Obfuscation of Node Centrality in Networks

Hiroki Kawamura*, Ryotaro Matsuo†, Hiroyuki Ohsaki*
* *School of Engineering Kwansei Gakuin University*
*E-mail: {hiroki-k, ohsaki}@kwansei.ac.jp
† *Faculty of Engineering Fukuoka University*
†E-mail: r-matsuo@fukuoka-u.ac.jp

***Abstract*—Centrality measures determined by the network's topology structure (e.g., degree centrality, eigenvector centrality, and betweenness centrality) are widely utilized in various fields, such as network analysis and data mining. While extensive research has proposed several centrality indices to measure how important a node is compared to others as well as fast centrality computation algorithms, there is a demand to obfuscate node centrality from a security and privacy perspective. Obfuscating node centrality offers potential applications, such as preventing attacks on important nodes from malicious actors. Therefore, this paper aims to develop an algorithm for obfuscating the centrality of arbitrary sets of nodes in a graph through the addition and deletion of links. Specifically, we introduce COBF (Centrality OBFuscation), which applies the concept of ProHiCo (Probabilistic algorithm to Hide Communities), a community obfuscation method proposed, to obfuscate node centrality. Experiments are conducted to analyze how effectively the centrality of specified nodes is obfuscated. The results demonstrate that COBF can effectively alter node rankings by approximately five positions in a network with a total of 100 nodes, highlighting its potential for enhancing privacy in network analysis.**

***Index Terms*—Node Centrality, Centrality Obfuscation, Network Security, Link Perturbation**

I. Introduction

The centrality of nodes, determined by the network's topological structure (e.g., degree centrality, eigenvector centrality, and betweenness centrality) [1-3], is widely utilized in various scenarios, such as network analysis and data mining. In recent years, networks that grow over time, such as sensor networks, communication networks, social networks, and transportation networks, have been emerging one after another and are expected to continue spreading worldwide [4]. Node centrality measures are frequently employed to analyze and understand the importance of individual nodes within these networks [5, 6].

Over the years, there has been active research on proposing reasonable centrality measures to measure how important a node is compared to others as well as fast centrality computation algorithms [7]. However, from the perspective of security and privacy protection, there is a demand to obfuscate node centrality. For example, highly central nodes possess information and value that can significantly influence the surrounding nodes and the entire network, making it necessary to obfuscate the centrality of such nodes. If node centrality can be obfuscated, it holds the potential for applications such as preventing attacks on important nodes from malicious actors [8].

In this paper, we address the following research questions.

- How can the node centrality obfuscation problem be formally defined to generalize obfuscation for arbitrary node sets in a graph?
- What algorithmic approaches can be developed to effectively solve the node centrality obfuscation problem?
- To what extent does the proposed method perform effectively across different network structures and centrality measures in obfuscating node centrality?

Therefore, this paper aims to develop an algorithm for obfuscating the centrality of arbitrary sets of nodes in a graph through the addition and deletion of links. Previous research has explored the obfuscation of specific community sets within networks through link rewiring [9], as well as the obfuscation of top-ranked nodes with high centrality scores [10]. However, to the best of our knowledge, there has been no investigation into the obfuscation of centrality for arbitrary nodes, including those that are middle-ranked or bottom-ranked nodes with intermediate or low centrality scores, respectively.

Specifically, we propose a centrality obfuscation algorithm called COBF (Centrality OBFuscation), which applies the concept of community obfuscation method ProHiCo (Probabilistic algorithm to Hide Communities) [9] to obfuscate node centrality. Through experiments, we investigate to what extent the centrality of specified nodes is obfuscated. ProHiCo is a method that obfuscates any community set within the network by adding and/or deleting a small number of links. ProHiCo addresses two main issues present in conventional community obfuscation methods, such as REM [11] and safeness [12]. First, it resolves the problem of link optimality, which depends on the objective function used when adding and/or deleting links. Second, it tackles the issue where changes in links at a particular stage affect subsequent link changes, leading to an imbalance in link rewiring resources. We adopt the approach from ProHiCo, where link rewiring decisions are based on pre-calculated weights, to obfuscate node centrality.

The main contributions of this paper can be summarized as follows.

