A Comprehensive Survey on Community Deception Approaches in Social Networks

N. Kalaichelvi, K. S. Easwarakumar

A Comprehensive Survey on Community Deception Approaches in Social Networks

Ms N Kalaichelvi[1] * and Dr K S Easwarakumar[2]

[1] Department of Computer Science and Engineering, Anna University, Chennai, Tamilnadu, India
pnkalai@gmail.com
[2] Department of Computer Science and Engineering, Anna University, Chennai, Tamilnadu, India

Abstract. Community detection techniques seek to find densely connected clusters within large networks. However, it raises privacy concerns, like the personal reveal or community members group information, and contradicts the desire of individual or group to remain anonymous. As a result, concealing a specific community in a network to avoid the finding by methods of community detection becomes critical. Some previous work focuses on hiding some sensitive communities in order to conceal the community association of the focused vertices. Community deception is achieved by changing the links between the vertices in a network minimally so that a specific community can hide as much as possible from a community detection algorithm. This article discusses a study on previously proposed community deception techniques, as well as a discussion of the performance measures used to evaluate community deception methods.

Keywords: Social network · Community detection · Community deception · Community hiding · Modularity · Permanence.

1 Introduction

As we move more and more of our activities online, the demand for social data continues to rise. Because social media is such a popular mode of communication, the massive amount of data it generates necessitates a massive level of data computation on the part of researchers and scientists. A community is a group of users who share similar characteristics and serves as the fundamental functional unit in any social network. Community detection [10] refers to the process of locating clusters of members with the same kind of interests in a network [16]. Several community detection algorithms have previously been developed [5], and research on this topic is still ongoing. Community determination is a critical task in a wide range of social network analysis applications, like customer segmentation, recommendation systems, link inference, vertex labelling, and analyzing influential members.

* Corresponding author

With the increased awareness of the importance of protecting personal privacy in online, we are now prioritizing the protection of privacy of communities in social networks instead of analyzing the information of possible communities in the networks. Community deception refers to the development of concealing techniques of the presence of a given community using community detection methods. Community deception assists users of social networks in concealing their identities from online monitoring. It also assists law enforcement organizations in criminal acts identification involving identity of online [6]. This could also be used by anti-terrorism division to insert secret agents into any terrorist network. This problem is solved by assisting the secret agents in determining with whom they can begin a fresh companionship with (adding edges) and who they must end the relationship (deleting edges) for concealing their identity of community. Paper [9] discussed deception algorithms and provided an overview of previous works on the topic.

Thus, concealing a single or more specific community in a network to avoid detection by analysis tools of social networks is extremely important. This article is a survey of previous works on community deception as well as the metrics used to evaluate community deception algorithms. Reference [7] outlines three ideal goals for measuring community privacy. As an example, consider a given network G and a specific community C, better concealing community C in G, which have three characteristics:

1. *Reachability Preservation:* To maintain information exchange, users of C must be in same connected component and reachable between them; so that, edge perturbation changes can not cut the connectivity of specific community C.
2. *Community Spread:* Users of C should be scattered across other communities in G to the maximum.
3. *Community Hiding:* Users of C must be distributed in the clusters which are most populous.

The rest of this paper is organized as follows. Section 2 explains the fundamentals of community deception. Section 3 discusses various existing deception algorithms for concealing a community. Then, evaluation parameters for measuring the deception are described in Section 4. Finally, Section 5 concludes the article.

2 Community Deception

This section briefly tells about community deception. $G = (V, E)$ denotes an undirected social network, where V is the set of nodes and E is the set of edges in the graph. The community deception algorithm for a network $G = (V, E)$ is to hide a given community C using network edge modifications by a parameter β, called budget update. As a optimization problem, it can be stated as follows:

$$\text{argmax } \eta(C, E(C), \beta, E'(C)) \tag{1}$$

where, $E'(C) = (E(C) \cup E^{add)} \setminus E^{del})$. Here E^{add} denotes the set of edges to be added and E^{del} denotes the set of edges to be deleted to hide C such that $\left|E^{add}\right| + \left|E^{del}\right| \leq \beta$.

