Social Network Analysis and Mining (2024) 14:166
https://doi.org/10.1007/s13278-024-01318-6

ORIGINAL ARTICLE

A benchmark dataset for community deception algorithms

Valeria Fionda[1]

Received: 22 April 2024 / Revised: 18 June 2024 / Accepted: 24 July 2024

Abstract
This paper introduces the Better Hide Communities (BHC) benchmark dataset aimed at standardizing evaluations in community deception across networks. BHC addresses the need for a common framework to assess the effectiveness of existing and perspective deception strategies by enabling their comparative analyses. BHC serves as a foundation for future work in developing sophisticated algorithms for community deception, enhancing the understanding of algorithmic abilities to employ deceptive measures within communities. Additionally, it offers valuable insights into the varying degrees of resilience that different detection algorithms exhibit against deception strategies.

Keywords Social networks · Community deception · Benchmark

1 Introduction

Network analysis is of utmost importance across various fields, spanning from social networks (Yang et al. 2013) to biological systems (Fionda et al. 2008, 2009) and online platforms (Fionda et al. 2016; Revelle et al. 2015). In particular, the detection and analysis of communities (Sobolevsky et al. 2014; Yang et al. 2013) within networks is one of the most critical tasks, enabling the understanding of complex network structures and the interactions within. This process not only helps in identifying densely connected groups of nodes, which often represent functional units or clusters of similar interest, but also in uncovering the underlying patterns and dynamics.

However, the phenomenon of community deception (Fionda and Pirrò 2018; Fionda and Pirrò 2022; Marcin et al. 2018), which intentionally obscures or falsifies the true structure of communities, has recently emerged as a significant challenge. This adversarial tactic can severely impact the accuracy of community detection methods by introducing manipulated data, leading to incorrect interpretations of network dynamics. Community deception's primary goal is to conceal or distort the structural organization of the underlying network for various reasons. These purposes can range from beneficial, such as preserving privacy within groups operating under covert circumstances (e.g., law enforcement units within terrorist networks) or protecting sensitive information, to harmful, including the spread of misinformation or strategic manipulation.

Despite the growing recognition of community deception and the subsequent development of a large number of methods to apply deceptive practices within networks, a significant gap remains in the field: the absence of a reference benchmark dataset. This lack of reference datasets for testing and comparison poses a considerable challenge for researchers and practitioners. Without a common benchmark, evaluating the effectiveness and robustness of different community deception methods becomes inherently difficult, leading to discrepancies in results and hindering the advancement of reliable solutions. Furthermore, the intricate nature of community structures across varied network types necessitates a dataset that is both diverse and representative, enabling the assessment of algorithms under different conditions and scenarios. This diversity is critical for ensuring that deception methods are not only effective in theoretical or isolated cases but are robust and adaptable across the broad spectrum of real-world applications. The establishment of such benchmark would not only facilitate the direct comparison of methodologies under uniform conditions but also accelerate the identification of best practices and the refinement of techniques.

✉ Valeria Fionda
valeria.fionda@unical.it

[1] Department of Mathematics and Computer Science, University of Calabria, via Pietro Bucci 30B, 87036 Rende, CS, Italy

Published online: 21 August 2024

The introduction of the Better Hide Communities (BHC) benchmark dataset responds to these needs by providing a specialized resource for the evaluation of community deception algorithms. BHC is designed to ensure a comprehensive platform that facilitates the analysis and comparison of deception strategies. Through BHC, researchers can explore a range of network configurations and community detection algorithms, enabling a deeper understanding of the strategies' effectiveness and paving the way for the development of more sophisticated and resilient deception methods. The design of BHC is guided by several key considerations to ensure its effectiveness and applicability in diverse scenarios:

- Network Heterogeneity: BHC incorporates a variety of network structures, facilitating the evaluation of algorithms across different network types. This heterogeneity ensures that algorithms tested with BHC are versatile and effective in handling a wide range of network configurations.
- Community Detection Algorithms: A comprehensive array of commonly used community detection algorithms is integrated into BHC. These algorithms serve as a basis for adversarial analysis, aiding in the identification of the most effective network modifications for hiding a specified community.
- Evaluation Metric: The Deception Score (Fionda and Pirrò 2018) is used as the primary metric for evaluation within BHC. This score effectively measures several crucial aspects: the preservation of reachability among nodes within the target community, the spread of the target community in the community structure, and the degree to which the members of the target community are hidden within larger communities.
- Optimal Deception Strategy: BHC provides tailored strategies for each community in each network and each detection algorithm. These strategies detail the optimal set of network modifications, including both edge additions and deletions, to most effectively hide the community from a specific detection algorithm. This feature allows for a comparative evaluation of various community deception algorithms.

Each of these elements is carefully integrated into BHC, creating a robust and comprehensive framework for analyzing and understanding community deception in network structures.

This paper extends "Better Hide Communities: Benchmarking Community Deception Algorithms" (Fionda 2023) by providing a more detailed dataset description, including additional network configurations and parameters, and expanding the results discussion, offering deeper insights into how different detection algorithms perform under various scenario and helping in identifying potential vulnerabilities and strengths within current approaches.

2 Related work

Research on community deception (Fionda and Pirrò 2018) and hiding (Marcin et al. 2018) explores strategies for obscuring a specific target community, denoted as $\mathscr{C}$, within a network's community structure to evade community detection algorithms. These approaches focus on identifying the most effective set of modifications to the network, through optimization of certain functions, to achieve the desired level of deception.

The main body of research on this topic works on undirected networks. Nagaraja (2010) proposed a method to evade detection and hide a community by adding a certain number of edges. Nodes involved in edge addition are chosen by considering some vertex centrality measures such as degree centrality. Marcin et al. (2018) and Fionda and Pirrò (2018) devise deception optimization functions based on modularity (Newman 2006). Since several community detection algorithms exploit modularity, the higher the better, the underlying idea is that by applying edge updates to the network to minimize modularity should mislead such community detection algorithms. Safeness-based deception (Fionda and Pirrò 2018; Chen et al. 2021) has been introduced to correct for some drawbacks of modularity-based deception. In particular, with modularity-based deception, one needs to know the entire network and community structure to identify the best set of edge modifications while Safeness-based deception only requires information about the target community members. Mittal et al. (2021) devised a deception strategy called NEURAL aiming at reducing the permanence of the network for a target community. Permanence (Chakraborty et al. 2016) is a vertex-centric metric that quantifies the containment of a node in a network community.

Some recent proposals also considered a node-centric perspective of the deception problem by allowing also for node deletions and additions. In this respect, Chen et al. (2022) investigate strategies for making minimal yet impactful modifications to improve the privacy of specific individuals by obscuring their connections within a community from detection algorithms. This method employs a selective process for choosing user pairs, coupled with an adversarial graph generator to efficiently alter network links. NDec (Pirrò 2023) introduces a method that leverages modularity loss as a means to identify the optimal series of node modifications (deletion, addition and reassignment of nodes from one community to another). On the samer line, NSaf (Madi and Pirrò 2023) presents a deception tactic that focuses on identifying the ideal node updates (both additions

and deletions) aimed at maximizing safeness, as defined in Fionda and Pirrò (2018).

Recently, some researchers proposed deception algorithms working on directed networks (Fionda et al. 2022), networks with weights or attributes (Fionda and Pirrò 2024) or overlapping communities (Liu et al. 2022). Some recent works investigated the slightly different problem of hiding the entire community structure (Liu et al. 2019; Li et al. 2020; Liu et al. 2022, 2022; Thomas et al. 2021; Chen et al. 2019; Yang et al. 2023; Zhao et al. 2023; Zhao and Cheong 2023). Some more flexible proposals (Chen et al. 2020; Liu et al. 2021; Zhang et al. 2023) can adapt to hide either entire community structures, a set of individual nodes or a community.

3 Preliminaries

3.1 Community deception

The goal of community deception is to design techniques to deceive community detection algorithms. Specifically, given an undirected network $G = (V, E)$, with $n = |V|$ nodes, and $m = |E|$ edges, for a given community $\mathscr{C} \subset V$, the aim is to identify a set of β edge modifications (additions and deletions) to ensure that $\mathscr{C}$ remains undetected by community detection algorithms.

In the following, the degree of a vertex u, denoted as $deg(u)$, is defined as the number of edges connected to u, i.e., $deg(u) = |(u, v) \in E|$. Communities within a network, as identified by a community detection algorithm $\mathscr{A}$, are represented by a community structure $\overline{C} = C_1, C_2, ..., C_k$, with $C_i \in \overline{C}$ being the i-th community in the community structure. We consider non overlapping community structures such that $C_i \cap C_j = \emptyset$ for each pair of communities in $\overline{C}$.

When considering a specific community C_i, edges are categorized as either intra-community or inter-community based on their connectivity. Intra-community edges, denoted as $E(C_i)$, include edges (u, v) where both u and v belong to C_i. Conversely, inter-community edges, denoted as $\tilde{C}_i$, include edges (u, v) where u is a member of C_i but v is not. For a node u within community C_i, $E(C_i, u)$ (and similarly, $E(\widetilde{C}_i, u)$) refers to the set of intra-community (or inter-community, respectively) edges connected to u. The community degree is the sum of the degrees of all nodes within the community, i.e., $\delta(C_i) = \sum_{u \in C_i} \delta(u)$.

The process of community deception is illustrated in Fig. 1 as involving a detector (that is, a community detection algorithm) and a deceptor (that is, a community deception algorithm). The role of the deceptor is to cleverly select a set of modifications (i.e., edge addition and deletions) that will alter the detection outcomes in favor of privacy protection. More formally, given a network $G = (V, E)$, a target community $\mathscr{C}$ and a predefined budget β of modifications, the task of the deceptor is to select the best possible sets E^+ (edge additions) and E^- (edge deletions), such that $|E^+| + |E^-| \leq \beta$, that alter the network's structure in a way that obfuscates the presence of community $\mathscr{C}$ from the detector.