- We formulate a new class of obfuscation problems (i.e., node centrality obfuscation problem) for obfuscating the centrality of arbitrary sets of nodes in a graph.
- We present an effective algorithm for the node centrality obfuscation problem called COBF.
- We extensively examine the effectiveness of COBF

979-8-3315-0694-0/25/$31.00 ©2025 IEEE

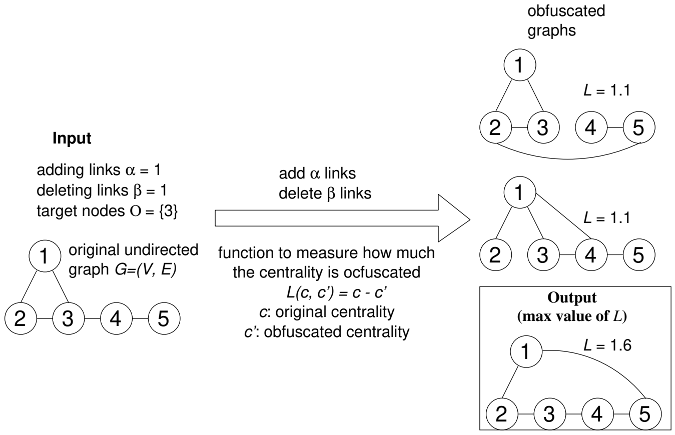


Fig. 1. An example of the obfuscation problem of node centrality (betweenness centrality) in a simple network with 5 nodes and 5 links

through a number of experiments with different network types and centrality measures.

The structure of this paper is as follows. In Section II, we formulate the node centrality obfuscation problem targeted in this paper. In Section III, we explain the COBF algorithm for obfuscating node centrality. In Section IV, we investigate the effectiveness of COBF through experiments. Finally, in Section V, we summarize the paper and outline future tasks.

II. Node Centrality Obfuscation Problem

The node centrality obfuscation problem addressed in this paper involves selecting links for addition and/or deletion in a network, such that the centrality indices of a specified set of nodes are obfuscated as much as possible.

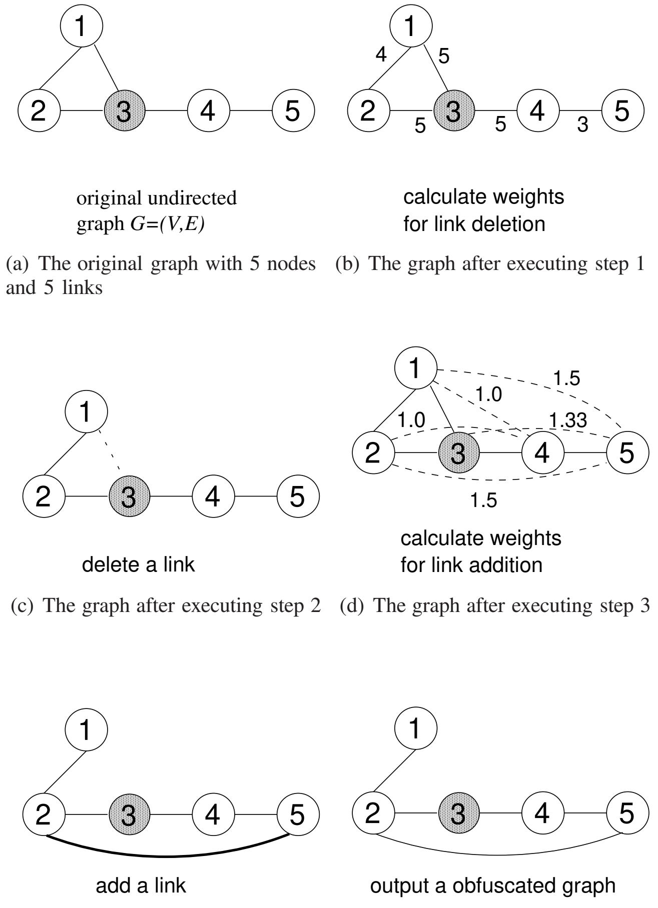
Consider an unweighted undirected graph $G = (V, E)$, a set of nodes for which centrality needs to be obfuscated, denoted as $O \subseteq V$, and the numbers of link additions, α, and link deletions, β, for graph G. Additionally, we are given a centrality function $f(G, v)$ for node v in graph G.

