and other methods (high degree and random selection) is reduced as central node grows, as can be seen from the results of Epinion and Slashdot presented in Figures 7(d) and 7(f), respectively.



(a) Twitter, random central-node threshold

(b) Twitter, same threshold for all central nodes

(c) Epinion, random central-node threshold

(d) Epinion, same threshold for all central nodes

(e) Slashdot, random central-node threshold

(f) Slashdot, same threshold for all central nodes

Fig. 7. Monitor size for multiple, vulnerable central nodes for source-central node Combination 2. Left column for different random thresholds for each of the central nodes; right column for same threshold for all central nodes.

The behaviors of datasets are different in Combination 2 compared to Combination 1 when it comes to central nodes' threshold assignment. More monitors are required when thresholds are chosen to be equal compared to the case when thresholds are chosen to be random for each of the central nodes. For instance, in Slashdot, less monitors are required when the thresholds are fixed compared to when the thresholds are assigned randomly. From Figure 7(e), we can find that while MSMN (high degree, random selection) needs 58 (59,69) monitors to protect $|R| = 5\%$ central nodes in the scenario when all of these central nodes have different thresholds, the same algorithms take 56 (58,60) monitors to protect them when each of them have the same thresholds for Slashdot, as can be seen in Figure 7(f). We also observe from Figures 7(a)–7(b) and Figures 7(c)–7(d), that Twitter and Epinion, respectively, also show similar behavior like Slashdot in terms of monitor requirement for same and different central-node thresholds. Central nodes were assigned thresholds separately and independently in Combinations 1 and 2, which might have resulted in the same node getting different threshold in those combinations. As a result, we observed different behavior for the algorithms in terms of monitor numbers for the two combinations. One observation worth noting in this regard is for the Twitter dataset, high-degree centrality performs better compared to other methods in terms of minimizing the monitor sets. As the average degree of nodes in Twitter is very high compared to other networks, it eventually makes the high degree perform different than other networks, and for higher values of R, it outperforms other methods.

In short, MSMN is able to efficiently find the minimum number of monitors required for protecting a variety of central nodes for different source-/central-node combinations in all of the datasets. Not only did it successfully outperform other methods, but it also showed some interesting insights, especially when the central nodes had disparate kinds of threshold distribution, which can be leveraged for devising a better protection mechanism that would go a long way in securing the social networks from misinformation.

8. RELATED WORK

In this section, we briefly review some important related work on diffusion modeling, influence maximization, misinformation detection, and monitor placement.

OSNs play an important role in interactions among people and information dissemination. A lot of efforts have been made in the research of the diffusion and cascading of information [Guille et al. 2013; Cheng et al. 2013]. Besides the detection of interesting topics, modeling the diffusion process is an important branch in the information diffusion research. Two well-known, discrete, time-diffusion models have been proposed: linear threshold (LT) model and independent cascading (IC) model. In LT models, all active neighbors of a node contribute certain weights, and when their sum is greater than a given threshold, this node becomes active [Kempe et al. 2003]. In IC models, each active node is given a single chance to activate a neighbor with some probability. To model multiple cascades, a generalized LT model is proposed [Pathak et al. 2010b]. Continuous time-diffusion models also can be seen in literature [Rodriguez and Schölkopf 2012].

Appropriate information-/influence-diffusion models can help to provide important insights into the design of social platforms and applications. One major application is viral marketing [Domingos and Richardson 2001], e.g., a company wants to promote a new product or technological innovation through word-of-mouth effects in OSNs. As an algorithmic technique for viral marketing, influence maximization was first defined by Kempe et al. [2003], which asks for a small set of seeding nodes in an OSN that maximizes the spread of influence under certain influence-diffusion models. Influence maximization is formulated as a discrete stochastic optimization problem, which has been proved to be NP-hard. In greedily selecting new seeds, a “lazy-forwar” optimization to significantly reduce the number of influence spread evaluations [Leskovec et al. 2007]. Based on the linear time computation of influence in directed acyclic graphs, Chen et al. proposed the scalable influence maximization algorithm for the LT model [Chen et al. 2010].