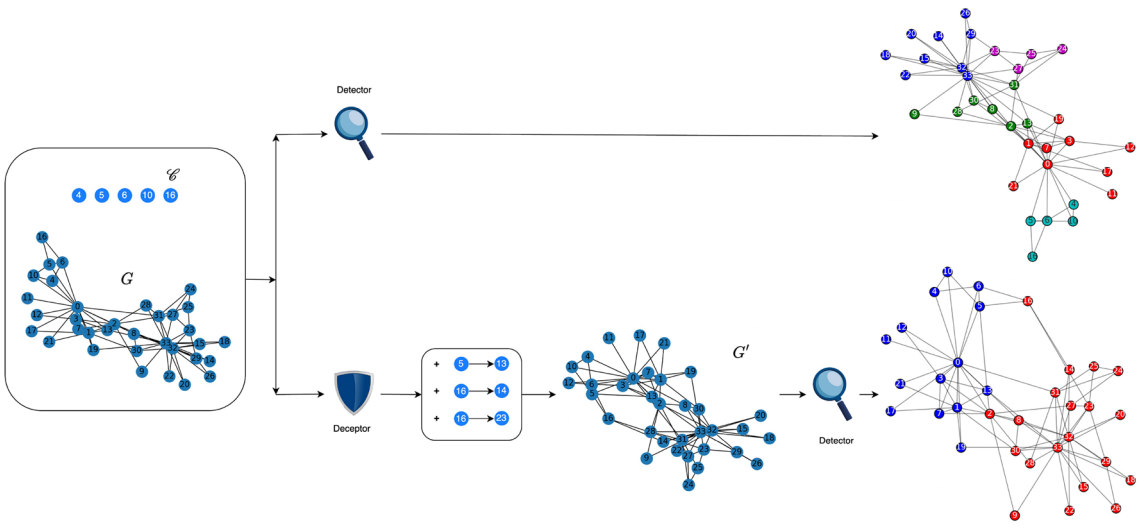

Fig. 1 The process of community deception. Given an input network G, a target community $\mathscr{C}$ and a budget β of modifications, a deceptor strategically performs β edge additions and/or deletions on G, resulting in an altered network G'. On this modified network G' the detector is no more able to clearly identify $\mathscr{C}$ as a distinct community

3.2 Deception score

In the context of community deception, the quantitative evaluation of how effectively a deception strategy hides a target community from detection algorithms is a critical point. Such an assessment not only provides insights into the success of the deception but also support the comparative analysis of different strategies and the refinement of deception techniques. To this end, the Deception Score has been recently introduced (Fionda and Pirrò 2018). The Deception Score is a metric designed to evaluate the extent to which a community has been hidden by a deceptor's actions. It incorporates three key dimensions:

- Reachability Preservation: This aspect measures how the modifications affect the internal connectivity of the target community. It assesses whether the changes preserve the ability of community members to reach each other (via other members of the community), thereby maintaining the functional coherence of the community despite efforts to conceal its presence.
- Community Spread: This dimension evaluates the dispersion of the target community members across the communities in the community structure identified on the modified network. A higher spread indicates a more dispersed community, and makes challenging for detection algorithms to identify the community as a cohesive unit.
- Community Hiding: Measures the effectiveness of embedding the target community members into larger communities detected post-deception.

By combining these factors, the Deception Score provides a comprehensive measure of a deception strategy's efficacy. A higher score corresponds to a greater level of deception achieved, indicating that the target community is more hidden from community detection algorithms.

More formally, given a target community $\mathscr{C}$ and a community structure $\overline{C} = \{C_1, C_2, ...C_k\}$, the deception score $H(\mathscr{C}, \overline{C})$ provides a value in the range [0, 1] and it is defined as follows:

$$\mathcal{H}(\mathscr{C}, \overline{C}) = \overbrace{\left(1\text{-}\frac{|S(\mathscr{C})|\text{-}1}{|\mathscr{C}|\text{-}1}\right)}^{\text{Reachability Preservation}} \times \left(\underbrace{\frac{1}{2}\left(1\text{-}\max_{C_i \in \overline{C}}\{\mathcal{R}(C_i, \mathscr{C})\}\right)}_{\text{Community Spread}} + \underbrace{\frac{1}{2}\left(1\text{-}\frac{\sum_{C_i \cap \mathscr{C} \neq \emptyset} \mathcal{P}(C_i, \mathscr{C})}{|C_i \cap \mathscr{C} \neq \emptyset|}\right)}_{\text{Community Hiding}}\right)$$

where $|S(\mathscr{C})|$ is the number of connected components in the subgraph induced by $\mathscr{C}$'s members; R is the recall of the community $C_i \in \overline{C}$ with respect to $\mathscr{C}$ defined as $R(C_i, \mathscr{C}) = \frac{\#\mathscr{C}\text{'s members in } C_i}{|\mathscr{C}|}$; P is the precision of the community $C_i \in \overline{C}$ with respect to $\mathscr{C}$ defined as $P(C_i, \mathscr{C}) = \frac{\#\mathscr{C}\text{'s members in } C_i}{|C_i|}$.

The optimal scenario occurs when each member of $\mathscr{C}$ is assigned to a different community by a detection algorithm, aiming for the lowest possible maximum recall value (ideally, $1/|\mathscr{C}|$). Community hiding relies on the average precision P and ideally, each C_i in $\overline{C}$ should encompass only a small fraction of $\mathscr{C}$'s nodes. In summary, the deception score approaches 1 if the following conditions are simultaneously satisfied: *(i)* the nodes of $\mathscr{C}$ form a single connected component, satisfying "Reachability Preservation"; *(ii)* each node in $\mathscr{C}$ is distributed across different and significantly sized communities to meet "Community Spread" and "Community Hiding". Conversely, it equals 0 if: *(i)* each $\mathscr{C}$ member is part of a distinct component; or *(ii)* $\mathscr{C}$ is entirely contained within another community. Figure 2 illustrates the variation in the deception score as a result of applying different sets of edge modifications to the network, while employing the same detection algorithm. This example highlights the impact of network updates on the effectiveness of deception strategies.

Reasons for choosing deception score In related research, several alternative metrics have been used to assess deception effectiveness. Metrics such as the Jaccard index, Normalized Mutual Information (NMI), and Recall are effective when evaluating deception across entire community structures. Although these metrics can be adapted to assess the hiding of a single community, they primarily focus on overall structural similarity. Specific metrics, like modularity loss, can measure how well a single community is hidden within a network, but they have limitations, such as being effective only when the detector is a modularity-based detection algorithm. Other possibilities include standard graph

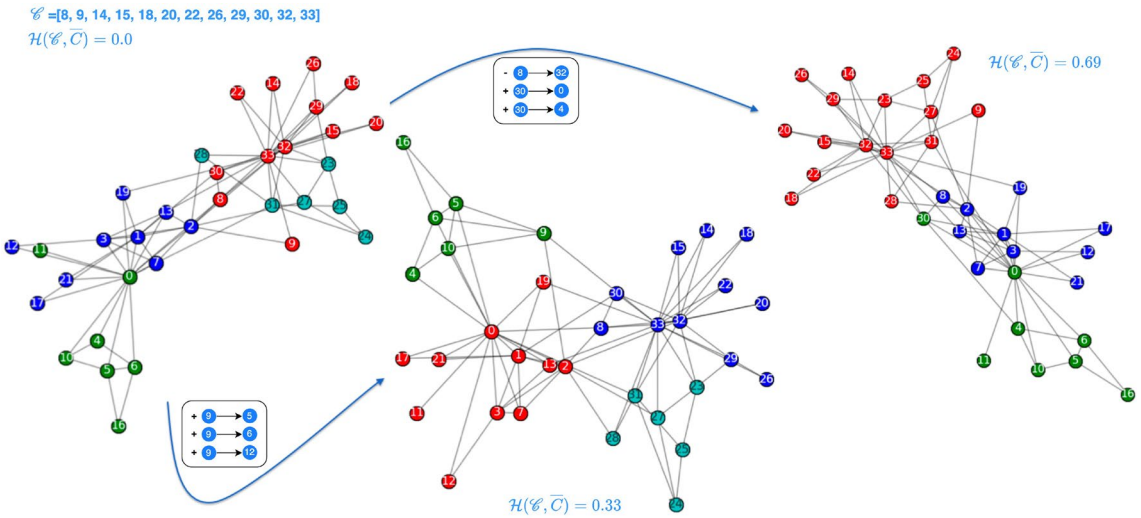

Fig. 2 Variations in Deception Score: an illustration of the effect of edge modifications on deception effectiveness using a consistent detection algorithm

centrality measures such as conductance or edge betweenness, which focus on boundary quality or edge importance rather than providing a comprehensive evaluation of the deception's effectiveness in hiding the community and keeping its internal cohesion.

The Deception Score, on the other hand, was primarily designed to provide a comprehensive assessment of how effectively a community is hidden by integrating multiple dimensions of deception, such as reachability preservation, community spread, and community hiding. It has been chosen over alternative measures for the following reasons:

- The Deception Score integrates multiple dimensions of deception, providing a complete assessment of how effectively a community is hidden, and capturing the multifaceted nature of community deception.
- It provides a quantitative measure that can be consistently applied across different networks and algorithms. This allows for objective comparisons and benchmarking of various community deception strategies.
- The Deception Score directly aligns with the primary goals of community deception, which are to obscure the true structure of the community while maintaining its internal cohesion. By focusing on these specific aspects, the metric ensures that the evaluation is relevant and meaningful for the intended purpose.
- The Deception Score has been successfully used in previous studies (Chen et al. 2021; Fionda and Pirrò 2018; Fionda et al. 2022; Fionda and Pirrò 2024; Madi and Pirrò 2023; Pirrò 2023), showing its effectiveness in measuring community deception. Its adoption in our work leverages this established foundation, ensuring consistency and comparability with prior research.
- The Deception Score is adaptable to various network types and community detection algorithms. This flexibility makes it a versatile tool for evaluating deception strategies in a wide range of scenarios.