Let G' be a graph obtained after adding to and deleting from graph G. The node centrality obfuscation problem can be formulated as

$$\arg\max_{G'} \sum_{v \in O} L(f(G, v), f(G', v)) \tag{1}$$

where $L(c, c')$ is a utility function that measures the effectivness of obfuscation. For example, if the goal is to decrease the centrality of specified top nodes, one can use $L(c, c') = c - c'$. Alternatively, if the goal is to maximize the sum of the changes in centrality for the specified nodes, one can use $L(c, c') = |c - c'|$ or MSE (Mean Squared Error), which calculates the average of the squared differences between the original centrality scores c and the modified centrality scores c', can be employed.

Fig. 1 provides an example of the node centrality obfuscation problem in a simple network with five nodes and five links. The figure illustrates a case for $\alpha = \beta = 1$ and the utility function of $L(c, c') = c - c'$ is used to obfuscate the centrality of the specified node 3.



(a) The original graph with 5 nodes and 5 links

(b) The graph after executing step 1

(c) The graph after executing step 2

(d) The graph after executing step 3

(e) The graph after executing step 4

(f) The graph after executing step 5

Fig. 2. An example of COBF-weighted execution on a simple graph with 5 nodes and 5 links ($\gamma = 1, \epsilon = 1$)

III. Centrality Obfuscation Algorithm COBF

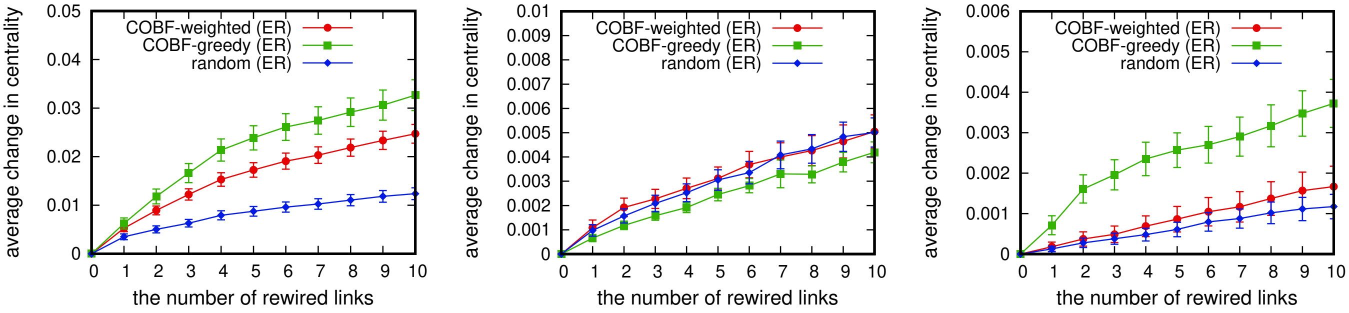
The node centrality obfuscation algorithm, COBF, is inspired by the community obfuscation framework ProHiCo [9]. Typically, obfuscation involves naively adding and/or deleting links around the nodes that are intended to be obfuscated, which can lead to the Matthew effect, where the obfuscation becomes biased toward specific nodes [9]. To prevent this, COBF precomputes link weights and systematically deletes and/or adds links based on these weights to achieve more balanced obfuscation.

A. COBF-weighted

COBF-weighted precomputes the weights of all links in graph G and deletes β links based on these weights. Similarly, it precomputes the weights of all non-links (node pairs without links) in graph G and adds α links according to these weights.

The following provides a detailed explanation of the algorithm for COBF-weighted.

1) For node pairs with links on graph G, $(u, v) \in E$, compute the link deletion weight $W^d = (w^d_{u,v})$ as

