Identification of the Source(s) of Misinformation Propagation Utilizing Identifying Codes

Kaustav Basu
NetXT Lab, SCIDSE, Arizona State University
kaustav.basu@asu.edu

ABSTRACT

With billions of users, social networks have become the go to platform for information diffusion for news media outlets. Lately, certain entities (users and/or organizations) have been active in generating misinformation in order to attract users to their respective websites, to generate online advertisement revenues, to increase followers, to create political instability, etc. With the increasing presence of misinformation on social networks, it is becoming increasingly difficult to not only distinguish between information and misinformation, but also, to identify the source(s) of misinformation propagation. This effort reviews my doctoral research on identifying the source(s) of misinformation propagation. Particularly, I utilize the mathematical concept of Identifying Codes to *uniquely* identify users who become active in propagating misinformation. In this paper, I formally present the computation of the Minimum Identifying Code Set (MICS) as a novel variation of the traditional Graph Coloring problem. Furthermore, I present an Integer Linear Program for the computation of the MICS. I apply the technique on various anonymous Facebook network datasets and show the effectiveness of the approach.

KEYWORDS

Identifying Codes; Misinformation Propagation; Unique Identification; Graph Analytics

ACM Reference Format:
Kaustav Basu. 2019. Identification of the Source(s) of Misinformation Propagation Utilizing Identifying Codes. In *Companion Proceedings of the 2019 World Wide Web Conference (WWW '19 Companion), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3308560.3314200

1 PROBLEM

The emergence of social networking has revolutionized the communication domain. It has successfully overcome the geographical barrier and has connected users with their friends and family members, located on other the side of the planet. With the advantages of a social network being in abundance, it also suffers from a few major drawbacks. In recent times, the inability of a social networking company to maintain the privacy of a user has become front page news. Moreover, adversaries are trying to utilize the connectivity and reach of a social network, to spread misinformation. The identification of the source of misinformation propagation is a chief problem in today's social network analysis. Hence, as a part of this research effort, I address the problem of *unique* identification of the source(s) of misinformation propagation, by utilizing the mathematical concept of *Identifying Codes*.

With the evolution of networks over the past decade, social networks have been utilized by news outlets and media houses for the propagation of a respective outlet's/house's agenda. The greater the utilization, the greater the reliance of the user on the social media content. Certain entities (individuals or organizations) on social networking websites have taken advantage of this reliance, to propagate misinformation. Agencies (or monitors) such as Politifact and Media Bias/Fact Check (MBFC), are trusted and verified agencies who can determine if a news article is informed or misinformed. Even with the existence of such monitors, it is becoming increasingly difficult to distinguish between informed and misinformed articles, due to the sheer volume of such articles on social networking websites. Moreover, such fact checking agencies do not verify news articles in real time. As a result, various research groups have undertaken the study of misinformation propagation to primarily (i) develop algorithms to detect whether a news article is informed or misinformed, and (ii) identify the source(s) of misinformation propagation.

Mathematically, a social network can be expressed as a graph $G = (V, E)$, which consists of a set of vertices V and a set of edges E, where two vertices $u, v \in V$ are said to be connected if there is an edge $e \in E$ between them. The set of vertices can be thought of as entities, such as individuals, organizations, locations, etc. and the set of edges can be thought of relationships between these entities. For example, in the case of a social network, users of the social network form the set of vertices, and friendship between two users form the set of edges.

As a part of my doctoral research, I am currently studying the accurate identification of source(s) of misinformation propagation. I am in the process of utilizing the mathematical concept of Identifying Codes in order to accurately identify users/organizations on the social network, by placing the minimum number of such monitors or sensors (fact checking agencies such as Politifact or MBFC) on a social network. It may be noted that in this effort, I not only study the propagation of misinformation in the network, but also present a novel approach which can accurately identify the source(s) of misinformation propagation.

One of the most frequently studied problems in this context is the Sensor Placement Optimization problem. It may be noted that a sensor in this context refers to the monitoring agencies, such as Politifact and MBFC. The goal of the sensor optimization problem is to find the smallest set of nodes on which sensors must be

WWW '19 Companion, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6675-5/19/05.
https://doi.org/10.1145/3308560.3314200