4 Benchmarking community deception: a testbed for deception algorithms

In this section, we introduce the Better Hide Communities (BHC) dataset, a comprehensive resource specifically designed to facilitate the study and optimization of community deception algorithms. Such dataset has been built by implementing optimal community deception to maximize deception score. To support reproducibility, further research in this domain and the use of the BHC benchmark dataset, we have made both the code and the dataset publicly available. Interested researchers and practitioners can access and download these resources from the provided GitHub repository at https://github.com/vfionda/BHC.

4.1 Networks

The BHC dataset includes networks ranging from 30 to 50 nodes. For these networks, we have computed the optimal set of updates that must be implemented to hide a given target community in terms of the deception score. This was achieved by exhaustively evaluating all potential subsets of edge modifications, both additions and deletions, within a modification budget ranging from 1 to 3.

The BHC dataset is composed of real and synthetic networks. Table 1 reports the characteristics of the two real networks used in the experimental campaign. We used two

Table 1 Summary of real networks in terms of number of nodes $|V|$, number of edges $|E|$, average degree k, density δ and number of ground truth communities (# coms)

Net	$\|V\|$	$\|E\|$	k	δ	# coms
Karate (Zachary 1977)	34	78	4.58	0.139	2
Dolphins (Lusseau et al. 2003)	62	159	5.12	0.084	4

networks: *(i)* the Zachary's karate club (Zachary 1977), a well-known social network that maps the friendships between 34 members of a karate club at a US university in the 1970s; *(ii)* the Dolphin network (Lusseau et al. 2003), a social network of 62 bottlenose dolphins connected by 159 edges, where an edge represents frequent associations between dolphins. While the degree distribution of the karate network resemble a scale-free network, the dolphin network shows no clear degree distribution pattern.

For the synthetic networks, we used the generator proposed by Lancichinetti et al. (2008). We kept the number of nodes constant while adjusting other parameters. For each fixed node count and specific parameter configuration, we generated either one (for 50-node networks) or five (for 30-node networks) distinct networks. In total, we created 48 networks, with 40 having 30 nodes and 8 having 50 nodes.

The explored network settings, in terms of mixing parameter μ, average degree k, maximum degree $\hat{k}$, minus exponent for the degree sequence t_1 and minus exponent for the community size distribution t_2, are reported in Table 2. This approach facilitated a thorough exploration of the parameter space, thereby ensuring that our benchmark dataset encompasses a diverse range of network structures. In particular, the exponent t_1 plays a crucial role in shaping the degree distribution of the generated networks. A lower value of t_1 typically results in networks with a heavier-tailed degree distribution, where a few nodes have significantly higher degrees compared to the majority. Conversely, a higher value of t_1 leads to a more evenly distributed degree sequence, with fewer nodes having extremely high degrees. On the other hand, the exponent t_2 specifies how the size of communities changes with respect to their frequency. A lower value of t_2 typically indicates that larger communities are more common, resulting in a distribution where a few large communities dominate. Conversely, a higher value of t_2 suggests a more even distribution of community sizes, where communities of various sizes are equally likely to occur. The choice of t_1 and t_2 pairs can have significant implications for the overall network structure. For example, selecting lower values of

Table 2 Summary of synthetic networks in terms of mixing parameter μ, average degree k, maximum degree $\hat{k}$, minus exponent for the degree sequence t_1, and minus exponent for the community size distribution t_2

$\|V\|$	μ	k	$\hat{k}$	t_1	t_2	# nets	# coms
30	0.2	5	10	2	1	5	4,4,4,5,5
30	0.2	5	10	3	2	5	4,4,4,4,5
30	0.4	5	10	2	1	5	4,4,5,5,7
30	0.4	5	10	3	2	5	4,5,5,5,6
30	0.6	5	10	2	1	5	4,4,5,5,7
30	0.6	5	10	3	2	5	5,5,5,5,6
30	0.8	5	10	2	1	5	4,4,5,5,6
30	0.8	5	10	3	2	5	5,5,5,5,6
50	0.2	8	15	2	1	1	4
50	0.2	8	15	3	2	1	4
50	0.4	8	15	2	1	1	5
50	0.4	8	15	3	2	1	4
50	0.6	8	15	2	1	1	4
50	0.6	8	15	3	2	1	5
50	0.8	8	15	2	1	1	4
50	0.8	8	15	3	2	1	5
						48	229

Each entry reports the number of networks generated for the specific configuration and the number of ground-truth communities for each network, with a total of 48 networks generated and 229 communities

both t_1 and t_2 may lead to networks with highly connected nodes forming large communities, resembling scale-free networks with distinct hubs. Conversely, choosing higher values for both exponents may result in networks with a more uniform distribution of node degrees and community sizes, resembling random networks.

In our benchmark we considered two value pairs. The first one, corresponding to $t_1 = 2$ and $t_2 = 1$ suggests a network where both node connectivity and community sizes follow a power-law distribution, but with a relatively gentle slope. This setup might represent social networks where influencers connect many users, and where large interest groups dominate. The second pair is $t_1 = 3$ and $t_2 = 2$ and suggests a network with a degree distribution that follows a power-law with a relatively steep slope. The connections are more evenly distributed among the nodes, meaning that while there are still hubs, the difference in connectivity between the most and least connected nodes is less pronounced than in networks with t_1 values closer to 2. Moreover, with $t_2 = 2$, the network's community size distribution also follows a power-law, but with a characteristic that suggests a balance between the presence of large communities and smaller ones.

In addition, we also considered different values of the mixing parameter μ that quantifies the extent to which nodes in a network are connected to nodes outside their community compared to within their own community. We varied the mixing parameter in the range [0.2, 0.8] with step 0.2. A mixing parameter equals to 0.2 implies that 20% of each node's connections are, on average, to nodes in different communities, while the remaining 80% are within its own community. This indicates a network with strong community structure, where nodes have a strong preference for forming connections within their community: networks with such a low mixing parameter are characterized by high modularity, meaning that they are clearly divided into communities or modules with dense connections internally and sparser connections between them. Conversely, a mixing parameter equals to 0.8 suggests that 80% of each node's connections are to nodes outside its own community, with only 20% of connections within the community. This represents a network with a very weak community structure, where the notion of a "community" is almost negligible. In such networks, the connections are predominantly between communities, indicating a highly integrated or mixed network structure with minimal clustering of nodes into distinct groups. In conclusion, The combination of t_1, t_2 and μ determines the overall structure of the network, including its modularity, assortativity, and resilience to perturbations.

4.2 Detection algorithms

To construct the BHC benchmark, we considered seven deception algorithms available in the CDlib library (Cazabet et al. 2022), each selected for its unique approach to community detection:

- Leiden (`leid`) (Traag et al. 2018) An enhanced version of the Louvain algorithm (Blondel et al. 2008), Leiden employs a multi-level modularity optimization strategy. Its multi-level modularity optimization approach offers greater robustness to noise and outliers in the network.
- WalkTrap (`walk`) (Pons and Latapy 2006) This algorithm leverages the notion that random walks tend to stay within the same community, using this principle to identify communities effectively. It can handle networks with varying densities effectively, making it suitable for networks with sparse or dense connectivity.
- Greedy (`greedy`) (Clauset et al. 2004) Based on a greedy modularity maximization approach, the Greedy algorithm iteratively optimizes the modularity measure to detect communities. It scales well to large networks, making it suitable for real-time or online applications where fast community detection is required.
- InfoMap (`infomap`) (Rosvall and Bergstrom 2008) InfoMap returns a community structure that provides the shortest description length for a random walk, making it a compelling choice for capturing community structure efficiently. It is robust to changes in network topology or dynamics, making it suitable for dynamic or evolving networks.
- Eigenvectors (`eig`) (Newman 2006) Newman's leading eigenvector method is utilized in this algorithm, which detects community structure based on modularity optimization. It offers a global optimization approach to community detection, capturing both local and global structure in the network. Moreover, it is robust to noise and outliers in the network, ensuring stable detection of community structure even in challenging or dynamic environments.
- Paris (`paris`) (Bonald et al. 2018) Inspired by modularity-based clustering techniques, Paris is a hierarchical graph clustering algorithm that offers a unique perspective on community detection. It remains effective across a wide range of network sizes and densities, making it suitable for networks of varying complexities.
- Combo (`combo`) (Sobolevsky et al. 2014) Combo is a modularity maximization implementation of a community detection algorithm, providing a robust approach to identifying communities within networks. Combo algorithm's modularity maximization strategy often yields

high-quality community partitions, making it a reliable choice for community detection tasks.

These algorithms were chosen for their diverse methodologies and strengths in capturing different aspects of community structure within networks. Each algorithm brings unique insights and capabilities to the detection process, ensuring a comprehensive exploration of community detection approaches.

4.3 BHC benchmark generation procedure

The procedure outlined in Algorithm 1 is an exhaustive search strategy designed to identify the set of network updates that maximizes the deception score for each community of a given network G and community detection algorithm $\mathscr{A}$. This algorithm takes a network and iteratively explores all potential network updates within each ground truth community of the network to find the optimal set of modifications at each budget level.

