Preserving Clustering Coefficients in Social Network Anonymization

Social Network Analysis for Computer Scientists — Course paper

Sadaf Esmaeili Rad
s.esmaeili.rad@umail.leidenuniv.nl
LIACS, Leiden University

Sepehr Moghiseh
s.moghiseh@umail.leidenuniv.nl
LIACS, Leiden University

ABSTRACT

This paper focuses on protecting privacy in social network data while keeping important structural properties like the clustering coefficient unchanged as much as possible. We propose a method that uses perturbation to anonymize the network by carefully adding and removing edges. Instead of random changes, our approach prioritizes preserving triangles, which are key for understanding community structures. This helps maintain the clustering coefficient, making the anonymized data still useful for analysis. We tested our method on several datasets and measured its impact on metrics like clustering coefficients, entropy, and graph diameter. The results show that our method can successfully balance privacy and utility, allowing meaningful social network analysis even after anonymization.

KEYWORDS

Social network analysis, anonymization, structural perturbation, re-identification risk, clustering coefficient.

1 INTRODUCTION

Privacy is a big concern when working with social network data because these datasets often contain sensitive information about people's relationships. Social networks are usually represented as graphs, where nodes represent individuals and edges show their connections. These graphs are useful for studying how people interact and for analyzing communities, but they come with risks. Even after anonymizing the data, attackers can sometimes re-identify individuals by using the structure of the network.

One common method to anonymize social networks is naive anonymization, which removes names or other identifiers from the graph. However, this approach often fails because the structure of the graph stays the same. Attackers can use patterns in the graph to link individuals back to external knowledge. For example, unique configurations of connections can act like a fingerprint, allowing attackers to figure out who a person is.[1]

In Figure 1, a simple example of naive anonymization is illustrated. This process involves removing identifiers from the graph, but the structure of the network remains unchanged, which can still lead to re-identification of individuals through patterns in the data.

Another method is perturbation, which changes the structure of the graph by randomly adding or removing edges. While this can make it harder for attackers to recognize patterns, it also has disadvantageous. Random changes can affect important properties, like the clustering coefficient. The clustering coefficient measures how connected nodes are within small groups, and it's very important for analyzing communities. When this information is lost, the graph becomes less useful for analysis.

It is also important to preserve triangles, since they reflect more basic social group interactions and community structures. Triangles very often represent tight connections of friend groups, family circles, or collaborative teams. Keeping these structures intact during the anonymization process helps preserve local cohesion and clustering characteristics of the network, important for many analytical tasks. Furthermore, this preservation of relationships allows the researchers to perform meaningful community detection and analyze social behavior while preserving the privacy of the users. This presents a balanced solution where the protection of privacy and the usefulness of the dataset are maximized.

In this work, we propose a new method to address these problems. Instead of making random changes, we focus on preserving triangles in the graph. Triangles are groups of three nodes that are all connected to each other. By carefully redistributing edges to keep these triangles intact, our method helps maintain the clustering coefficient while also protecting privacy.

Our approach is designed to balance privacy and utility. By keeping key structures in the graph, we ensure the data remains useful for analyzing communities, while reducing the risk of re-identification.

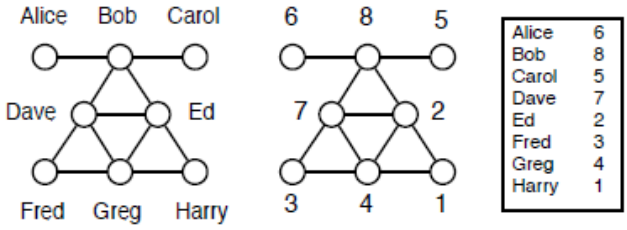


Figure 1: Naive Anonymization Example

2 RELATED WORK

Backstrom et al[2]. studied attacks on naively anonymized social networks, focusing on an 'active attack' where an adversary adds a distinct subgraph before anonymization to re-identify targets in the published graph. They provided algorithms to construct such subgraphs with high distinguishability.

Most prior anonymization research[3] focuses on tabular data, such as k-anonymity, l-diversity, and t-closeness [4], which are unsuitable for preserving graph structures. Extensions to network data, like interactive privacy mechanisms and synthetic data generation, offer strong privacy guarantees but limit usability for tasks like

visualization and clustering.