As OSNs become a fertile land for innovation and product promotion, some people find ways to abuse it. One major undesirable type of abuse is the dissemination of misinformation. Starting with a small set of users, misinformation may spread through the network from person to person, in a virus manner. Political astroturf can be observed on Twitter [Ratkiewicz et al. 2011]. Kwon et al. [2013] studied the rumor-spreading pattern on Twitter and the classification of rumors based on temporal, structural, and linguistic features. A spam classifier is developed to detect collective-attention spam, such as YouTube breaking videos, Twitter trending topics, and popular Facebook profiles [Lee et al. 2012]. Users assess the credibility of information based on trust relationship, and the propagation or trust and distrust is also of research interests [Guha et al. 2004].

Destructive effects of the misinformation diffusion motivate the design of strategies to detect or control it [Fan et al. 2013]. By placing monitors, misinformation spreading on OSNs can be detected. Monitor placement has been investigated in different domains [Krause and Guestrin 2011; Leskovec et al. 2007]. The effectiveness of monitoring friends of selected individuals in the detection of contagious outbreaks is shown in Christakis and Fowler [2010]. Leskovec et al. [2007] studied the information-cascade detection problem in the blogosphere and proposed a near-optimal sensor-placement solution. There are some literatures focusing on limiting the spread of misinformation. Budak et al. [2011] modeled the propagation of good information and misinformation as competing campaigns and proposed an algorithm to select seeds of good information such that the number of people that adopt the misinformation is minimized. A rumor combat strategy based on social trust is proposed [Tripathy et al. 2013], in which anti-rumor messages are broadcast. Nguyen et al. [2012] formulated a β_T^I-Node protectors problem to find the smallest set of influential nodes whose decontamination with good information helps to contain misinformation. However, to apply control strategies, they should know the content of misinformation and disseminate corresponding "good information."

Instead of developing techniques to identify misinformation or design control strategies, in this article, we aim to bridge these two areas—that is, design monitor-placement strategies in misinformation detection, which provides support in misinformation containment. None of the existing work deals with the fundamental problem of restricting the propagation distance of misinformation before detection, nor do they emphasize the importance of monitoring central nodes. To fill this gap, we propose efficient solutions for timely detection of misinformation, i.e., limit the distance between source nodes, and monitors. Also, we emphasize the importance of "central nodes" by restricting the detection probability.

9. CONCLUSION

OSNs provide a content-rich platform for people's communication and the dissemination of real-time information. However, the threat of misinformation propagation comes alongside these promising features. The *misinformation propagation* is the spread of false or inaccurate information, which can lead to undesirable and even devastating effects. Consequently, timely detection of misinformation is of practical significance in order to avoid these effects.

In this article, we focused on the monitor placement for misinformation detection in OSNs. Two scenarios have been considered and solutions are proposed accordingly. Specifically, when the knowledge about misinformation sources is unavailable, we can select monitors by a simple, greedy strategy and obtain an approximation ratio of $1 - 1/e$. We define a τ-MP problem for the scenario where we have the knowledge about the potential misinformation sources, and propose an algorithm based on cut-set_2. Furthermore, an algorithm using sampling techniques is proposed to address the general case where there are multiple vulnerable nodes we want to protect. Extensive experiments are conducted to demonstrate the superiority of the proposed solutions.

There remains some open issues which are not addressed in this article. First, in experimental evaluation, the edge transmission probability is assigned uniformly at random. The impact of different distributions of transmission probability on monitor placement is an interesting future research direction. Second, it is intriguing to evaluate the performance of proposed algorithms for detecting the misinformation following LT-like behavior or other diffusion patterns. Moreover, if monitors are allowed to be placed everywhere in OSNs, it is worth seeing the trade-off between the required monitor set size and the number of users being activated.

REFERENCES

Ceren Budak, Divyakant Agrawal, and Amr El Abbadi. 2011. Limiting the spread of misinformation in social networks. In *WWW*. 665–674.