Algorithm 1 BHC benchmark generation

```
1: function BHCGeneration(G, β, 𝒜, C*)
2:     res = []
3:     for 𝒞 ∈ C* do
4:         intraDel =list intra-𝒞 deletion
5:         intraAdd =list intra-𝒞 addition
6:         interDel =list inter-𝒞 deletion
7:         interAdd =list inter-𝒞 addition
8:         for b ∈ {1, .., β} do
9:             maxDec = 0
10:            bestMods = ∅
11:            for mods ∈ combination(intraDel ∪ intraAdd ∪ interDel ∪ interAdd, b) do
12:                G′ =apply network editing in mods to G
13:                for i = 1 to 5 do
14:                    C = 𝒜(G′)
15:                    dec = computeDeceptionScore(C, G′, 𝒞)
16:                    if dec > maxDec then
17:                        maxDec = dec
18:                        bestMods = mods
19:                    end if
20:                end for
21:            end for
22:            res.append((𝒞, β, bestMods, maxDec))
23:        end for
24:    end for
25:    return res
26: end function
```

At each iteration for a given community, the algorithm enumerates all possible intra-community and inter-community updates-namely, additions and deletions (lines 4–7). It then analyzes every possible combination of these updates up to a specified budget limit (line 11). For each combination of updates the algorithm tries to apply such set to the network (line 12), and recomputes the community structure (line 14) to assess the corresponding deception score (line 15). To account for the potential non-determinism inherent in some detection algorithms (e.g., Leiden), the community detection and subsequent deception score calculation are

repeated five times, with the best outcome out of these trials being selected (loop at line 13).

The main goal is to determine the combination of network modifications that most effectively deceives a given detection algorithm, for each ground truth community, reflecting a maximum discrepancy from the ground truth community structure. The results of this process are a set of network updates paired with their associated deception scores for each budget, which are ultimately aggregated and returned as the output of the algorithm.

4.4 Discussion

In this section we present the results of of the BHC dataset generation accompanied by some general considerations.

Figures 3 and 4 present the deception scores for all the community detection algorithms in BHC on the karate and dolphins networks, respectively, by considering the predefined ground truth communities (two for the karate network and four for the dolphins network). The figure differentiates the scores using a color-coding scheme where blue bars correspond to the first community's deception score, red bars to the second's, grey bars to the third and yellow bars to the fourth community. Each algorithm is evaluated based on its initial deception score and its response to the deception-wise best one, two, and three modifications (denoted as 'initial', 1, 2, 3). The initial bars per algorithm serves as a benchmark, reflecting the inherent capability of each algorithm to identify the two ground truth communities prior to any interference.

As expected, subsequent modifications result in a stepwise increase of deception scores, suggesting that it becomes more difficult for detection algorithms to discover the original community as the number of modifications increase. Notably, certain algorithms such as Leiden and Eigenvectors for com0 and Paris for com1 of the karate network, and Greedy, Paris and Combo for some of the communities of the dolphins network, exhibit a steep rise in deception scores even with the first network editing, indicating their high sensitivity to even minimal structural changes within the network. On the contrary, algorithms like InfoMap display a more gradual increase of deception scores upon subsequent modifications, suggesting a certain robustness or perhaps a threshold up to which they can withstand perturbations without significant degradation in community identification.

By comparing the deception scores of the two communities of the karate network in Fig. 3, it emerges that com1 maintains consistently higher deception scores across all detection algorithms and modification budgets; This could indicate that com1 has some characteristics making it more challenging to be detect. By analyzing in more details the results on the dolphin network reported in Fig. 4 it can be noted that Leiden, InfoMap, WalkTrap and Eigenvectors generally show higher deception scores across all budgets, indicating that the communities detected by these algorithms are more susceptible to manipulation. Moreover, it can be noted that Greedy, Paris and Combo show more variability in deception scores across different communities and budgets.

Figure 5 shows in a stacked bar chart the distribution and frequency of modifications applied to the karate network for obtaining the maximum deception score across the various community detection algorithms for the two baseline communities, by considering budget 1, 2, and 3.

By analyzing the chart, several trends emerge. Under the most restrictive budget (1), on the majority of algorithms

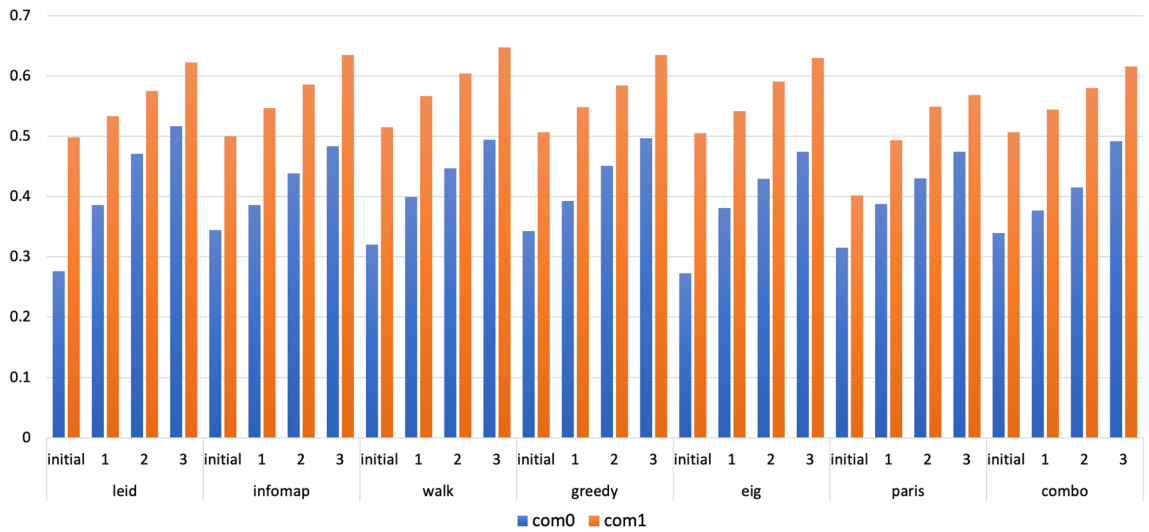

Fig. 3 Deception score for the karate network and the two ground truth communities for all detection algorithms and budget $\beta \in \{1, 2, 3\}$

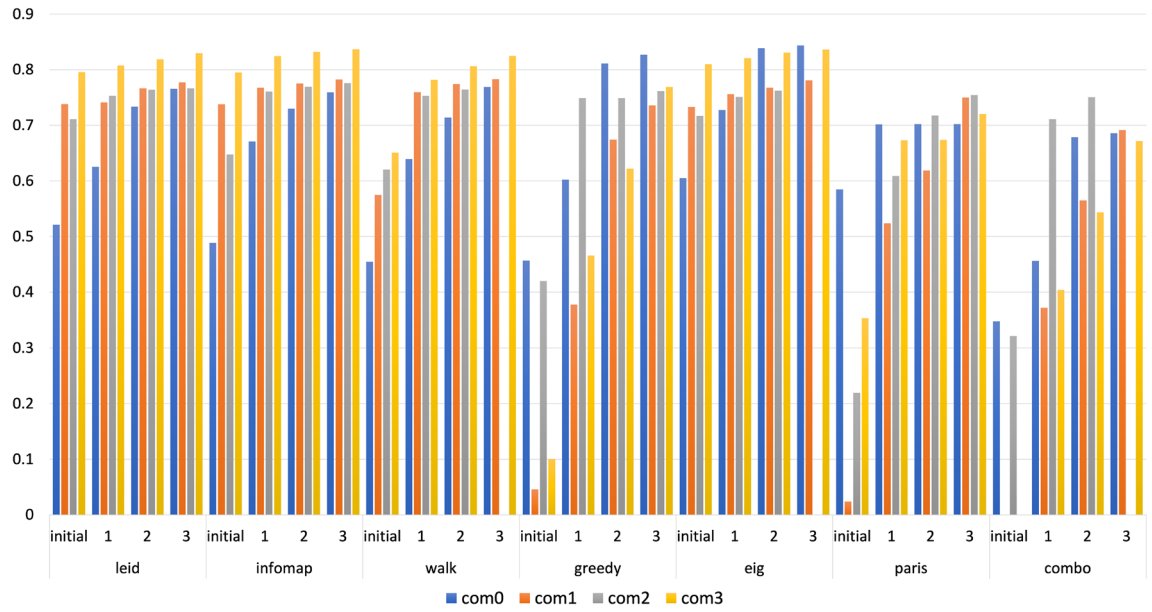

Fig. 4 Deception score for the dolphins network and the four ground truth communities for all detection algorithms and budget $\beta \in \{1, 2, 3\}$

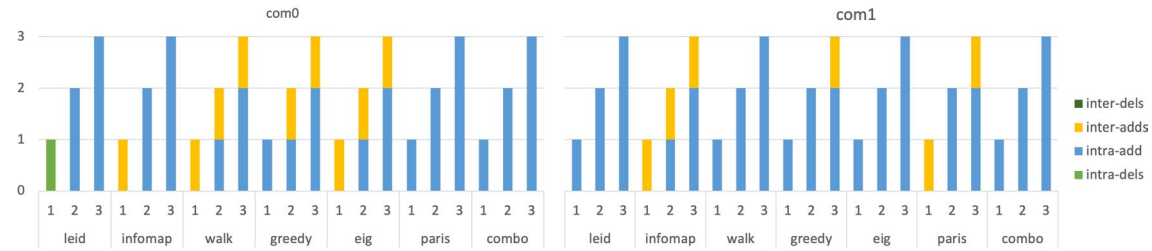

Fig. 5 Distribution of types of modification for community detection algorithms on the karate network by budget values

the best deception strategy rely primarily on intra-community and inter-community additions, except for the Leiden algorithm when considering com0, where best deception is obtained with intra-community deletions. As the budget increases, there is a noticeable shift towards inter-community additions, highlighting that this type of modification can be considered more effective in increasing deception when more changes are allowed.