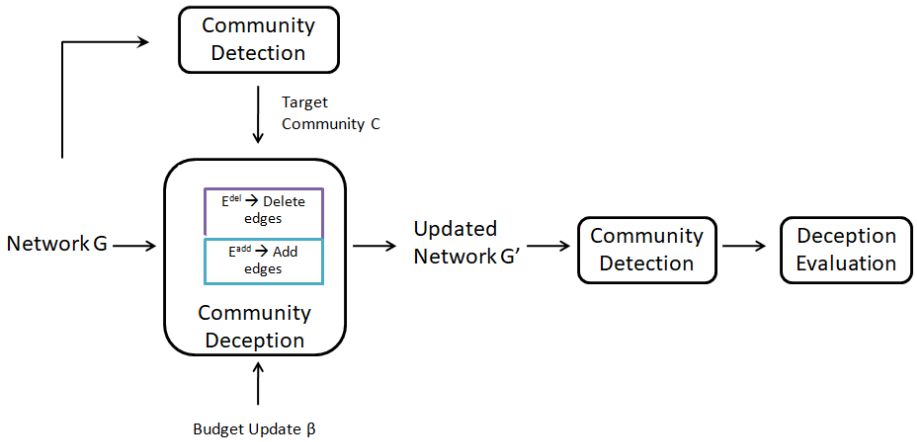

Fig. 1. Deception framework

Figure 1 shows the flow of community deception method. First, community structure is identified for the given network using a community detection technique. Then the target community from the community structure, budget update β and the network G are given to the community deception module in which β edges of the target community are modified and gives the updated network G'. Finally, the deception score is evaluated for the updated network. Various community deception algorithms are discussed in the following section.

3 Community Deception Approaches

3.1 Modularity based Community Deception Methods

Modularity [15] is a popular metric for assessing the quality of any given community structure. It encourages structures with solid connections within communities but light connections between them. As a result, community detection algorithms are typically designed to obtain a structure with the highest modularity. Modularity is calculated by subtracting the expected number of edges placed at random from the number of links falling within groups. Modularity is denoted by:

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(C_i, C_j) \tag{2}$$

where, m denotes the number of links in the network, A_{ij} represents the element of the adjacency matrix of the network, i.e., $A_{ij} = 1$, if there is an relationship among i and j, $A_{ij} = 0$ or else. k_i and k_j are the degrees of vertex i and vertex j respectively. C_i and C_j are the community labels of vertex i and vertex j. Delta, the Kronecker delta function is 1 if both i and j are in one community, and 0 otherwise [11].

Nagaraja The primary goal of a community deception algorithm is to keep hiding the community C from being revealed by any community detection algorithms. Nagaraja was the first to present this problem [14]. The author investigated how far any community can conceal by changing a few links; this approach only drives on addition of edges toward vertices which have high centrality values (like degree centrality). Vertex centrality measures are used to select the endpoints of newly added edges. Experiments were carried out on a small particular email network versus one modularity maximized detection algorithm.

DICE Waniek et al. [17] too addressed this issue by implying a modularity based strategy. DICE, how their algorithm works is by arbitrarily removing internal links between C members and adding external links between C members and non members. It is a heuristic algorithm that hides communities by disconnecting internally and connecting externally. However, this strategy does not always result in a loss of modularity. Certainly, the edge update suggested by the author's strategy increases modularity in some cases.

***D_{mod}* Algorithm** Fionda and Pirr [6] used modularity to solve the community deception problem as well. For a network G with m edges, modularity of the community set of this network $\overline{C} = \{C_1, C_2, ...C_k\}$, is calculated by [7]:

$$M_G(\overline{C}) = \frac{\eta}{m} - \frac{\delta}{4m^2} \tag{3}$$

where $\eta = \sum_{C_i \in \overline{C}} |E(C_i)|$, $\delta = \sum_{C_i \in \overline{C}} deg(C_i)^2$.

A natural approach for attaining deception is by using the modularity loss $ML = M_G(\overline{C}) - M_{G'}(\overline{C}); G'$ is the updated network after deception. the goal here is to discover the group of β relationship changes with ML is highest. Maximizing the loss makes the splitting C provided by the community detection algorithm less ideal and thus increases the level of concealment of community C.