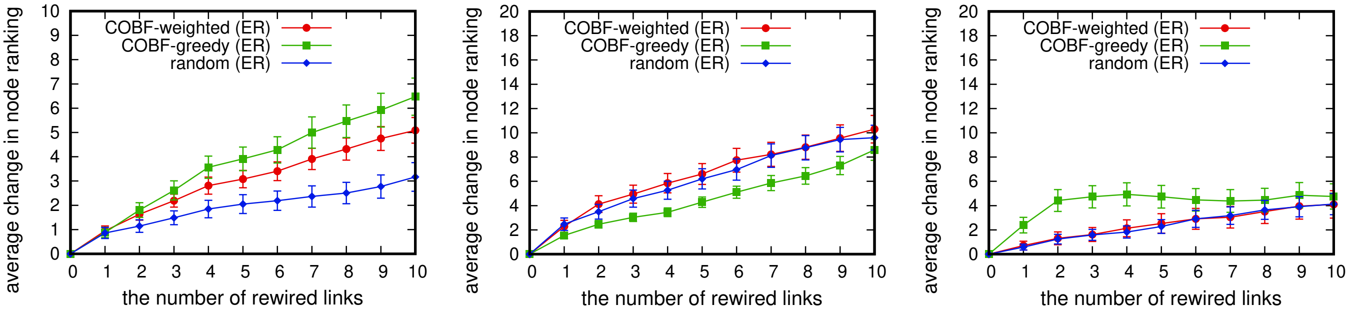


(a) Top (1st, 5th, and 10th)

(b) Middle (45th, 50th, and 55th)

(c) Bottom (90th, 95th, and 100th)

Fig. 3. The obfuscation of betweenness centrality in random graphs generated by ER model



(a) Top (1st, 5th, and 10th)

(b) Middle (45th, 50th, and 55th)

(c) Bottom (90th, 95th, and 100th)

Fig. 4. The obfuscation of betweenness centrality ranking in random graphs generated by ER model

$$w^d_{u,v} = (c_u + c_v)^\gamma \tag{2}$$

where c_v is the centrality of node v, and γ is a parameter.

2) Select a node pair randomly with probabilities proportional to the link deletion weight W^d, and delete the corresponding link from graph G.
3) For node pairs without links on graph G, $(u, v) \notin E$, compute the link addition weight $W^a = (w^a_{u,v})$ as

$$w^a_{u,v} = \left(\frac{1}{c_u} + \frac{1}{c_v}\right)^\epsilon \tag{3}$$

where ϵ is a parameter.

4) Select a node pair randomly with probabilities proportional to the link addition weight W^a, and add the corresponding link to graph G.
5) Repeat steps 1 and 2 for β times, and repeat steps 3 and 4 for α times.
6) The graph G' which results from adding and/or deleting links to the original graph G, is output.

B. COBF-greedy

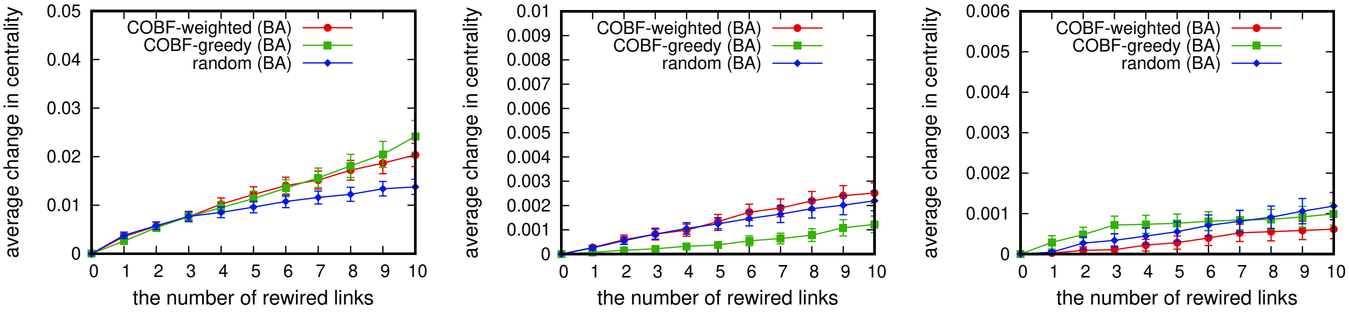
COBF-greedy is fundamentally the same as COBF-weighted, but it differs in how it selects node pairs in steps 2 and 4. In COBF-weighted, pairs are chosen randomly, while COBF-greedy selects node pairs greedily, choosing those with the maximum weights first. This method focuses on nodes with higher weights, which can result in more significant changes to the network structure and centralities compared to the probabilistic approach of COBF-weighted, while still maintaining a balanced approach to obfuscation.