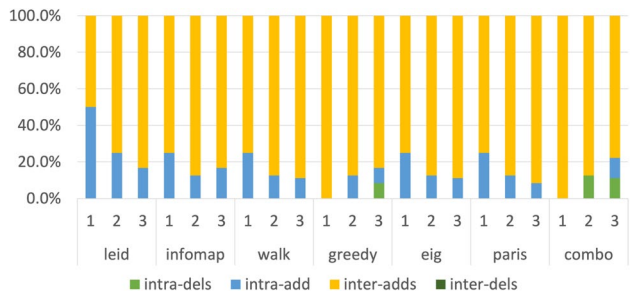

Fig. 6 Distribution of types of modifications for community detection algorithms on the dolphins network by budget values aggregated over the four ground truth communities

The optimal deception of algorithms such as WalkTrap, Greedy, Eigenvector, and Combo show a consistent pattern of modifications across all budget levels. In contrast, optimal deception in case of algorithms like Leid, InfoMap, and Paris displays variable strategies for increasing budgets, which may indicate a strategic shift when given more freedom to modify the network. Inter-community deletions is notably absent across all algorithms and budgets. As one can expect, this suggest that such modifications are less effective at deceiving deception algorithms.

Figure 6 shows the percentage of each modification type (intra-community deletions, intra-community additions,

max increase

			leid			infomap			walk			greedy			eig			paris			combo		
μ	t1	t2	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
0.2	2	1	0.228	0.250	0.250	0.006	0.167	0.170	0.187	0.254	0.256	0.560	0.704	0.754	0.167	0.289	0.323	0.675	0.675	0.755	0.556	0.556	0.689
0.4	2	1	0.313	0.364	0.489	0.370	0.397	0.398	0.249	0.288	0.364	0.501	0.620	0.691	0.267	0.344	0.349	0.585	0.698	0.800	0.494	0.699	0.799
0.6	2	1	0.256	0.498	0.497	0.361	0.391	0.445	0.306	0.495	0.528	0.328	0.450	0.617	0.333	0.415	0.359	0.420	0.576	0.658	0.390	0.542	0.560
0.8	2	1	0.404	0.570	0.574	0.376	0.491	0.569	0.324	0.495	0.564	0.399	0.441	0.587	0.389	0.576	0.594	0.463	0.562	0.602	0.288	0.536	0.595
0.2	3	2	0.027	0.193	0.254	0.000	0.188	0.193	0.021	0.193	0.197	0.552	0.668	0.806	0.270	0.366	0.394	0.635	0.743	0.776	0.000	0.532	0.670
0.4	3	2	0.267	0.399	0.415	0.329	0.399	0.400	0.235	0.380	0.390	0.677	0.792	0.815	0.464	0.376	0.435	0.756	0.794	0.816	0.751	0.762	0.806
0.6	3	2	0.329	0.355	0.357	0.000	0.386	0.400	0.506	0.507	0.538	0.532	0.622	0.627	0.315	0.320	0.315	0.501	0.603	0.644	0.498	0.667	0.704
0.8	3	2	0.418	0.498	0.595	0.383	0.581	0.623	0.305	0.478	0.568	0.285	0.427	0.604	0.309	0.612	0.614	0.310	0.558	0.596	0.256	0.509	0.604

avg increase

			leid			infomap			walk			greedy			eig			paris			combo		
μ	t1	t2	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
0.2	2	1	0.020	0.056	0.086	0.001	0.036	0.059	0.023	0.060	0.101	0.271	0.474	0.566	0.058	0.098	0.122	0.249	0.396	0.489	0.139	0.312	0.438
0.4	2	1	0.086	0.137	0.155	0.149	0.196	0.223	0.065	0.117	0.146	0.236	0.350	0.384	0.110	0.153	0.162	0.280	0.378	0.418	0.328	0.492	0.577
0.6	2	1	0.138	0.203	0.224	0.050	0.207	0.264	0.143	0.214	0.236	0.198	0.297	0.336	0.126	0.174	0.164	0.231	0.313	0.359	0.186	0.278	0.319
0.8	2	1	0.212	0.347	0.411	0.155	0.248	0.321	0.203	0.326	0.417	0.180	0.316	0.404	0.217	0.335	0.394	0.192	0.345	0.428	0.162	0.296	0.378
0.2	3	2	0.001	0.057	0.094	0.000	0.027	0.071	0.001	0.041	0.082	0.302	0.473	0.606	0.086	0.130	0.154	0.286	0.454	0.534	0.000	0.249	0.405
0.4	3	2	0.088	0.143	0.169	0.106	0.164	0.190	0.069	0.117	0.151	0.286	0.416	0.474	0.126	0.151	0.176	0.293	0.414	0.456	0.285	0.437	0.541
0.6	3	2	0.103	0.129	0.137	0.000	0.076	0.148	0.100	0.135	0.147	0.225	0.303	0.322	0.126	0.154	0.151	0.247	0.315	0.339	0.188	0.234	0.261
0.8	3	2	0.195	0.347	0.405	0.154	0.249	0.309	0.197	0.333	0.396	0.174	0.310	0.398	0.201	0.358	0.424	0.218	0.383	0.446	0.180	0.331	0.412

Fig. 7 Heatmaps for the maximum increase (upper chart) and average increase (lower chart) of deception scores in 30-node synthetic networks for all community detection algorithms under budget constraints $\beta = \{1, 2, 3\}$

max increase

			leid			infomap			walk			greedy			eig			paris			combo		
μ	t1	t2	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
0.2	2	1	0.000	0.000	0.138	0.000	0.000	0.003	0.000	0.003	0.008	0.551	0.706	0.719	0.021	0.037	0.142	0.553	0.730	0.766	0.000	0.000	0.358
0.4	2	1	0.105	0.106	0.106	0.000	0.401	0.404	0.101	0.169	0.186	0.243	0.545	0.704	0.043	0.073	0.057	0.302	0.346	0.395	0.288	0.533	0.649
0.6	2	1	0.036	0.051	0.054	0.000	0.000	0.000	0.092	0.117	0.126	0.296	0.303	0.310	0.044	0.064	0.074	0.405	0.497	0.503	0.219	0.222	0.234
0.8	2	1	0.278	0.373	0.383	0.044	0.089	0.089	0.278	0.383	0.410	0.158	0.218	0.287	0.150	0.235	0.169	0.171	0.233	0.289	0.159	0.222	0.288
0.2	3	2	0.000	0.000	0.137	0.000	0.001	0.136	0.000	0.002	0.136	0.359	0.561	0.718	0.134	0.135	0.159	0.554	0.731	0.759	0.000	0.000	0.359
0.4	3	2	0.032	0.051	0.063	0.000	0.331	0.374	0.035	0.051	0.055	0.322	0.547	0.696	0.025	0.029	0.033	0.377	0.466	0.501	0.378	0.743	0.757
0.6	3	2	0.376	0.394	0.394	0.000	0.000	0.000	0.129	0.158	0.163	0.144	0.256	0.269	0.151	0.167	0.061	0.478	0.523	0.589	0.656	0.677	0.684
0.8	3	2	0.271	0.398	0.414	0.051	0.103	0.103	0.165	0.328	0.429	0.166	0.212	0.232	0.233	0.321	0.348	0.337	0.378	0.383	0.184	0.223	0.234

average increase

			leid			infomap			walk			greedy			eig			paris			combo		
μ	t1	t2	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
0.2	2	1	0.000	0.000	0.040	0.000	0.000	0.002	0.000	0.001	0.005	0.406	0.501	0.640	0.007	0.021	0.068	0.413	0.614	0.657	0.000	0.000	0.348
0.4	2	1	0.027	0.038	0.045	0.000	0.286	0.376	0.067	0.095	0.117	0.135	0.293	0.345	0.030	0.047	0.045	0.214	0.303	0.338	0.125	0.363	0.514
0.6	2	1	0.028	0.039	0.044	0.000	0.000	0.000	0.034	0.047	0.055	0.171	0.178	0.186	0.038	0.056	0.072	0.231	0.270	0.279	0.100	0.110	0.117
0.8	2	1	0.120	0.183	0.208	0.025	0.050	0.057	0.195	0.261	0.289	0.087	0.146	0.170	0.116	0.185	0.084	0.094	0.149	0.184	0.091	0.147	0.170
0.2	3	2	0.000	0.000	0.045	0.000	0.000	0.036	0.000	0.001	0.037	0.281	0.462	0.607	0.063	0.075	0.098	0.332	0.466	0.487	0.000	0.000	0.351
0.4	3	2	0.029	0.046	0.054	0.000	0.083	0.270	0.029	0.043	0.051	0.188	0.290	0.339	0.009	0.022	0.029	0.322	0.389	0.408	0.149	0.545	0.577
0.6	3	2	0.163	0.178	0.182	0.000	0.000	0.000	0.056	0.079	0.086	0.091	0.144	0.152	0.093	0.111	0.061	0.429	0.472	0.496	0.250	0.331	0.345
0.8	3	2	0.132	0.182	0.202	0.021	0.037	0.042	0.112	0.191	0.234	0.118	0.165	0.186	0.166	0.215	0.235	0.229	0.319	0.347	0.098	0.161	0.179

Fig. 8 Heatmaps for the maximum increase (upper chart) and average increase (lower chart) of deception scores in 50-node synthetic networks for all community detection algorithms under budget constraints $\beta = \{1, 2, 3\}$

inter-community additions, and inter-community deletions) by considering altogether the four ground truth communities and the best set of modifications allowing to reach the highest value of deception score. Across all algorithms and budgets, inter-community additions represent the majority of the editings, indicating that adding edges toward different communities is the most effective modification type for this network. Intra-community additions also represent a significant portion of the modifications but are less prevalent than inter-community additions. The pattern remains consistent across different budgets, though the proportion of inter-adds tends to slightly increase with higher budgets.