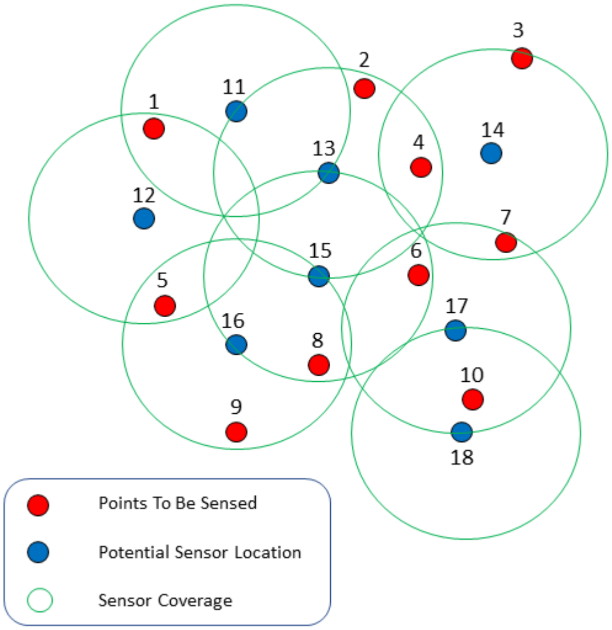

Figure 1: Potential Sensors and Sensing Locations

placed, so that all the nodes in the network can be sensed. In other words, every node should be under the coverage area of at least one deployed sensor. In the context of misinformation propagation in a social network, find the smallest set of users who should be monitored by the fact checking entities entities, so that all the users in the social network are under the coverage of at least one monitored user.

Although a number of studies on sensor placement optimization problem follow the set cover formulation to find a solution, the set cover based approach has a serious limitation on the accurate identification of the node, where some abnormality is detected by one or more of the deployed sensors. We illustrate this point with the help of an example. In Fig. 1, the ten red points (numbered from 1-10) indicate the points to be sensed (monitored), the eight blue points (numbered from 11-18) indicate the potential locations where the sensors can be deployed and the green circles (centered on each blue point) indicate the coverage area of a sensor deployed at that blue point. We can easily construct a bipartite graph $G = (V_1 \cup V_2, E$, from the set of points in Fig. 1. The set V_1 could denote the set of red points and the set V_2 could denote the set of blue points. If a red point is within the sensing range of a blue point, then we have an edge between the two points. The resulting graph from Fig. 1 is illustrated in Fig. 2. In this example, it can be verified that if the set cover approach was followed, and sensors were deployed in locations 11, 13, 14, 16 and 17, then all points from 1 to 10 will be within the sensing range of at least one sensor. Specifically, the points (1-10) covered (sensed) by the sensors 11, 13, 14, 16, 17 are shown in Table 1. In Table 2, we present the sensors that are actually sensing the points 1-10, using the Set Cover approach.

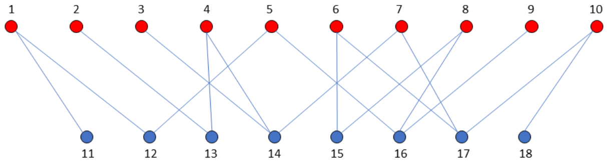

Figure 2: Network Corresponding To Fig. 1

Table 1: Points Covered By Each Sensor

Sensor Location	Points Sensed	Sensor Location	Points Sensed
11	1	15	6, 8
12	1, 5	16	5, 8, 9
13	2,4	17	6, 7, 10
14	3,4,7	18	10

Table 2: Sensors Covering Each Point

Points Sensed	Sensor Location	Points Sensed	Sensor Location
1	11	6	17*
2	13	7	14, 17
3	14	8	16*
4	13, 14	18	16*
5	16**	10	17*

Table 3: Sensors Covering Each Point

Points Sensed	Sensor Location	Points Sensed	Sensor Location
1	12	6	15, 17
2	13	7	14, 17
3	14	8	15, 16
4	13, 14	18	16
5	12, 16	10	17

The serious limitation of the set cover based approach to optimal sensor placement problem is that, it may fail to uniquely identify the point where an abnormality is detected by the sensor. We elaborate this point with the results shown in Tables 1 and 2. In this example, sensors were deployed at locations 11, 13, 14, 16, 17 and this deployment ensured that all points to be sensed were within the coverage area of at least one sensor. Suppose a control center has five indicator lamps A, B, C, D and E corresponding to five sensors located at 11, 13, 14, 16 and 17. If the sensor does not sense an abnormality at the point it is sensing, then the corresponding lamp is lit green. If a sensor senses an abnormality at a point, then the corresponding lamp turns red. From Table 2, it can be seen that points 6 and 10 are sensed by sensor 17 only, and points 5, 8 and 9 are sensed by sensor 16 only. The implication of this is that if lamp E (corresponding to sensor 17) turns red, then it will not be possible to ascertain if the abnormality was detected at point 6 or 10. Similarly, if lamp D (corresponding to sensor 16) turns red, then it will not be possible to ascertain if the abnormality was detected at point 5 or 8 or 9.