C. Execution Example of COBF

In a simple network with five nodes and five links, an example of COBF-weighted for the problem of obfuscating the centrality of node 3 is shown in Fig. 2. The figure presents the results for the parameters where $\alpha = \beta = \gamma = \epsilon = 1$. In the figure, circles represent nodes, gray-filled circles represent nodes with obfuscated centrality, solid lines represent links between nodes, dashed lines in graph (c) represent deleted links, and dashed lines in graph (d) represent links that can be added. First, the weight for link deletion is calculated for any node pair on graph G based on Eq. (2). Then, based on the calculated weight for link deletion, a node pair is randomly selected with a probability proportional to the weight, and one link is deleted from graph G. Next, the weight for link addition is calculated for any node pair without links on graph G based on Eq. (3). Subsequently, based on the calculated weight for link addition, a node pair is randomly selected with a probability proportional to the weight, and one link is added to graph G. Finally, the resulting graph G' after link addition and deletion is output.

IV. Experiment

In the following, we quantitatively evaluate the effectiveness of COBF across different network types and node centralities. In this experiment, we use two common centrality measures:

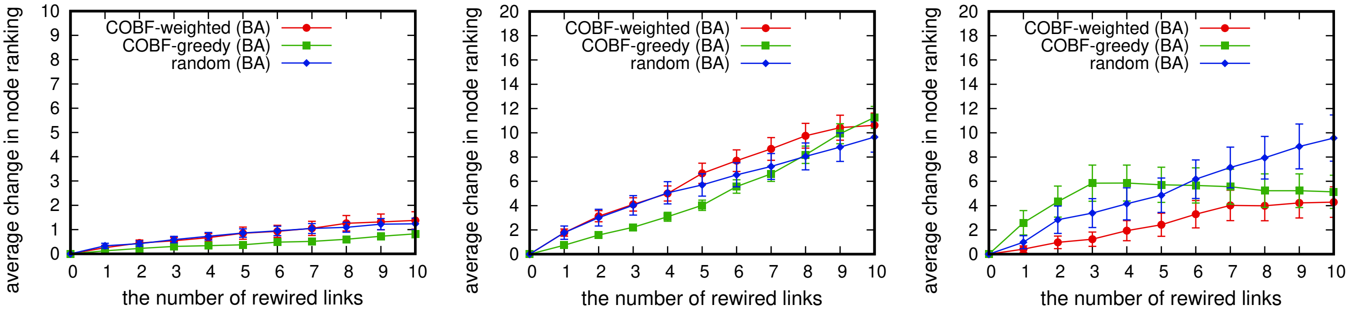


(a) Top (1st, 5th, and 10th)

(b) Middle (45th, 50th, and 55th)

(c) Bottom (90th, 95th, and 100th)

Fig. 5. The obfuscation of betweenness centrality in scale-free graphs generated by BA model



(a) Top (1st, 5th, and 10th)

(b) Middle (45th, 50th, and 55th)

(c) Bottom (90th, 95th, and 100th)

Fig. 6. The obfuscation of betweenness centrality ranking in scale-free graphs generated by BA model

betweenness centrality and eigenvector centrality, to assess the obfuscation performance on typical node centrality metrics.

Using the ER (Erdõs-Rényi) [13] and BA (Barabási-Albert) [14] network generation models, we generated 100 random graphs and 100 scale-free graphs, each with 100 nodes. We varied the number of deleted links β and added links α equally between 1 and 10. Three node sets were selected for obfuscation, based on centrality: a top-ranked set (1st, 5th, and 10th ranked nodes), a middle-ranked set (45th, 50th, and 55th ranked), and a bottom-ranked set (90th, 95th, and 100th ranked).

For each graph G and its obfuscated version G', we measured the obfuscation degree by comparing the centrality $c(v)$ and $c'(v)$ of node v using the following function to express the centrality change:

$$L(c(v), c'(v)) = |c(v) - c'(v)| \tag{4}$$

We also evaluated rank changes in descending centrality order using:

$$L(r(v), r'(v)) = |r(v) - r'(v)| \tag{5}$$

COBF's control parameters were set to $\gamma = 1$ and $\epsilon = 1$.

In addition to COBF, we employed a random method (randomly deleting β links and adding α links).

For the 100 generated graphs, we performed one obfuscation trial for each specified node set O by adding α edges and deleting β edges, calculating the average obfuscation degree and 95% confidence interval for the three methods: COBF-weighted, COBF-greedy, and random method.