The maximum loss is obtained by selecting the clusters with the maximum degrees as the source and target communities for adding the links. An inter-community edge addition results in the most significant modularity loss. With respect to modularity loss, the best edge for deleting is an intra-community link in the community C_i with the least degree. This algorithm compares the two most convenient edge updates at every move and selects the update with maximum modularity loss. Understanding the degrees is required to select the change of link that will result in greatest modularity loss.

3.2 Safeness based Community Deception Methods

***D_{Saf}* Algorithm** Fionda and Pirr [6] proposed a new metric called safeness and greedily optimized it in order to conceal users of a specific community from detection by community detection algorithms. They created a greedy optimization algorithm to conceal any given community depends upon the metric safeness gain, a new measure defined to find how secure a vertex is on attack. In addition to this new metric, the authors also defined a deception score to calculate the impact of the community deception method on the given network. They also confirmed that this technique outclasses other approaches depending on modularity.

Let $G = (V, E)$ be a network, $C \subseteq V$ a community, and $v \in C$. The safeness $\sigma(v, C)$ of v in G is calculated by [6]:

$$\sigma(v, C) = \frac{1}{2}\frac{|V_C^v| - |E(v, C)|}{|C| - 1} + \frac{1}{2}\frac{\left|\widetilde{E}(v, C)\right|}{deg(v)} \tag{4}$$

where $V_C^v \subseteq C$ is the group of vertices accessible from v passing only through vertices in C. $E(v, C)$ represents the set of intra-community edges for a node $v \in C$ and $\widetilde{E}(v, C)$ represents the set of inter-community edges of the node v.

From this, the safeness of community C is gievn below:

$$\sigma(C) = \sum_{v \in C} \frac{\sigma(v, C)}{|C|} \tag{5}$$

Safeness of C can be defined by starting with the safeness of the elements, allowing us to find the least safe elements and modify their edges to raise the overall C safeness-score. By using the general deception formulation, the algorithm uses the safeness-gain $\xi_C = \sigma(C') - \sigma(C)$, here C' is the community after a few link updates. This algorithm employs a greedy strategy, selecting an edge modification which yields the maximum ξ_C at every stage. For additions, the algorithm only considers inter-C edges and ignores intra-C edges in order to achieve the best safeness gain.

Hs Algorithm - Community Hiding via Safeness Chen et al., [3] used the modified safeness for community deception problem which is given below:

$$\sigma(C) = \frac{1}{2}\psi(C) + \frac{1}{2}\varphi(C) \tag{6}$$

$\psi(C)$ is Intra Community Safeness and denoted by:

$$\psi(C) = \frac{\rho(C) - 2(n-1)^2}{\sum_{k=1}^{n}(\sum_{i=1}^{n-k} i + \sum_{j=0}^{k-1} j) - 2(n-1)^2} \tag{7}$$

If $E(v, C)$ as the set of intracommunity edges for node $v \in C$ and $\widetilde{E}(v, C)$ is the set of intercommunity links , then the $\varphi(C)$ is defined as:

$$\varphi(C) = \sum_{v \in C} \frac{\left|\widetilde{E}(v, C)\right|}{deg(v)} \tag{8}$$

For the community deception problem, Chen designed a safeness-gain $\Delta(C) = \sigma(C') - \sigma(C)$, here C' is the community after few edge perturbation changes. So, this method chooses the suitable edge modification which results in the maximum $\Delta(C)$ at every step. This technique uses inter-C edge additions while adding and uses intra-C edge removals for the better safeness gain. Hs algorithm outperforms D_{saf} in most cases.