This limitation of failure to uniquely identify the point where abnormality is detected by the sensor, can be overcome by deployment of additional sensors. In this example, instead of deploying sensors at locations 11, 13, 14, 16 and 17, if they were deployed at locations 12, 13, 14, 15, 16 and 17, then each point would have been sensed in the way as shown in Table 3. It may be noted that deployment of six sensors, instead of five, avoids the problem of failure of unique identification of points where abnormality is detected. The mathematical foundation of computing the least number of sensors needed to uniquely identify locations (or nodes) where abnormality is detected, is called *Identifying Codes.*

2 RELATED WORK

Over the past half decade, there has been significant studies in the field of misinformation propagation. In this section, I highlight few efforts which motivated my research. Abbasi in [1], studied a method to measure user credibility in social media based on the user‘s profile. Ciampaglia in [5], showed that the complexities of human fact checking can be approximated by finding the shortest path between concept nodes under properly defined semantic proximity metrics on knowledge graphs. 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 in [6]. Shi in [13], viewed link-prediction task in a knowledge graph to accurately determine the veracity of a fact. Shu *et. al.* in [14] presented a survey of detecting fake news on social media. Real-world datasets measuring users trust level on fake news was constructed in [15]. Tachhini *et. al.* in [17] showed that Facebook posts can be classified with high accuracy as hoaxes or non-hoaxes on the basis of the users who “liked” them.

Karpovsky *et. al.* introduced the concept of Identifying Codes in [7] and provided results for Identifying Codes for graphs with specific topologies, such as binary cubes and trees. Using Identifying Codes, Laifenfeld *et. al.* studied covering problems in [8]. Charon *et. al.* studied studied complexity issues and showed that in several types of graph, the problem is NP-hard in [4]. Approximation algorithms for computation of Identifying Codes for some special types of graphs are presented in [18] and [16]. Ray *et. al.* studied location detection problem in emergency sensor networks, using Identifying Codes [10]. In this paper, they also introduced the concept of robust Identifying Codes to deal with faults in sensor networks. They presented an algorithm for generating *irreducible* Identifying Codes in polynomial time. It may be noted that irreducible Identifying Code is only a *minimal* Identifying Code and may not be the *minimum* (or optimal) Identifying Code. In contrast to the algorithm presented in [10], we present an algorithm for construction of optimal Identifying Code for the problem scenario under study. Sen in [11] introduced the novel graph coloring with seepage problem a novel variation of the computation of minimum Identifying Code. Sen in [12] and Basu in [3] analyzed the unique identification of terrorists in terrorist networks. Finally, Basu in [2] studied the monitoring of the health of critical power system equipments by utilizing Identifying Codes.

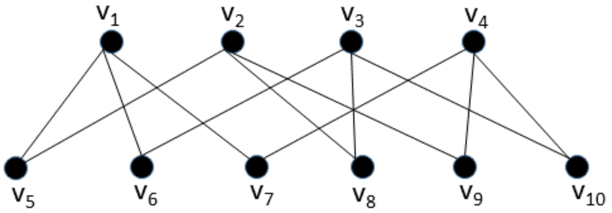

Figure 3: Graph with Identifying Code Set {v_1, v_2, v_3, v_4}

3 PROPOSED APPROACH

In this report, I use Identifying Code of the *simplest form* and define it as follows. *A vertex set V' of a graph $G = (V, E)$ is defined as the Identifying Code Set (ICS) for the vertex set V, if for all $v \in V$, $N^+(v) \cap V'$ 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 Identifying Code Set of *smallest cardinality.* The vertices of the set V' may be viewed as *alphabets* of the code, and the *string* made up with the alphabets of $N^+(v)$ may be viewed as the unique “code” for the node v. For instance, consider the graph $G = (V, E)$ shown in Fig. 3. In this graph $V' = \{v_1, v_2, v_3, v_4\}$ is an ICS as it can be seen from Table 4 that $N^+(v) \cap V'$ is *unique* for all $v_i \in V$. From the table, it can be seen that the code for node v_1 is v_1, the code for v_5 is v_1, v_2, the code for v_{10} is v_3, v_4, etc.