In the random graph generated by the ER model, the changes in betweenness centrality and rank are analyzed as the number of link deletions and additions is equally varied. For the top-ranked, middle-ranked, and bottom-ranked node sets, the change in betweenness centrality is given by

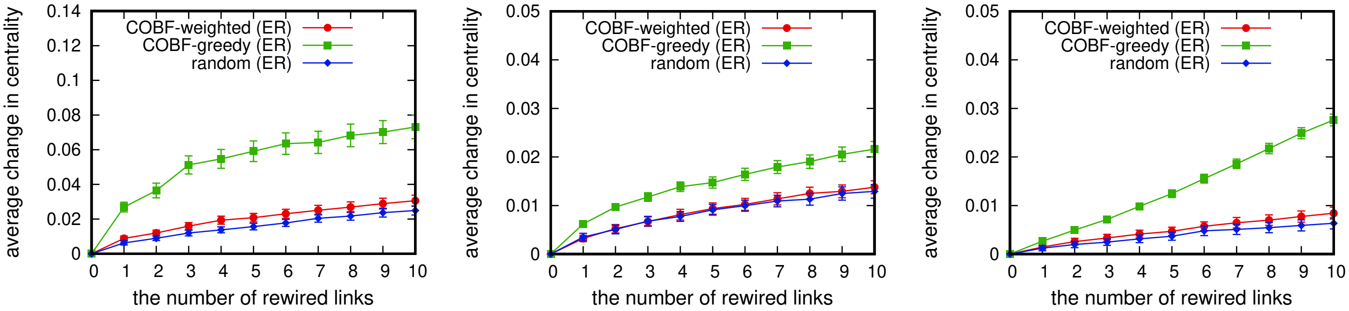
$$\frac{1}{|O|} \sum_{v \in O} L(c(v), c'(v)), \tag{6}$$

and the change in rank by

$$\frac{1}{|O|} \sum_{v \in O} L(r(v), r'(v)). \tag{7}$$

These changes are shown in Fig. 3 and 4. Similarly, the changes in betweenness centrality and rank for the top-ranked, middle-ranked, and bottom-ranked node sets in the scale-free graph generated by the BA model are illustrated in Fig. 5 and 6. These figures present the results of three centrality obfuscation methods: COBF-weighted, COBF-greedy, and random.

From these results, it is evident that, across all graphs, obfuscation of both the top-ranked and bottom-ranked node sets is most effectively achieved by COBF-greedy, with this trend being particularly pronounced in the ER graphs. As for the middle-ranked node sets, all three methods performed similarly, though COBF-weighted appears to show a slight advantage. In terms of numerical results, both COBF-weighted and COBF-greedy consistently alter the rankings by about five positions, demonstrating their ability to successfully obfuscate centrality and achieve the desired effect.

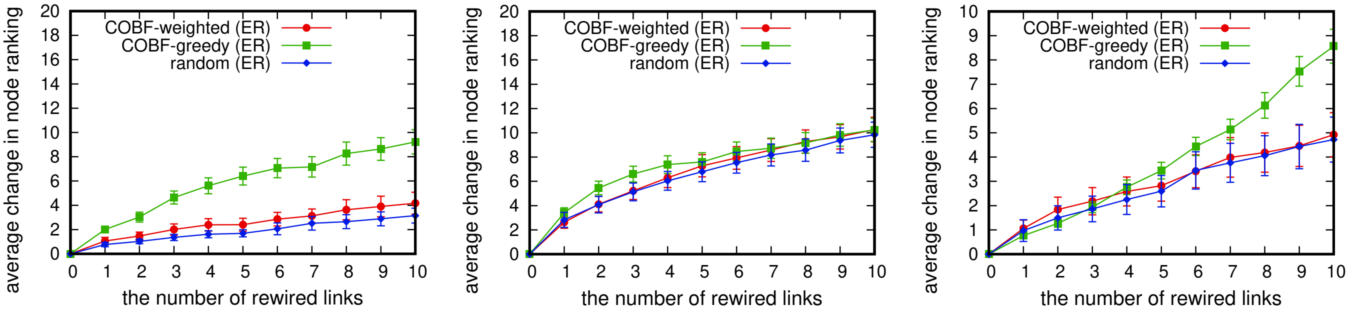


(a) Top (1st, 5th, and 10th)

(b) Middle (45th, 50th, and 55th)

(c) Bottom (90th, 95th, and 100th)