3.3 Permanence based Community Deception Methods

Permanence is used as a measure to find how to efficiently modify any given network in order for concealing the given community [1]. Permanence is a measure that quantifies a node w's containment in a network community C [2]. The definition of permanence of w is:

$$Perm(w, G) = \frac{I(w)}{E_{max}(w)} \times \frac{1}{deg(w)} - (1 - C_{in}(w)) \tag{9}$$

This value specifies that a vertex will be in the same cluster if its internal pull $I(w)$ exceeds its external pull $E_{max}(w)$. Here, $deg(w)$ denotes the degree of node w and $C_{in}(w)$ denotes the internal clustering coefficient of w. Then the permanence of a network G is defined as follows:

$$Perm(G) = \sum_{w \in V} \frac{Perm(w)}{|V|} \tag{10}$$

NEURAL The primary goal was to develop an algorithm capable of concealing an specific community C from a community detection method with least edge rewiring. In other words, the community detection method should not reveal the real community membership information of vertices within C. This was accomplished by rearranging the network's structure by means of a predetermined number of link updates (β - budget). Considered only intercommunity edges when adding edges, and only intracommunity edges when deleting edges.

Shravika et al., used the permanence loss as the new objective function in the NEURAL algorithm [13]. This community deception method reduced the network's durability for a specific community C to remain unseen from community detection methods. The authors proposed this because reducing a vertex's permanence would disorder its containment in the actual cluster, altering the network's community framework and building it hard for detection techniques to find the actual communities. Link updates (adding/deleting) were sought out by maximizing the permanence loss at each step, which is given below:

$$P_l = Perm(G) - Perm(\widetilde{G}) \tag{11}$$

here G denotes the actual network and $\widetilde{G}$ denotes the updated network subsequent to modifying the links, related to the given community C.

3.4 Entropy based Community Deception Methods

REM This raises the issue of community structure deception (CSD), which seeks methods to minimally modify the network so that a given community structure conceals from community detection methods as much as possible. Liu et al. [12] projected an approach for hiding the complete community structure (rather than just one community) with least network rewiring. They expanded the task of concealing a specific community to concealing the complete formation of the community. The authors proposed a community structure deception algorithm depending upon information theory and network entropy minimization. They approached the problem with community-based structural entropy and proposed a residual minimization (REM) algorithm.

3.5 Community Deception in Weighted Networks

SECRETORUM The authors of [8] proposed a novel approach for deception in weighted networks using the secrecy gain like D_{saf} algorithm. They have defined the secrecy of a node v in a weighted network $G = (V, E, w)$ is:

$$\sigma_w(v, C) = \frac{1}{2}\frac{|V_C^v| - W_C^v}{|C| - 1} + \frac{1}{2}\frac{\widetilde{W_C^v}}{W^v} \tag{12}$$

where $V_C^v \subseteq C$ is the group of vertices accessible from v passing only through vertices in C. W_C^v is the sum of the weights of all intra-community edges for the node $v \in C$. $\widetilde{W_C^v}$ denotes the total weight of all inter-community edges and W^v represents the total weights of all edges adjacent to v. Then the secrecy of the community C is :

$$\sigma_w(C, G) = \sum_{v \in C} \frac{\sigma(v, C)}{|C|} \tag{13}$$

This deception technique finds the best edge modification which gives the best secrecy gain $\sigma_w(C, G') - \sigma_w(C, G)$ at each step. As usual this algorithm also considers inter-community edge additions and intra-community edges deletions for rewiring the graph.

4 Evaluation Measures for Community Deception

Evaluation metrics are used to quantify the performance of community deception algorithms. The performance indicators used in existing community deception approaches are listed below.

4.1 Deception Score

For a target community CT and a community structure $\overline{CS} = \{C_1, C_2, ...C_k\}$ detected using a community detection technique, deception score of the community is determined as below [6]:

$$H(CT, \overline{CS}) = \left(1 - \frac{|S(CT) - 1|}{|CT - 1|}\right) \times \left(\frac{1}{2}\left(1 - \max_{C_i \in \overline{CS}} \{R(C_i, CT)\}\right) + \frac{1}{2}\left(1 - \frac{\sum_{C_i \cap CT \neq \emptyset} P(C_i, CT)}{|C_i \cap CT \neq \emptyset|}\right)\right) \tag{14}$$

where $|S(CT)|$ denotes total connected components in the subgraph induced by CT's elements. H takes the value from 0 to 1. If all of the vertices in CT are present in a single component and are consealed optimally, then $H = 1$. $H = 0$ denotes that every vertex in CT is from different components. $P(C_i, CT)$ is the precision and $R(C_i, CT)$ is the recall and both are given by:

$$P(C_i, CT) = \frac{\#\text{ CT's members in } C_i \text{ detected by } CD}{|C_i|} \; \forall C_i \cap CT \neq \emptyset \tag{15}$$

$$R(C_i, CT) = \frac{\#\text{ CT's members in } C_i \text{ detected by } CD}{|C|} \; \forall C_i \in \overline{CS} \tag{16}$$