Salvatore A. Catanese, Pasquale De Meo, Emilio Ferrara, Giacomo Fiumara, and Alessandro Provetti. 2011. Crawling Facebook for social network analysis purposes. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*. ACM, 52.

Meeyoung Cha, Hamed Haddadi, Fabricio Benevenuto, and P. Krishna Gummadi. 2010. Measuring user influence in Twitter: The million follower fallacy. *ICWSM* 10, (2010), 10–17.

Meeyoung Cha, Alan Mislove, and Krishna P. Gummadi. 2009. A measurement-driven analysis of information propagation in the Flickr social network. In *WWW*. 721–730.

Wei Chen, Yifei Yuan, and Li Zhang. 2010. Scalable influence maximization in social networks under the linear threshold model. In *ICDM*. 88–97.

Shin-Ming Cheng, Vasileios Karyotis, Pin-Yu Chen, Kwang-Cheng Chen, and Symeon Papavassiliou. 2013. Diffusion models for information dissemination dynamics in wireless complex communication networks. *Journal of Complex Systems* 2013 (2013), 972352.

Nicholas A. Christakis and James H. Fowler. 2010. Social network sensors for early detection of contagious outbreaks. *PloS one* 5, 9 (2010), e12948.

Pedro Domingos and Matt Richardson. 2001. Mining the network value of customers. In *KDD*. 57–66.

Lidan Fan, Zaixin Lu, Weili Wu, Bhavani Thuraisingham, Huan Ma, and Yuanjun Bi. 2013. Least cost rumor blocking in social networks. In *Proceedings of the 2013 IEEE 33rd International Conference on Distributed Computiing Systems (ICDCS)*. IEEE, 540–549.

Amit Goyal, Francesco Bonchi, and Laks V. S. Lakshmanan. 2010. Learning influence probabilities in social networks. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining*. ACM, 241–250.

Ramanthan Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. 2004. Propagation of trust and distrust. In *Proceedings of the 13th International Conference on World Wide Web*. ACM, 403–412.

Adrien Guille, Hakim Hacid, Cécile Favre, and Djamel A. Zighed. 2013. Information diffusion in online social networks: A survey. *ACM SIGMOD Record* 42, 2 (2013), 17–28.

http://snap.stanford.edu/data. Stanford large network dataset collection.

Amanda Lee Hughes and Leysia Palen. 2009. Twitter adoption and use in mass convergence and emergency events. *International Journal of Emergency Management* 6, 3 (2009), 248–260.

Jing Jiang, Christo Wilson, Xiao Wang, Wenpeng Sha, Peng Huang, Yafei Dai, and Ben Y Zhao. 2013. Understanding latent interactions in online social networks. *ACM Transactions on the Web (TWEB)* 7, 4 (2013), 18.

Fang Jin, Edward Dougherty, Parang Saraf, Yang Cao, and Naren Ramakrishnan. 2013. Epidemiological modeling of news and rumors on Twitter. In *SNAKDD*. 8.

D. Kempe, J. M. Kleinberg, and V. Tardos. 2003. Maximizing the spread of influence through a social network. In *KDD*. 137–146.

Andreas Krause and Carlos Guestrin. 2011. Submodularity and its applications in optimized information gathering. *ACM Transactions on Intelligent Systems and Technology (TIST)* 2, 4 (2011), 32.

Sejeong Kwon, Meeyoung Cha, Kyomin Jung, Wei Chen, and Yajun Wang. 2013. Prominent features of rumor propagation in online social media. In *ICDM*. 1103–1108.

Kyumin Lee, James Caverlee, Krishna Y. Kamath, and Zhiyuan Cheng. 2012. Detecting collective attention spam. In *Proceedings of the 2nd Joint WICOW/AIRWeb Workshop on Web Quality*. ACM, 48–55.

Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. 2007. Cost-effective outbreak detection in networks. In *KDD*. 420–429.

Stephan Lewandowsky, Ullrich K. H. Ecker, Colleen M. Seifert, Norbert Schwarz, and John Cook. 2012. Misinformation and its correction: Continued influence and successful debiasing. *Psychological Science in the Public Interest* 13, 3 (2012), 106–131.