A necessary and sufficient condition for Identifying Codes to exist is that the underlying graph be “twin-free”. In other words, this approach fails if there are two nodes $u, v \in V$ such that $N^+(u) = N^+(v)$. In this case, nodes u, v cannot be uniquely identified. In such scenarios, one workaround is to combine nodes u, v to form a super-node.

Table 4: $N^+(v) \cap V'$ results for all $v \in V$ for the graph in Fig. 3

$N^+(v_1) \cap V' = \{v_1\}$	$N^+(v_2) \cap V' = \{v_2\}$
$N^+(v_3) \cap V' = \{v_3\}$	$N^+(v_4) \cap V' = \{v_4\}$
$N^+(v_5) \cap V' = \{v_1, v_2\}$	$N^+(v_6) \cap V' = \{v_1, v_3\}$
$N^+(v_7) \cap V' = \{v_1, v_4\}$	$N^+(v_8) \cap V' = \{v_2, v_3\}$
$N^+(v_9) \cap V' = \{v_2, v_4\}$	$N^+(v_{10}) \cap V' = \{v_3, v_4\}$

`Graph Coloring with Seepage (GCS) Problem:`

The MICS computation problem can be viewed as a novel variation of the classical Graph Coloring problem. I will refer to this version as the *Graph Coloring with Seepage (GCS)* problem. In the classical graph coloring problem, when a color is *assigned* (or injected) to a node, only that node is colored. The goal of the classical graph coloring problem is to use as few distinct colors as possible such that (i) every node receives a color, and (ii) no two adjacent nodes of the graph have the same color. In the GCS problem, when a color is assigned (or injected) to a node, not only does that node receive the color, but also the color *seeps* into all the adjoining nodes. For example, if a node v_i is adjacent to two other nodes v_j and v_k in the graph, then if the color red is injected to v_j, not only v_j will become red, but also v_i will become red as it is adjacent to v_j. Now if the color blue is injected to v_k, not only v_k will become blue, but also the color blue will seep in to v_i as it is adjacent to v_k. Since v_i was already colored red (due to seepage from v_j), after color seepage from v_k, it's color will be a *combination of red and blue (purple).* At this point, all three nodes v_j, v_k, and v_i will have "distinct" colors red, blue, and purple, respectively. *The color assigned to a node may be due to: (i) only injection at that node, (ii) only seepage from other adjoining nodes and (iii) a combination of injection and seepage.* The colors injected at the nodes are referred to as *atomic* colors. The colors formed by the combination of two or more atomic colors are referred to as *composite* colors. The colors injected at the nodes (atomic colors) are all *unique.* The goal of the GCS problem is to inject colors to as few nodes as possible, such that (i) every node receives a color, and (ii) no two nodes of the graph have the same color.

Suppose that the node set V' is an ICS of a graph $G = (V, E)$ and $|V'| = p$. In this case if p distinct colors are injected to V' (one distinct atomic color to one node of V'), then by the definition of ICS for all $v \in V$, if $N^+(v) \cap V'$ is unique, all nodes of $G = (V, E)$ will have a unique color (either atomic or composite). Thus computation of MICS is equivalent to solving the GCS problem.

4 METHODOLOGY

The objective of this study is to uniquely determine the source of misinformation propagation in a social network. With billions of users on social networking sites and only a handful of entities (monitors) to verify the veracity of online posts (such as Politifact and Media Bias/Fact Check), it is imperative to have an effective strategy in place, in order to immediately detect the source of such propagation and take swift action against them.

It is evident from the previous section that the mathematical concept of Identifying Code relates to an underlying graph. A social network can be easily represented using a graph $G = (V, E)$, where V denotes the set of users and two users are connected by an edge $e \in E$ if they are friends. The monitors (Politifact or MBFC) should be placed (on the nodes) in a manner such that, if a user (or node) becomes active in propagating misinformation, then they will be uniquely identified. This approach is based on the fact that all the immediate friends or followers of the user initiating the misinformation, will get to know about the misinformation (via Twitter or Facebook posts) . In reality, not all of these friends will participate in the propagation of the misinformation in the next time step. However, in our approach, we assume the contrary, i.e., all the friends (of the initiating user) participate in the propagation of misinformation in the next time step. Thus, the computation of minimum nodes (users) to be monitored, is equivalent to solving the MICS problem. Below, an Integer Linear Program is presented, which solves the MICS problem.

Instance: $G = (V, E)$, an undirected graph, where the node set V denotes the set of users in the social network and there is an edge $e_{ij} \in E$ if i and j are friends.

Problem: Find the smallest subset $V' \subseteq V$, such that injection of colors at these nodes, ensures that each node $v_i \in V$, receives a unique color (either atomic or composite) through seepage.