4.2 Normalized Mutual Information (NMI) and Modified NMI (MNMI)

NMI score among the actual community structure of given network with N vertices, $C_A = (C_1, C_2, ..., C_k)$, the updated community structure detected from a community detection method on the modified network, $C_B = (C'_1, C'_2, ..., C'_k)$,. This metric value ranges from 0 (no overlap of C_A and C_B) to 1 (whole overlap of C_A and C_B) [4].

$$NMI(A, B) = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} C_{ij} log(C_{ij}N/C_iC_j)}{\sum_{i=1}^{C_A} C_i log\frac{C_i}{N} + \sum_{j=1}^{C_B} C_j log\frac{C_j}{N}} \tag{17}$$

NMI between nodes in the target communities and their immediate neighbors' community memberships before and after the edge updates is called MNMI. It has the same frequency range as NMI.

4.3 Community Splits

This measure represents the total clusters in the new community structure CS' that contain vertices from the given community C in the modified network G' [13].

$$CommS = \sum_{C'_i \in CS'} h(C'_i, C); h(C'_i, C) = \begin{cases} 1 \; V_C \cap V_{C'_i} \neq \emptyset \\ 0 \; V_C \cap V_{C'_i} = \emptyset \end{cases} \tag{18}$$

here V_C denotes the group of vertices in C and V_{C_i} denotes the group of nodes in community $C_i \in CS'$. This metric values are from 1 (all vertices of C is in single cluster in CS') to $\left|CS'\right|$ (all nodes of C get allocated to various communities of CS').

4.4 Community Uniformity

This metric measures how vertices in the specific community C are allotted across communities in the new community structure CS' [13]. It is measured by finding the entropy of given community vertices distributed between the communities in CS' as follows:

$$CommU = \sum_{C'_i \in CS'} - \left(\left|V_{C,C'_i}\right| / |V_C|\right) \log \left(\left|V_{C,C'_i}\right| / |V_C|\right) \tag{19}$$

where $\left|V_{C,C'_i}\right|$ denotes the total vertices in C present in $C'_i \in CS'$ and $|V_C|$ denotes the total number of nodes present in C. It values are from 0 (entire nodes of C in one community of CS') to $\log \left|CS'\right|$ (entire nodes of C allocated into various communities of CS').

According to the survey, the NEURAL [13], a permanence-based community deception approach, outperforms the other existing community deception techniques in terms of the performance metrics NMI, MNMI, CommS and CommU. However, the authors did not compare the results to the deception score, which is a basic performance measure of community deception. Chen et al. [3] demonstrated that their approach outperforms the techniques proposed in paper [6] in terms of deception score and running time.

5 Conclusion

Most researchers have thus far concentrated on the development of community detection algorithms. Though, in few other situations, it is necessary to conceal the presence of a community. Community deception is the practice of preventing community detection methods from determining the community association of vertices in a specific community. We have listed the existing works that are proposed for community deception in this paper. In addition, we discussed the evaluation measures used to assess the effectiveness of community deception approaches. All of the approaches listed in this paper will be useful to other researchers in developing new methods for community deception that outperform existing methods.