The comparison of the results reported in Figs. 5 and 6 suggests that, in general the type of modifications that allow to obtain the maximum possible deception higly depends on the type and characteristics of the network. In fact, recall that while the degree distribution of the karate network resemble that of a scale-free network, the dolphin network shows no clear degree distribution pattern and thus it is not possible to classify it.

Figures.7 and 8 both provide quantitative assessments of deception scores for synthetic 30-node and 50-node networks, respectively, by considering for each network the ground truth communities provided by the generator. Each

figure features two heatmaps that detail the maximum and average increases in deception scores obtained on the various detection algorithms in response to optimal modifications under budget constraints $\beta = 1, 2, 3$. Columns in each heatmap represent different detection algorithms, while rows outline network characteristics.

For both network sizes, the heatmaps are calculated as aggregates across all ground truth communities within networks in each category and benchmarked against initial deception values obtained before applying any modifications. In total, the 48 networks generated have between 4 to 7 ground truth communities depending on the specific parameters used (see Table 2). The maximum increase heatmap highlights the most effective modifications at each budget level, offering insights into the peak deception capability achieved. In contrast, the average increase heatmap provides a panoramic view of how deception scores have generally risen across all modifications and communities.

The data allows for an in-depth comparison of the resilience of various detection algorithms against deceptive modifications. A consistent trend observed is the increase in deception scores with respect to the budget of allowed modifications, except for the Eigenvectors algorithm, which shows non-convergence issues at the highest budget ($\beta = 3$) in both network scenarios. In detailed analysis, for the 30-node networks, algorithms such as Greedy, Paris, and Combo show the highest increases in both maximum and average deception scores, suggesting a higher susceptibility to deception strategies. These algorithms typically start with the lowest initial deception scores, which occasionally reveal the ground truth community completely, making them more prone to effective deception. For the 50-node networks, Paris exhibits the most significant increases in deception scores, indicating its particular vulnerability. Greedy and Combo also show high deception increases but tend to start with lower initial deception values for some communities, echoing similar patterns observed in the 30-node assessments.

As μ increases, indicating more inter-community edges, the deception scores generally increase. This trend is consistent across both 30-node and 50-node networks, indicating that higher mixing leads to more opportunities for deceptive modifications to disrupt community structures. However, there are exceptions for the Greedy, Paris, and partially Combo algorithms, where an inverse relationship is observed. Networks with higher t_1 and t_2 values often exhibit higher deception scores in the 30-node networks, particularly in algorithms like Greedy and Paris. However, we cannot observe the same strong tendency in 50-node networks. This suggests that small networks resembling random networks are more susceptible to deceptive strategies.

Overall, these analyses underscore which algorithms and types of networks are more susceptible to deceptive strategies and how they respond to increasing complexities introduced by higher budget modifications. This comprehensive overview helps identify the relative robustness of each algorithm, enhancing our understanding of their performance across different network characteristics and deceptive scenarios.

Figures 9 and 10 present a stacked bar chart for the Leiden, Infomap, Eigenvector and Paris community detection algorithms, showing the types and proportions of modifications applied to synthetic networks of 30 and 50 nodes respectively, to achieve the highest deception scores under budget $\beta \in 1, 2, 3$. These modifications are aggregated across all ground truth communities within each network category. The values of μ, t_1, and t_2, which characterize the networks, are reported at the bottom of each bar, emphasizing the influence of these parameters on the network's structure and the subsequent optimal deception strategies. The analysis across both figures indicates that the type of best modifications to deceive each algorithm varies significantly depending on the network characteristics and budget levels, showcasing adaptive strategies based on the allowed number of network modifications. For both the 30-node and 50-node networks, intra-community deletions emerge as a prevalent strategy among most algorithms, particularly at lower budget levels and in networks with lower values of the mixing parameter μ. This suggests that removing internal community links is an effective initial strategy for increasing deception, targeting communities with dense intra-links to confuse the detection algorithms. Conversely, the addition of intra-community edges is preferred in networks with higher mixing parameter, indicating a strategic preference to increase internal connectivity as a means of deception. This strategy inversely correlates with the values of t_1 and t_2. Particularly, strategies to deceive algorithms like Eigenvectors and Paris consistently show a strong preference for inter-community additions across both network sizes. We observed similar trends for the other community detection algorithms as well (complete charts are provided in the Appendix). The data from these charts not only provide insights into the modifications that are preferable to deceive each deception algorithm but also highlight deception algorithms vulnerabilities and the effectiveness of different deceptive strategies tailored to the network's structural characteristics. This comprehensive analysis aids in understanding which algorithms are more susceptible to specific types of modifications and how they might be instructed against such deceptive tactics in practical applications.

4.5 Potential real-world applications

The BHC dataset and the findings from our benchmark analysis have several potential real-world applications, particularly in enhancing privacy and developing more robust community detection algorithms.

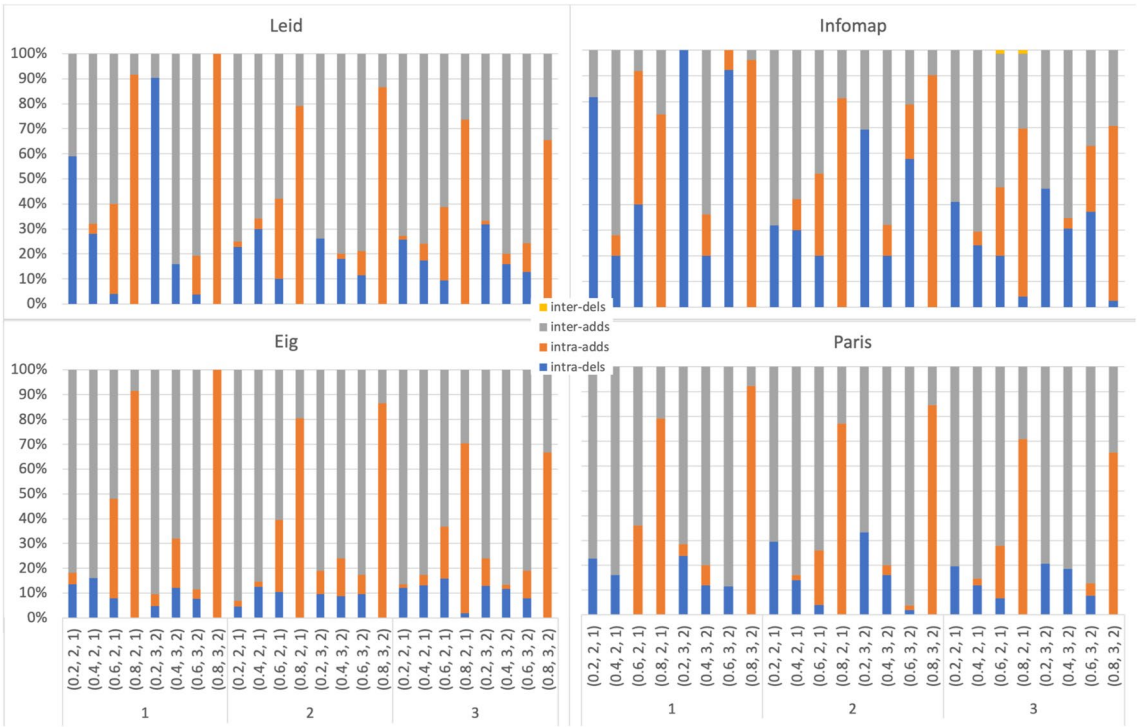

Fig. 9 Distribution of modification types for community detection algorithms and network characteristics by budget values over 30-node synthetic networks

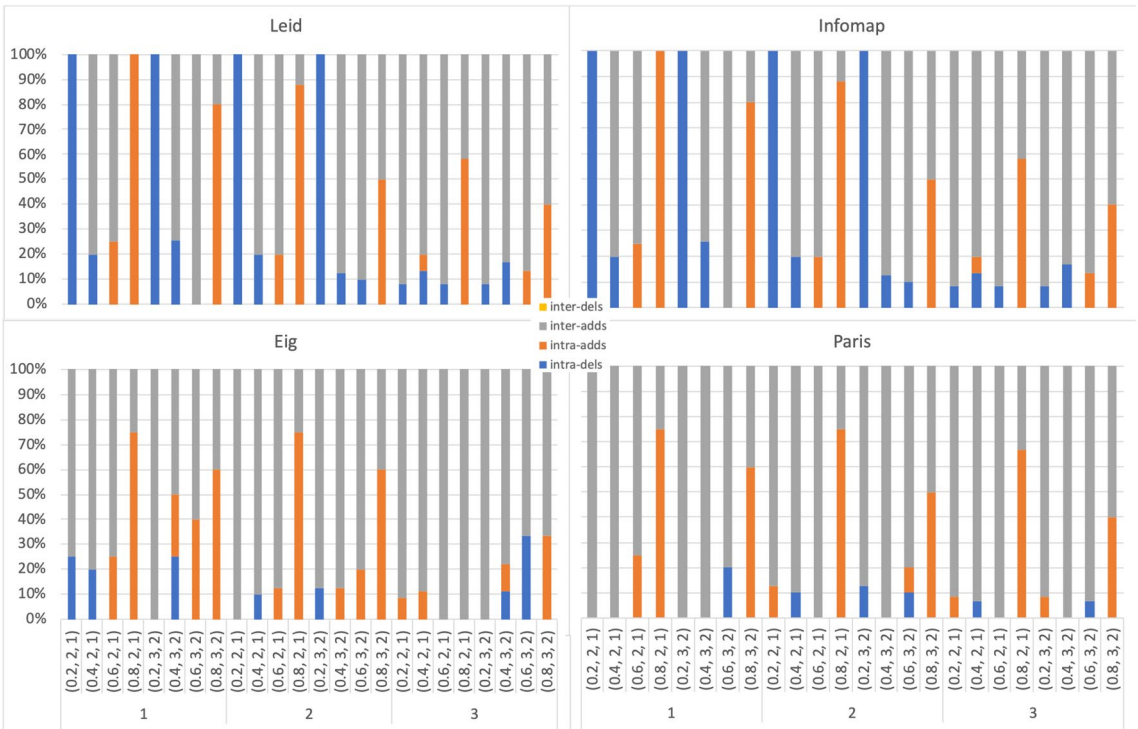

Fig. 10 Distribution of modification types for community detection algorithms and network characteristics by budget values over 50-node synthetic networks

Enhancing privacy in sensitive networks By understanding how to effectively hide communities within networks, law enforcement agencies can better protect the identities of undercover operatives or sensitive operations from being detected by adversarial analysis. Furthermore, companies can use insights from the BHC dataset to protect sensitive information about their internal structures, such as R &D teams or strategic partnerships, from being revealed through social network analysis.