We use the notation $N(v_i)$ to denote the neighborhood of v_i, for any $v_i \in V$. Corresponding to each $v_i \in V$, we use an indicator variable x_i,

$$x_i = \begin{cases} 1, & \text{if a color is injected at node } v_i, \\ 0, & \text{otherwise} \end{cases}$$

Objective Function: *Minimize* $\sum_{v_i \in V} x_i$

Coloring Constraint: $\sum_{v_i \in N(v_j)} x_i \geq 1,$
$\forall v_j \in V$

Unique Coloring Constraint: $\sum_{v_i \in \{N(v_j) \bigoplus N(v_k)\}}$
$x_i \geq 1, \; \forall v_j \neq v_k, \in V$

$N(v_j) \bigoplus N(v_k)$ denotes the Exclusive-OR (symmetric set difference) of the node sets $N(v_j)$ and $N(v_k)$. It may be noted that the objective function ensures that the fewest number of nodes in V are assigned a color. The Coloring Constraint ensures that every node in V receives at least one color through seepage from the colors injected at nodes in V. A consequence of the Coloring Constraint is that, a node in V may receive more than one color through seepage from the colors injected at nodes in V. The Unique Coloring Constraint ensures that, for every pair of nodes (v_j, v_k) in V, at least one node in the node set $N(v_j) \bigoplus N(v_k) \subseteq V$ is injected with a color. This guarantees that v_j and v_k will not receive identical colors through the color seepage from the nodes in V.

5 RESULTS

In this section, I present a set of preliminary results obtained from the Facebook dataset available on SNAP [9]. Table 5 contains the results of the ILP presented in the previous section. In the table, the rows denote the various Facebook networks studied in this effort. All of the graphs under consideration had certain number of twins. As mentioned earlier, one necessary and sufficient requirement for the computation of Identifying Codes is that the graph be "twin-free". To get around this constraint, two nodes which are "twins" can be condensed to form a super node. This formation of a super node ensures that the graph becomes "twin-free" and as a result, an MICS of such a modified graph exists. It may be noted that in such a scenario, the super node receives a unique signature (or identification). To distinguish between the two nodes which formed a super node, deeper investigation is required, which can be accomplished by analyzing the attributes (or behaviour) of the two users separately. This analysis can be done using various data mining techniques.

In the table, the first highlights the results for the "0" graph in the Facebook dataset. Originally, the graph had 333 nodes and 2519 edges. Using a simple twin detection algorithm, we condensed the twin nodes to form super nodes. At the end of this process, it was observed that the number of nodes in the graph decreased to 312, and correspondingly, the number of edges in the graph became 2418. After applying our ILP, we noted that *only* 85 seed nodes (or users) are required to be monitored by entities such as Politifact/MBFC in order for them to have unique identification for all the nodes in the graph (or network). In other words, by monitoring the behaviour of these 85 users, the fact-checking company can uniquely identify

Table 5: Identifying Code Results

Input Graph	Number of Nodes	Number of Edges	Number of Nodes after Twin Removal	Number of Edges after Twin Removal	Number of Nodes to Target	Reduction in Resources
0	333	2519	312	2418	85	**72.75%**
107	1034	26749	1026	2418	125	**87.81%**
348	224	3192	220	3173	40	**81.81%**
414	150	1693	144	1587	32	**77.77%**
686	168	1656	166	1651	30	**81.92%**
698	61	270	56	243	23	**62.26%**
3437	534	4813	517	4686	113	**78.14%**
3980	52	146	46	135	18	**60.87%**

a user if they become active, with respect to propagating misinformation. Once a user becomes active in propagating misinformation, the respective authorities can take corrective measures. Thus for the “0” graph, 85 nodes are sufficient for unique monitoring of 312 nodes in the graph. This signifies a **72.75%** (((312 - 85) / 312) * 100) reduction in computational resources. This is as opposed to monitoring the behavior of all the users in the network. The results for the other graphs are tabulated in Table 5. The average reduction in resources for all the graphs under study turns out to be **75.42%**. This is a significant reduction in resources.