Fig. 7. The obfuscation of eigenvector centrality in random graphs generated by ER model



(a) Top (1st, 5th, and 10th)

(b) Middle (45th, 50th, and 55th)

(c) Bottom (90th, 95th, and 100th)

Fig. 8. The obfuscation of eigenvector centrality ranking in random graphs generated by ER model

In the random graph generated by the ER model, the changes in eigenvector centrality and rank are analyzed as the number of link deletions and additions is equally varied. For the top-ranked, middle-ranked, and bottom-ranked node sets, the change in eigenvector centrality is given by Eq. (6) and and the change in rank by Eq. (7). These changes are shown in Fig. 7 and 8. Similarly, the changes in eigenvector centrality and rank for the top-ranked, middle-ranked, and bottom-ranked node sets in the scale-free graph generated by the BA model are illustrated in Fig. 9 and 10. These figures present the results of three centrality obfuscation methods: COBF-weighted, COBF-greedy, and random.

From these results, it is clear that COBF-greedy consistently outperforms both COBF-weighted and the random method across all node sets top-ranked, middle-ranked, and bottom-ranked achieving nearly double the effectiveness in both centrality and rank changes. In terms of numerical outcomes, both COBF-weighted and COBF-greedy altered rankings by around five positions overall, but COBF-greedy demonstrated a more pronounced effect, consistently shifting ranks by as much as ten positions, confirming its stronger ability to obfuscate node centrality effectively.

V. Conclusion

This paper introduced COBF, an algorithm for obfuscating node centrality indices, inspired by the ProHiCo community obfuscation framework. Two variants of COBF were proposed: COBF-weighted and COBF-greedy. Both methods proved to be effective in concealing node centrality, with COBF-greedy demonstrating particularly strong performance. The results indicated that both approaches could alter node rankings by approximately five positions, and COBF-greedy consistently achieved even greater ranking shifts. These findings confirm COBF's capability to effectively obfuscate centrality, making it a promising approach for enhancing privacy in network analysis.

While COBF has shown its effectiveness, several avenues for future exploration remain. First, scalability to larger, real-world networks is an important aspect to investigate. As networks grow in size and complexity, ensuring that COBF maintains its effectiveness and efficiency will be critical. Another area for future work is extending COBF to obfuscate additional centrality measures, such as closeness or betweenness centrality, beyond those analyzed in this study. Lastly, optimizing the computational efficiency of COBF, especially for applications involving large-scale networks, is an essential direction for enhancing its practical applicability.

Acknowledgements

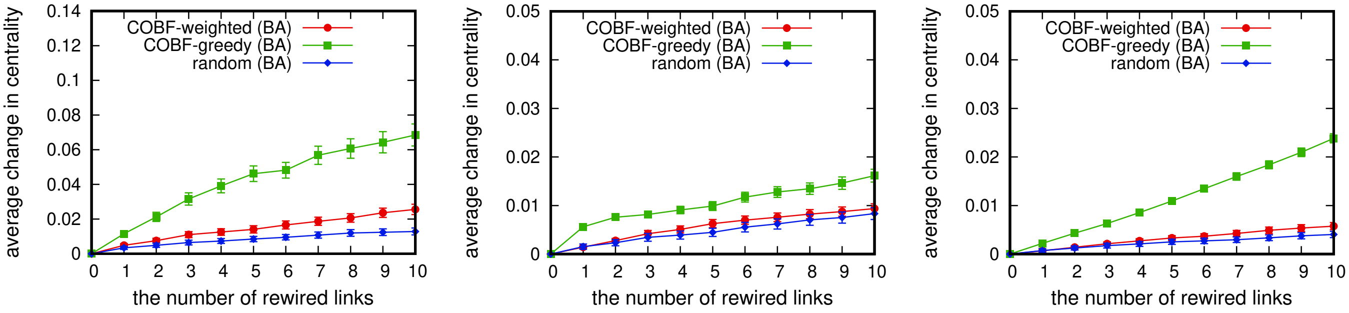
This work was supported by JSPS KAKENHI Grant Number 24K02936.

References

[1] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social Networks*, vol. 1, pp. 215–239, Jan 1978.