Advanced privacy-preserving techniques, such as *l-diversity* and *differential privacy*, have been proposed to address the limitations of earlier methods like k-anonymity. *l-Diversity* builds on k-anonymity by requiring that, within any group of indistinguishable records, there is sufficient diversity in sensitive attributes. This reduces the likelihood of attribute disclosure, where attackers could deduce sensitive information even if they cannot precisely re-identify individuals. On the other hand, *differential privacy* takes a probabilistic approach by carefully adding noise to the data or the results of queries on the data. This makes sure that adding or removing any individual has very little effect on the dataset's overall results. thereby offering strong privacy guarantees against a broad range of potential attacks. Both techniques have significantly contributed to reducing the risk of re-identification.

Our work builds on these approaches by proposing a perturbation technique that prioritizes preserving structural metrics, such as the clustering coefficient, while reducing re-identification risks. By focusing on triangles, we achieve a better balance between privacy and utility, allowing meaningful community analyses even in anonymized social networks.

3 PRELIMINARIES

This section introduces key concepts and formal definitions essential for understanding the risks of re-identification in anonymized social networks. We focus on candidate sets, adversarial queries, and vertex refinement.

3.1 Social Network Graphs and Re-identification Risks

A social network can be represented as an undirected graph $G = (V, E)$, where: - V is the set of nodes (vertices), each representing an individual. - E is the set of edges, where each edge $(u, v) \in E$ indicates a connection or relationship between two nodes u and v, when the graph is undirected. This graph structure provides a foundation for analyzing relationships within the network and identifying patterns that may pose re-identification risks. An adversary with partial knowledge of the network may exploit structural patterns to re-identify individuals in anonymized graphs using candidate sets and refinement techniques.

3.2 Candidate Sets and Vertex Refinement

In the context of social network anonymization, the *candidate set* for a target node x, denoted $\text{cand}_Q(x)$, consists of nodes in the anonymized graph G' that exhibit structural characteristics matching those of x. Formally, the candidate set is defined as:[1]

$$\text{cand}_Q(x) = \{y \in \mathrm{V} \mid Q(x) = Q(y)\}$$

Here, $Q(x)$ represents a set of structural attributes of x (degree or neighborhood properties), and V shows the set of nodes in G'. Nodes y that satisfy $Q(y) = Q(x)$ are considered potential matches for x. To limit re-identification certainty, the confidence level $C_{Q,x}[y]$ for mapping y to x is limited as follows:

$$\forall x \in V,\ \forall y \in \text{cand}_Q(x) :\ C_{Q,x}[y] \leq \frac{1}{k}$$

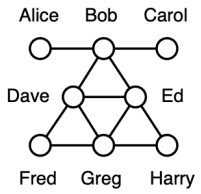
The formula ensures that each node x in the original graph has at least k possible matches y in the anonymized graph, with each match being equally possible.

An adversary can iteratively refine the candidate set through *vertex refinement*, narrowing down potential matches using neighborhood information. At each iteration i, the neighborhood attributes $H_i(x)$ for a node x are computed based on the attributes of its neighbors $z_1, z_2, \ldots, z_m$:

$$H_i(x) = \{H_{i-1}(z_1), H_{i-1}(z_2), \ldots, H_{i-1}(z_m)\}$$

Here, $H_0(x)$ represents the basic attributes of node x, such as its label or degree, while $H_1(x)$ includes the attributes of x's neighbors. Each iteration $H_i(x)$ builds on the previous one by adding more detailed neighborhood information.

This refinement process iteratively differentiates nodes in $\text{cand}_Q(x)$, which can sometimes lead to identifying a unique candidate. During this process, nodes that remain indistinguishable after multiple iterations form 'equivalence classes'. An equivalence class is a group of nodes that cannot be distinguished based on the refinement process and share identical structural characteristics up to a given iteration. Larger equivalence classes indicate stronger anonymization since many nodes share the same structural attributes, making it harder for an adversary to identify a single node.