6 FUTURE WORK AND ADVICE SOUGHT

Effective utilization of Identifying Codes will enable fact-checking and social networking companies to uniquely identify the source of misinformation propagation. As a part of my research, I am also currently studying several variants of the MICS problem. The Budgeted MICS problem introduces a limitation on the number of monitors (or sensors) that may be placed in the network. The Targeted MICS problem does not require all the nodes (or users) in the network to have unique signatures, except only the targeted nodes (or users). The Augmented MICS problem assumes that there are already some monitors present in the network, but additional monitors must be adequately placed to enable unique monitoring of all the users. Furthermore, over the past couple of decades, various research groups have studied Identifying Codes and presented approximate algorithms for the same. As a novel contribution, I am trying to leverage the advancements in network embeddings to develop an algorithm for the computation of MICS.

One major assumption of the approach discussed thus far is that the retweeting or sharing misinformation (corresponding to color seepage) is deterministic. In contrast, a user may or may not propagate the misinformation it receives, to its friends. This creates a probabilistic scenario of MICS and I intend to seek insights from the peers and experts at the consortium into this challenge.

ACKNOWLEDGEMENT

I would like to thank my advisor Dr. Arun Sen, for his invaluable inputs and continuous support in my research efforts.

REFERENCES

[1] Mohammad-Ali Abbasi and Huan Liu. 2013. Measuring user credibility in social media. In *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*. Springer, 441–448.

[2] Kaustav Basu, Malhar Padhee, Sohini Roy, Anamitra Pal, Arunabha Sen, Matthew Rhodes, and Brian Keel. 2018. Health Monitoring of Critical Power System Equipments Using Identifying Codes. In *International Conference on Critical Information Infrastructures Security*. Springer, 29–41.

[3] Kaustav Basu, Chenyang Zhou, Arunabha Sen, and Victoria Horan Goliber. 2019. A Novel Graph Analytic Approach to Monitor Terrorist Networks. *arXiv preprint arXiv:1902.02836* (2019).

[4] Irène Charon, Olivier Hudry, and Antoine Lobstein. 2003. Minimizing the size of an identifying or locating-dominating code in a graph is NP-hard. *Theoretical Computer Science* 290, 3 (2003), 2109–2120.

[5] Giovanni Luca Ciampaglia, Prashant Shiralkar, Luis M Rocha, Johan Bollen, Filippo Menczer, and Alessandro Flammini. 2015. Computational fact checking from knowledge networks. *PloS one* 10, 6 (2015), e0128193.

[6] Mehrdad Farajtabar, Jiachen Yang, Xiaojing Ye, Huan Xu, Rakshit Trivedi, Elias Khalil, Shuang Li, Le Song, and Hongyuan Zha. 2017. Fake news mitigation via point process based intervention. *arXiv preprint arXiv:1703.07823* (2017).

[7] Mark G Karpovsky, Krishnendu Chakrabarty, and Lev B Levitin. 1998. On a new class of codes for identifying vertices in graphs. *IEEE Transactions on Information Theory* 44, 2 (1998), 599–611.

[8] Moshe Laifenfeld and Ari Trachtenberg. 2008. Identifying codes and covering problems. *IEEE Transactions on Information Theory* 54, 9 (2008), 3929–3950.

[9] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data.

[10] Saikat Ray, Rachanee Ungrangsi, De Pellegrini, Ari Trachtenberg, and David Starobinski. 2003. Robust location detection in emergency sensor networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, Vol. 2. IEEE, 1044–1053.

[11] Arunabha Sen, Victoria H Goliber, Kaustav Basu, Chenyang Zhou, and Sumitava Ghosh. 2019. On upper and lower bounds of identifying code set for soccer ball graph with application to satellite deployment. In *Proceedings of the 20th International Conference on Distributed Computing and Networking*. ACM, 307–316.

[12] Arunabha Sen, Victoria Horan Goliber, Chenyang Zhou, and Kaustav Basu. 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*. Springer, 329–339.

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

[14] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter* 19, 1 (2017), 22–36.

[15] Kai Shu, Suhang Wang, and Huan Liu. 2018. Understanding user profiles on social media for fake news detection. In *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE, 430–435.

[16] Jukka Suomela. 2007. Approximability of identifying codes and locating-dominating codes. *Inform. Process. Lett.* 103, 1 (2007), 28–33.

[17] Eugenio Tacchini, Gabriele Ballarin, Marco L Della Vedova, Stefano Moret, and Luca de Alfaro. 2017. Some like it hoax: Automated fake news detection in social networks. *arXiv preprint arXiv:1704.07506* (2017).

[18] Ying Xiao, Christoforos Hadjicostis, and Krishnaiyan Thulasiraman. 2006. The d-identifying codes problem for vertex identification in graphs: probabilistic analysis and an approximation algorithm. In *International Computing and Combinatorics Conference*. Springer, 284–298.