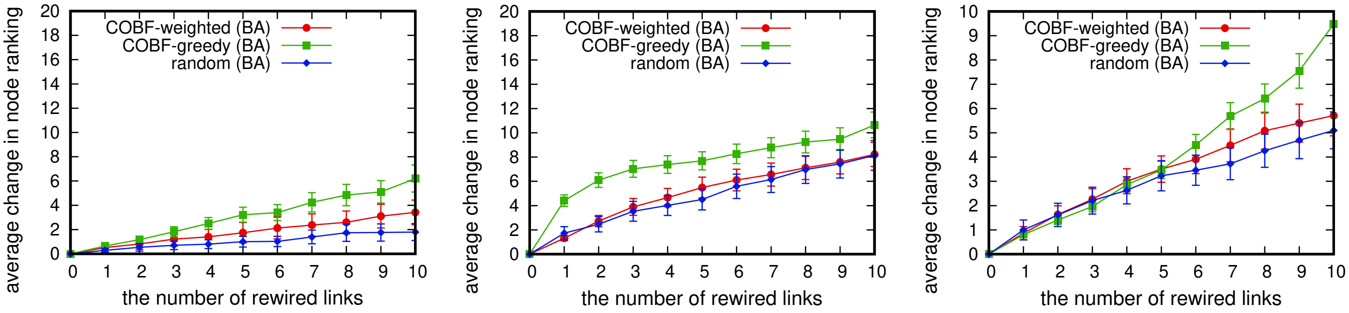
[2] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, vol. 40, pp. 35–41, Mar 1977.

[3] P. Bonacich, "Factoring and weighting approaches to status scores and clique identification," *The Journal of Mathematical Sociology*, vol. 2, pp. 113–120, Jan 1972.



(a) Top (1st, 5th, and 10th) (b) Middle (45th, 50th, and 55th) (c) Bottom (90th, 95th, and 100th)

Fig. 9. The obfuscation of eigenvector centrality in scale-free graphs generated by BA model



(a) Top (1st, 5th, and 10th) (b) Middle (45th, 50th, and 55th) (c) Bottom (90th, 95th, and 100th)

Fig. 10. The obfuscation of eigenvector centrality ranking in scale-free graphs generated by BA model

[4] K. David and H. Berndt, “6g vision and requirements: Is there any need for beyond 5g?,” *IEEE Vehicular Technology Magazine*, vol. 13, pp. 72–80, Jul 2018.

[5] S. Younis and A. Ahsan, “Know your stars before they fall apart: A social network analysis of telecom industry to foster employee retention using data mining technique,” *IEEE Access*, vol. 9, pp. 16467–16487, Jan 2021.

[6] T. Peng, D. Zhang, X. Liu, S. Wang, and W. Zuo, “Central author mining from co-authorship network,” in *2013 Sixth International Symposium on Computational Intelligence and Design*, vol. 1, pp. 228–232, Oct 2013.

[7] K. Bergermann and M. Stoll, “Fast computation of matrix function-based centrality measures for layer-coupled multiplex networks,” *Phys. Rev. E*, vol. 105, pp. 034305–034321, Mar 2022.

[8] S. Zhang, W. Si, T. Qiu, and Q. Cao, “Toward more effective centrality-based attacks on network topologies,” in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pp. 1–6, Jun 2020.

[9] X. Liu, L. Fu, X. Wang, and J. E. Hopcroft, “ProHiCo: A Probabilistic Framework to Hide Communities in Large Networks,” in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM 2022)*, pp. 1–10, May 2021.

[10] M. Waniek, T. P. Michalak, M. J. Wooldridge, and T. Rahwan, “Hiding individuals and communities in a social network,” *Nature Human Behaviour*, vol. 2, pp. 139–147, Jan 2018.

[11] Y. Liu, J. Liu, Z. Zhang, L. Zhu, and A. Li, “REM: From structural entropy to community structure deception,” in *Advances in Neural Information Processing Systems*, vol. 32, pp. 1–11, Dec 2019.

[12] V. Fionda and G. Pirrò, “Community deception or: How to stop fearing community detection algorithms,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, pp. 660–673, Apr 2018.

[13] P. Erdös and A. Rényi, “On random graphs I.,” *Mathematicae*, vol. 6, pp. 290–297, Nov 1959.

[14] A.-L. Barabási and R. Albert, “Emergence of scaling in random networks,” *Science*, vol. 286, pp. 509–512, Oct 1999.