Shuyang Lin, Fengjiao Wang, Qingbo Hu, and Philip S. Yu. 2013. Extracting social events for learning better information diffusion models. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 365–373.

Wuqiong Luo, Wee Peng Tay, and Mei Leng. 2013. Identifying infection sources and regions in large networks. *IEEE Transactions on Signal Processing* 61, 11 (2013), 2850–2865.

Marcelo Mendoza, Barbara Poblete, and Carlos Castillo. 2010. Twitter under crisis: Can we trust what we RT? In *SOMA*. 71–79.

Pasquale de Meo, Emilio Ferrara, Fabian Abel, Lora Aroyo, and Geert-Jan Houben. 2013. Analyzing user behavior across social sharing environments. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5, 1 (2013), 14.

Nam P. Nguyen, Guanhua Yan, My T. Thai, and Stephan Eidenbenz. 2012. Containment of misinformation spread in online social networks. In *WebSci*. 213–222.

N. Pathak, A. Banerjee, and J. Srivastava. 2010a. A generalized linear threshold model for multiple cascades. In *ICDM*. 965–970.

Nishith Pathak, Arindam Banerjee, and Jaideep Srivastava. 2010b. A generalized linear threshold model for multiple cascades. In *Proceedings of the 2010 IEEE 10th International Conference on Data Mining (ICDM)*. IEEE, 965–970.

B. Aditya Prakash, Jilles Vreeken, and Christos Faloutsos. 2012. Spotting culprits in epidemics: How many and which ones? In *ICDM*, Vol. 12. 11–20.

Vahed Qazvinian, Emily Rosengren, Dragomir R. Radev, and Qiaozhu Mei. 2011. Rumor has it: Identifying misinformation in microblogs. In *EMNLP*. 1589–1599.

Jacob Ratkiewicz, Michael Conover, Mark Meiss, Bruno Gonçalves, Alessandro Flammini, and Filippo Menczer. 2011. Detecting and tracking political abuse in social media. In *ICWSM*.

Manuel Gomez Rodriguez and Bernhard Schölkopf. 2012. Influence maximization in continuous time diffusion networks. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*. 313–320.

Eunsoo Seo, Prasant Mohapatra, and Tarek Abdelzaher. 2012. Identifying rumors and their sources in social networks. In *SPIE Defense, Security, and Sensing*. International Society for Optics and Photonics, 83891I–83891I.

Devavrat Shah and Tauhid Zaman. 2011. Rumors in a network: Who's the culprit? *Information Theory, IEEE Transactions on* 57, 8 (2011), 5163–5181.

Rudra M. Tripathy, Amitabha Bagchi, and Sameep Mehta. 2010. A study of rumor control strategies on social networks. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. ACM, 1817–1820.

Rudra M. Tripathy, Amitabha Bagchi, and Sameep Mehta. 2013. Towards combating rumors in social networks: Models and metrics. *Intelligent Data Analysis* 17, 1 (2013), 149–175.

Leslie G. Valiant. 1979. The complexity of enumeration and reliability problems. *SIAM J. Comput.* 8, 3 (1979), 410–421.

Gadi Wolfsfeld, Elad Segev, and Tamir Sheafer. 2013. Social media and the Arab Spring: Politics comes first. *The International Journal of Press/Politics* 18, 2 (2013), 115–137.

Maria S. Zaragoza, R. S. Belli, and Kristie E. Payment. 2006. Misinformation effects and the suggestibility of eyewitness memory. *Do Justice and Let the Sky Fall: Elizabeth F. Loftus and her Contributions to Science, Law, and Academic Freedom* (2006), 35–63.

Zhao Zhang, Wen Xu, Weili Wu, and Ding-Zhu Du. 2015. A novel approach for detecting multiple rumor sources in networks with partial observations. *Journal of Combinatorial Optimization* (2015), 1–15.

Kai Zhu and Lei Ying. 2014. A robust information source estimator with sparse observations. *Computational Social Networks* 1, 1 (2014), 1–21.

Received April 2015; revised December 2015; accepted January 2016