References

1. Chakraborty, T., Srinivasan, S., Ganguly, N., Mukherjee, A., Bhowmick, S.: On the permanence of vertices in network communities. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery

and Data Mining. p. 13961405. KDD '14, Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2623330.2623707, https://doi.org/10.1145/2623330.2623707

2. Chakraborty, T., Srinivasan, S., Ganguly, N., Mukherjee, A., Bhowmick, S.: Permanence and community structure in complex networks. CoRR **abs/1606.01543** (2016), http://arxiv.org/abs/1606.01543
3. Chen, X., Jiang, Z., Li, H., Ma, J., Yu, P.S.: Community hiding by link perturbation in social networks. IEEE Transactions on Computational Social Systems **8**(3), 704–715 (2021). https://doi.org/10.1109/TCSS.2021.3054115
4. Danon, L., Daz-Guilera, A., Duch, J., Arenas, A.: Comparing community structure identification. Journal of Statistical Mechanics: Theory and Experiment **2005**(09), P09008P09008 (Sep 2005). https://doi.org/10.1088/1742-5468/2005/09/p09008, http://dx.doi.org/10.1088/1742-5468/2005/09/P09008
5. El-moussaoui, M., Agouti, T., Tikniouine, A., Adnani, M.E.: A comprehensive literature review on community detection: Approaches and applications. Procedia Computer Science **151**, 295–302 (2019). https://doi.org/https://doi.org/10.1016/j.procs.2019.04.042, https://www.sciencedirect.com/science/article/pii/S1877050919305046, the 10th International Conference on Ambient Systems, Networks and Technologies (ANT 2019) / The 2nd International Conference on Emerging Data and Industry 4.0 (EDI40 2019) / Affiliated Workshops
6. Fionda, V., Pirr, G.: Community deception or: How to stop fearing community detection algorithms. IEEE Transactions on Knowledge and Data Engineering **30**(4), 660–673 (2018). https://doi.org/10.1109/TKDE.2017.2776133
7. Fionda, V., Pirrò, G.: From community detection to community deception. CoRR **abs/1609.00149** (2016), http://arxiv.org/abs/1609.00149
8. Fionda, V., Pirró, G.: Community deception in weighted networks. In: Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. p. 278282. ASONAM '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3487351.3488337, https://doi.org/10.1145/3487351.3488337
9. Fionda, V., Pirró, G.: Community deception in networks: Where we are and where we should go. Complex Networks & Their Applications X. COMPLEX NETWORKS 2021. Studies in Computational Intelligence, vol 1016. Springer, Cham (2022). https://doi.org/https://doi.org/10.1007/978-3-030-93413-2_13
10. Fortunato, S., Hric, D.: Community detection in networks: A user guide. Physics Reports **659**, 1–44 (2016). https://doi.org/https://doi.org/10.1016/j.physrep.2016.09.002, https://www.sciencedirect.com/science/article/pii/S0370157316302964, community detection in networks: A user guide
11. Kalaichelvi Nallusamy, K.S.E.: Cgram: Enhanced algorithm for community detection in social networks. Intelligent Automation & Soft Computing **31**(2), 749–765 (2022). https://doi.org/10.32604/iasc.2022.020189, http://www.techscience.com/iasc/v31n2/44541
12. Liu, Y., Liu, J., Zhang, Z., Zhu, L., Li, A.: Rem: From structural entropy to community structure deception. In: Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019), https://proceedings.neurips.cc/paper/2019/file/328347805873e9a9c700591812fb0ec2-Paper.pdf

13. Mittal, S., Sengupta, D., Chakraborty, T.: Hide and seek: Outwitting community detection algorithms. IEEE Transactions on Computational Social Systems **8**(4), 799–808 (2021). https://doi.org/10.1109/TCSS.2021.3062711
14. Nagaraja, S.: The impact of unlinkability on adversarial community detection: Effects and countermeasures. In: Privacy Enhancing Technologies. PETS 2010. Lecture Notes in Computer Science. vol. 6205, pp. 253–272 (07 2010). https://doi.org/10.1007/978-3-642-14527-8_15
15. Newman, M.E.J.: Modularity and community structure in networks. Proceedings of the National Academy of Sciences **103**(23), 8577–8582 (2006). https://doi.org/10.1073/pnas.0601602103, https://www.pnas.org/content/103/23/8577
16. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Physical Review E **69**(2) (Feb 2004). https://doi.org/10.1103/physreve.69.026113, http://dx.doi.org/10.1103/PhysRevE.69.026113
17. Waniek, M., Michalak, T.P., Rahwan, T., Wooldridge, M.J.: Hiding individuals and communities in a social network. CoRR **abs/1608.00375** (2016), http://arxiv.org/abs/1608.00375