Developing robust community detection algorithms Social media companies can leverage the BHC dataset to test and enhance their community detection algorithms, making them more resilient to deceptive practices. This can help in better identifying and mitigating misinformation campaigns or coordinated inauthentic behavior. Furthermore, in cybersecurity detecting and understanding deceptive tactics used in network structures can aid in identifying malicious activities such as botnets or coordinated cyber-attacks.

Interdisciplinary applications The insights gained from studying community deception using the BHC dataset extend beyond the realm of social network analysis and can be applied to various interdisciplinary domains. As an example in epidemiology understanding how diseases spread within communities is critical for controlling outbreaks. By applying community deception insights, epidemiologists can develop strategies to obscure the true structure of transmission networks, thereby protecting sensitive information about infected individuals and preventing stigmatization. This can also aid in designing interventions that are less predictable and more effective in containing the spread. Furthermore, political scientists can apply detection and deception techniques to study the formation and influence of political groups within social networks. By understanding how communities form and how they can be deceptively modified, researchers can gain insights into the dynamics of political movements and develop strategies to counteract misinformation and manipulation in political campaigns. Finally, marketers can use community detection to identify influential groups within consumer networks and tailor their strategies accordingly. Insights from community deception can help in designing marketing campaigns that are less susceptible to competitive interference and more effective in reaching target audiences.

4.6 Limitations

While the Better Hide Communities (BHC) benchmark dataset provides a robust framework for evaluating community deception algorithms, several limitations need to be acknowledged. First of all, the current study focuses exclusively on undirected and unweighted networks. Real-world networks can involve directed and/or weighted edges, which can significantly impact the performance and evaluation of community deception algorithms. Secondly, this benchmark is limited to deception obtained by edge additions and deletions only. However, node-based modifications (such as node addition, deletion, or reassignment) and modifications to node attributes can also play a critical role in community deception. Finally, the synthetic networks generated for this study follow specific configurations (e.g., mixing parameter μ, degree distribution exponent t_1, and community size distribution exponent t_2). Although these configurations cover a range of typical scenarios, they may not encompass all possible real-world network structures.

5 Conclusions and future work

The introduction of the Better Hide Communities (BHC) dataset represents a significant advancement in the field of network security, particularly in the niche but increasingly important area of community deception. This benchmark dataset is poised to standardize evaluations across this domain, providing researchers and practitioners with a robust tool to assess the efficacy of different community deception strategies under controlled and consistent conditions.

Moreover, by providing detailed insights into how different detection algorithms perform under various scenarios, it helps identify potential vulnerabilities and strengths within current approaches. This not only aids in refining existing techniques but also in designing new algorithms that are more resilient against deception.

The BHC dataset has set the stage for numerous future investigations. One immediate area of potential research is the enrichment of the dataset to include more complex network scenarios, such as those involving direct networks or overlapping communities, as well as dynamic networks. Additionally, exploring the integration of machine learning techniques to predict and optimize deception strategies could further enhance the capabilities of community deception methodologies.

Appendix A

Figures 11 and 12 present a stacked bar chart for each community detection algorithm, showing the types and proportions of modifications applied to synthetic networks of 30 and 50 nodes respectively, to achieve the highest deception scores under budget $\beta \in 1, 2, 3$. These modifications are

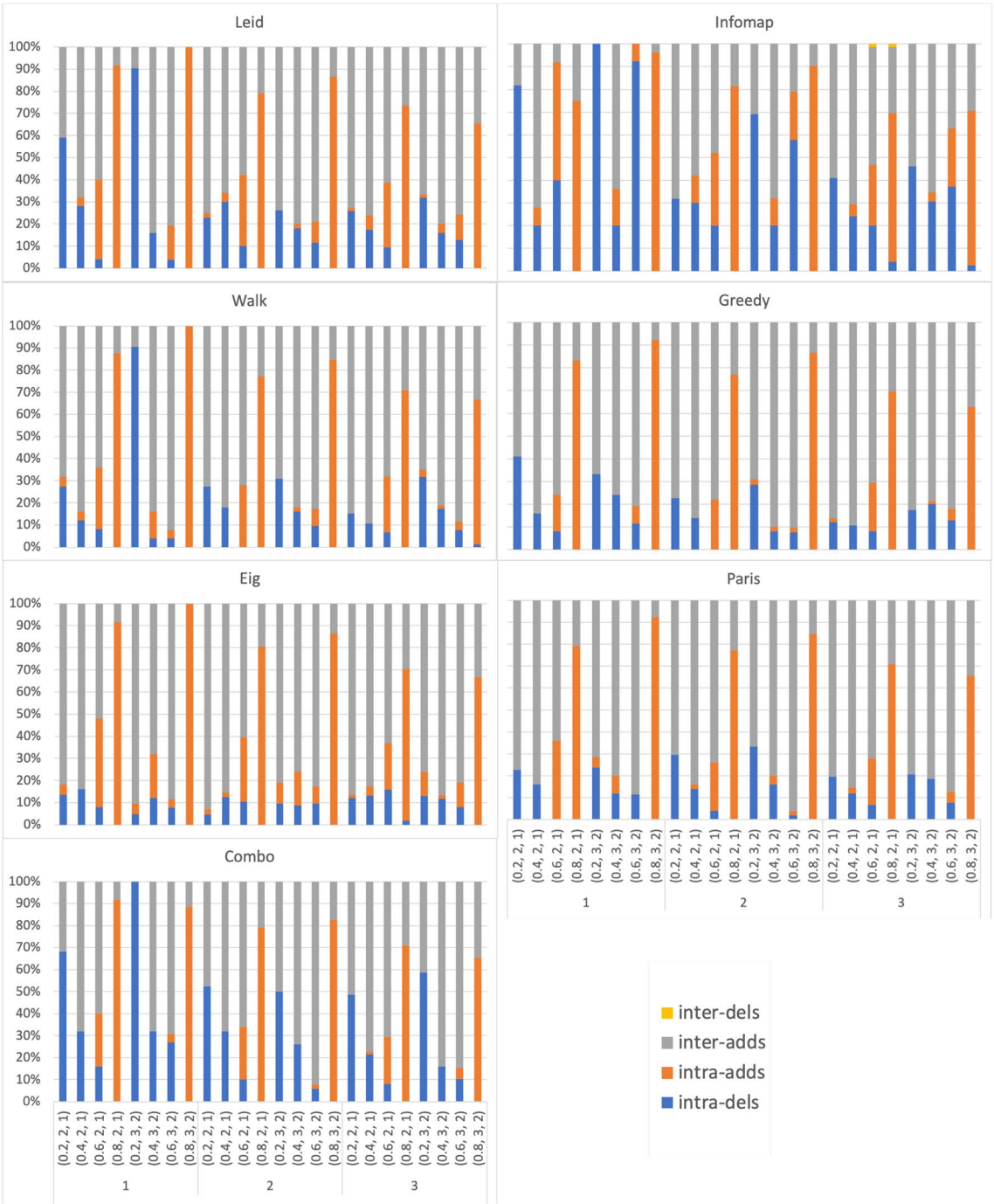

Fig. 11 Distribution of modification types for community detection algorithms and network characteristics by budget values over 30-node synthetic networks