(a) graph

Node ID	$\mathcal{H}_0$	$\mathcal{H}_1$	$\mathcal{H}_2$
Alice	ϵ	1	$\{4\}$
Bob	ϵ	4	$\{1,1,4,4\}$
Carol	ϵ	1	$\{4\}$
Dave	ϵ	4	$\{2,4,4,4\}$
Ed	ϵ	4	$\{2,4,4,4\}$
Fred	ϵ	2	$\{4,4\}$
Greg	ϵ	4	$\{2,2,4,4\}$
Harry	ϵ	2	$\{4,4\}$

(b) vertex refinements

Equivalence Relation	Equivalence Classes
$\equiv_{\mathcal{H}_0}$	$\{A,B,C,D,E,F,G,H\}$
$\equiv_{\mathcal{H}_1}$	$\{A,C\}\quad\{B,D,E,G\}\quad\{F,H\}$
$\equiv_{\mathcal{H}_2}$	$\{A,C\}\{B\}\{D,E\}\{G\}\{F,H\}$
$\equiv_A$	$\{A,C\}\{B\}\{D,E\}\{G\}\{F,H\}$

(c) equivalence classes

Figure 2: (a) A sample graph; (b) Vertex refinement queries H_1, H_2, and H_3 computed for each individual in the graph; (c) The groups of nodes (equivalence classes) created based on vertex refinement. Larger equivalence classes indicate stronger anonymization by reducing the ability to identify individual nodes.

3.3 Triangles and Clustering Coefficient

Triangles are subgraphs consisting of three nodes where each node is connected to the other two, forming a closed loop. In social networks, the presence of triangles indicates strong community connections, as they represent tightly knit groups.

The clustering coefficient is a measure of the degree to which nodes in a graph tend to cluster together. For a node v with m neighbors, the clustering coefficient $C(v)$ is defined as:

$$C(v) = \frac{\text{number of triangles connected to } v}{\binom{m}{2}}$$

where $\binom{m}{2}$ equals the total number of possible edges between v's neighbors.

The average clustering coefficient for the graph is the mean value of $C(v)$ over all nodes in the network. This metric is essential for analyzing the cohesion within social networks and identifying community structures. In this work, we focus on preserving the clustering coefficient during perturbation to maintain data utility and ensure meaningful community analysis.

4 APPROACH

Our method anonymizes social networks by carefully modifying the graph's structure while preserving important properties like the clustering coefficient. The process focuses on adding and removing edges during the same iteration for each node. By ensuring that these changes minimally destroy the graph's triangles, we maintain the clustering coefficient, which is essential for analyzing community structures.

According to our method, it consists of the following steps:

4.1 Triangle Identification

For each node, we identify its neighbors and non-neighbors. Non-neighbors are sampled because connecting a node to a non-neighbor can form new triangles, which helps preserve the clustering coefficient. Neighbors are checked as they are the only candidates for edge removal. This distinction ensures that edge additions and removals target specific parts of the graph, balancing privacy and utility without destroying its structure.

4.2 Simultaneous Edge Addition and Removal

Edge Addition: Non-neighboring nodes are sampled as candidates for adding new edges, but not all candidates are selected. Edges are added *probabilistically*, meaning there is a random chance of adding an edge based on a predefined probability p_{add}. For instance, if $p_{add} = 0.1$, there is a 10% chance for each candidate edge to be added.

This probabilistic approach introduces randomness into the graph perturbation process, making the modifications less predictable to attackers. At the same time, edges are only added if they form new triangles, which helps maintain the clustering coefficient and ensures the graph remains useful for community-based analyses.

Edge Removal: Neighboring nodes are sampled, and edges are removed probabilistically, but only if their removal does not ruin existing triangles. This preserves the clustering coefficient while achieving the anonymization goal.

4.3 Iterative Perturbation

The edge addition and removal steps are performed simultaneously for each node in the graph. This iterative process continues across the graph until the desired level of perturbation is achieved. By adding and removing edges in a balanced way, the graph remains structurally similar to the original while being anonymized.

The following pseudocode outlines the steps of the proposed algorithm.

The code can be accessed through github.[5]

Algorithm 1 Perturbation with Clustering Preservation

Input: Graph G, Probability for Adding Edges p_{add}, Probability for Removing Edges p_{remove}, Maximum Candidates

```
Initialize lists for added edges ins_connection ← [] and removed edges del_connection ← []
for each node v ∈ G.nodes do
    Identify neighbors N(v) and non-neighbors N'(v) of v
    Randomly sample up to m non-neighbors from N'(v)
    if random probability < p_add then
        for each sampled non-neighbor u do
            Find shared neighbors S = N(v) ∩ N(u)
            if |S| > 0 then
                Add edge (v, u) to G
                Append (v, u) to ins_connection
                break (Add only one edge per iteration for v)
            end if
        end for
    end if
    Randomly sample up to m neighbors from N(v)
    if random probability < p_remove then
        for each sampled neighbor u do
            Find shared neighbors S = N(v) ∩ N(u)
            if |S| > 1 then
                Remove edge (v, u) from G
                Append (v, u) to del_connection
                break (Remove only one edge per iteration for v)
            end if
        end for
    end if
end for
return del_connection, ins_connection
```