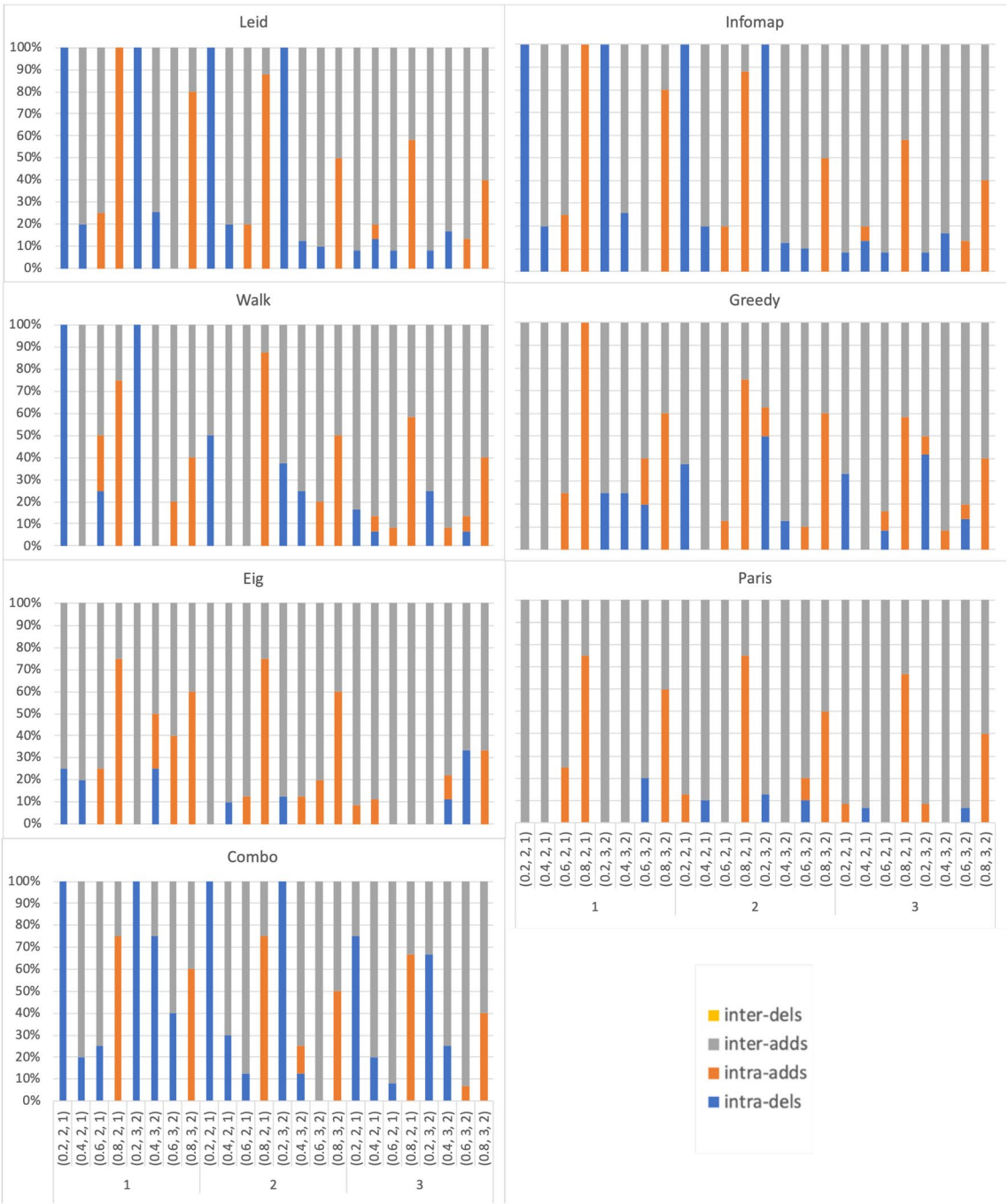

Fig. 12 Distribution of modification types for community detection algorithms and network characteristics by budget values over 50-node synthetic networks

aggregated across all ground truth communities within each network category. The values of μ, t_1, and t_2, which characterize the networks, are reported at the bottom of each bar.

Acknowledgements We acknowledge the support of the PRIN PNRR project DISTORT Prot. P2022KHTX7, CUP H53D23008170001, the PRIN project HypeKG Prot. 2022Y34XNM, CUP H53D23003710006 under the MUR program funded by the European Union - NextGenerationEU; and the PNRR project FAIR - Future AI Research (PE00000013), Spoke 9 WP 9.2 - Green-aware AI, under the NRRP MUR program funded by the NextGenerationEU.

Funding Open access funding provided by Università della Calabria within the CRUI-CARE Agreement.

References

Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. J Stat Mech-Theory E 10:P10008

Bonald T, Charpentier B, Galland A, Hollocou A (2018) Hierarchical graph clustering using node pair sampling. arxiv:abs/1806.01664

Cazabet R, Rossetti G, Milli L (2022) CDlib: a python library to extract, compare and evaluate communities from complex networks (extended abstract). In: Proceedings of MARAMI , CEUR-WS.org

Chakraborty T, Srinivasan S, Ganguly N, Mukherjee A, Bhowmick S (2016) Permanence and community structure in complex networks. ACM TKDD 11(2):1–34

Chen J, Chen L, Chen Y, Zhao M, Shanqing Yu, Xuan Q, Yang X (2019) Ga-based q-attack on community detection. IEEE Trans Comput Soc Syst 6(3):491–503

Chen J, Chen Y, Chen L, Zhao M, Xuan Q (2020) Multiscale evolutionary perturbation attack on community detection. IEEE Trans Comput Soc Syst 8(1):62–75

Chen X, Jiang Z, Li H, Ma J, Philip SY (2021) Community hiding by link perturbation in social networks. IEEE Trans Comput Soc Syst 8(3):704–715

Chen C, Jiang Z, Ma J (2022) Privacy protection for marginal-sensitive community individuals against adversarial community detection attacks. IEEE Trans Comput Soc Syst 11(1):782–794

Clauset A, Newman MEJ, Moore C (2004) Finding community structure in very large networks. Phys Rev E 70(6):066111

Fionda V, Palopoli L, Panni S, Rombo SE (2008) Protein-protein interaction network querying by a "focus and zoom" approach. In BIRD, CCIS, vol. 13. pp 331–346

Fionda V, Palopoli L, Panni S, Rombo SE (2009) A technique to search for functional similarities in protein-protein interaction networks. Int J Data Min Bioinform 3(4):431–453

Fionda V, Gutierrez C, Pirrò G (2016) Building knowledge maps of Web graphs. Artif Intell 239:143–167

Fionda V, Pirrò G (2018) Community deception or: how to stop fearing community detection algorithms. IEEE Trans Knowl Data Eng 30(4):660–673

Fionda V, Pirrò G (2022) Community deception in networks: where we are and where we should go. In: Proceedings of Complex Networks & Their Applications X, Springer International Publishing, Cham, pp 144–155

Fionda V, Madi SA, Pirrò G (2022) Community deception: from undirected to directed networks. Soc Netw Anal Min 12(1):74

Fionda V (2023) Better hide communities: benchmarking community deception algorithms. In: Proceedings of Complex Networks & Their Applications, Springer International Publishing, pp 378–387

Fionda V, Pirrò G (2024) Community deception in attributed networks. IEEE Trans Comput Soc Syst 11(1):228–237

Lancichinetti A, Fortunato S, Radicchi F (2008) Benchmark graphs for testing community detection algorithms. Phys Rev E 78(4):046110

Li J, Zhang H, Han Z, Rong Y, Cheng H, Huang J (2020) Adversarial attack on community detection by hiding individuals. In: The Web Conference, pp 917–927

Liu Y, Liu J, Zhang Z, Zhu L, Li A(2019) REM: from structural entropy to community structure deception. Adv Neural Inf Process Syst, 32

Liu X, Fu L, Wang X, Hopcroft JE (2021) Prohico: a probabilistic framework to hide communities in large networks. In: IEEE INFOCOM

Liu D, Chang Z, Yang G, Chen E (2022) Hiding ourselves from community detection through genetic algorithms. J Inf Sci 614:123–137

Liu D, Yang G, Wang Y, Jin H, Chen E (2022) How to protect ourselves from overlapping community detection in social networks. IEEE Trans Big Data 8(4):894–904

Liu D, Chang Z, Yang G, Chen E (2022) Community hiding using a graph autoencoder. Knowl Based Syst 253:109495

Lusseau D, Schneider K, Boisseau OJ, Haase P, Slooten E, Dawson SM (2003) The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. Behav Ecol Sociobiol 54(4):396–405

Madi SA, Pirrò G (2023) Node-centric community deception based on safeness. IEEE Trans Comput Soc Syst 11(2):2955–2965

Magelinski T, Bartulovic M, Carley KM (2021) Measuring node contribution to community structure with modularity vitality. IEEE Trans Netw Sci Eng 8(1):707–723

Mittal S, Sengupta D, Chakraborty T (2021) Hide and seek: outwitting community detection algorithms. IEEE Trans Comput Soc Syst 8(4):799–808

Nagaraja S (2010) The impact of unlinkability on adversarial community detection: effects and countermeasures. In: PETS, pp 253–272

Newman MEJ (2006) Modularity and community structure in networks. PNAS 103(23):8577–8582

Newman MEJ (2006) Finding community structure in networks using the eigenvectors of matrices. Phys Rev E 74(3):036104

Pirrò G (2023) Community deception from a node-centric perspective. IEEE Trans Netw Sci Eng 11(1):969–981

Pons P, Latapy M (2006) Computing communities in large networks using random walks. J Graph Algorithms Appl 10(2):191–218

Revelle M, Domeniconi C, Sweeney M, Johri A (2015) Finding community topics and membership in graphs. In: ECML/PKDD, pp 625–640

Rosvall M, Bergstrom CT (2008) Maps of random walks on complex networks reveal community structure. Proc Natl Acad Sci USA 105(4):1118–1123

Sobolevsky S, Campari R, Belyi A, Ratti C (2014) General optimization technique for high-quality community detection in complex networks. Phys Rev E 90(1):012811

Traag VA, Waltman L, van Eck NJ (2018) From Louvain to Leiden: guaranteeing well-connected communities. arXiv:abs/1810.08473

Waniek M, Michalak TP, Wooldridge MJ, Rahwan T (2018) Hiding individuals and communities in a social network. Nature Human Behav 2(2):139–147

Yang J, McAuley J, Leskovec J (2013) Community Detection in Networks with Node Attributes. In: ICMD, pp 1151–1156

Yang H, Chen L, Cheng F, Qiu J, Zhang L (2023) Lsha: a local structure-based community detection attack heuristic approach. IEEE Trans Comput Soc Syst 11(2):2966–2978

Zachary WW (1977) An information flow model for conflict and fission in small groups. J Anthropol Res 33:452–473

Zhang C, Fu L, Ding J, Cao X, Long F, Wang X, Zhou L, Zhang J, Zhou C (2023) Community deception in large networks: through the lens of laplacian spectrum. IEEE Trans Comput Soc Syst 11(2):2057–2069

Zhao J, Wang Z, Cao J, Cheong KH (2023) A self-adaptive evolutionary deception framework for community structure. IEEE Trans Syst Man Cybern 53(8):4954–4967

Zhao J, Cheong KH (2023) Obfuscating community structure in complex network with evolutionary divide-and-conquer strategy. IEEE Trans Evol Comput 27(6):1926–1940