5 DATASETS

- **The Wiki-Vote Dataset** is derived from the voting behavior of Wikipedia users during administrator elections. Each node represents a user, and a directed edge indicates that one user voted for another. This dataset is made available through the Stanford Large Network Dataset Collection (SNAP). It contains 7,115 nodes and 103,689 edges. This

dataset is useful for studying influence and decision-making within communities.[6]

- **The Soc-Epinions1 Dataset** represents the trust relationships among users of Epinions.com, an online platform for product reviews. Nodes are users, and directed edges indicate trust relationships, where one user trusts another. This dataset, also from SNAP, consists of 75,879 nodes and 508,837 edges. It is ideal for analyzing trust dynamics and reputation systems in online social networks.[7]
- **The Email-Eu-core Dataset** is derived from email communications within a large European research institution. Each node corresponds to an email address, and a directed edge indicates that one person sent at least one email to another. This dataset includes 1,005 nodes and 25,571 edges. It is well-suited for studying communication patterns and the structure of organizational networks.[8]
- **The Facebook Combined Dataset** contains a combined network of Facebook users, where each node represents a user, and an edge indicates a friendship between two users. The dataset includes 4,039 nodes and 88,234 edges. It is useful for studying social interactions, community structures, and how users are connected in online social networks.[9]
- **The p2p-Gnutella08 Dataset** is a representation of the Gnutella peer-to-peer network. Each node in the dataset represents a computer (peer) in the network, and a directed edge indicates a connection between two peers. This dataset, made available through the Stanford Large Network Dataset Collection (SNAP), contains 6,301 nodes and 20,777 edges. It is useful for studying the structure and behavior of decentralized networks, including peer communication and network resilience.[10]

6 EXPERIMENTS

In our experiments, we worked with five datasets: Wiki-Vote, Email-Eu-core, Soc-Epinions1,Facebook Combined and p2p-Gnutella08 . For each dataset, we applied three levels of perturbation: 5%, 10%, and 100%. We evaluated the effects of these perturbations on the clustering coefficient, clustering coefficient entropy, and graph diameter to understand how the structure of the graphs changed. We also looked at how the community structures were affected by comparing partitions before and after perturbation using the normalized mutual information score. The results of all these evaluations are presented in the following sections to show how well our method balances privacy and preserving important graph properties.

6.1 Clustering Coefficient under graph perturbation

The clustering coefficient was evaluated across five datasets, with perturbations applied at 5%, 10%, and 100%. The objective was to observe how these changes in the graph structure affected the clustering coefficient and the overall structure of the graphs.

The method used aimed to preserve triangles in the graph, which are important for maintaining the clustering coefficient. This is why the clustering coefficient did not change much with 5% and 10% perturbations.

For example, in the Wiki-Vote dataset, the clustering coefficient increased slightly, which was expected since the method tried to keep the triangles.

In the Soc-Epinions1 dataset, the clustering coefficient remained almost unchanged, which suggests that trust-based networks are more stable and less affected by small changes in the graph.

For the Facebook Combined dataset, the clustering coefficient dropped more significantly at higher perturbation levels, especially at 100%. This was expected, as social networks like Facebook tend to have tighter connections, and larger changes in the graph structure are more likely to disrupt the clustering.

The Email-Eu-core and p2p-Gnutella08 datasets showed minimal changes in the clustering coefficient, suggesting that these networks are not as dependent on highly connected communities, so they are less sensitive to changes.

Table 1: Clustering Coefficient for Different Datasets.

Dataset	Perturbation Level			
	0%	**5%**	**10%**	**100%**
Wiki-Vote	0.081	0.086	0.089	0.147
Email-Eu-core	0.399	0.402	0.394	0.400
Soc-Epinions1	0.1101	0.1102	0.1099	0.1090
facebook_*combined*	0.3027	0.3020	0.300	0.284
p2p-Gnutella08	0.0054	0.0054	0.0059	0.0077

We also visualized the structure of triangles before and after perturbation, on figures below you can observe the visualisation for Wiki-Vote dataset as an example.

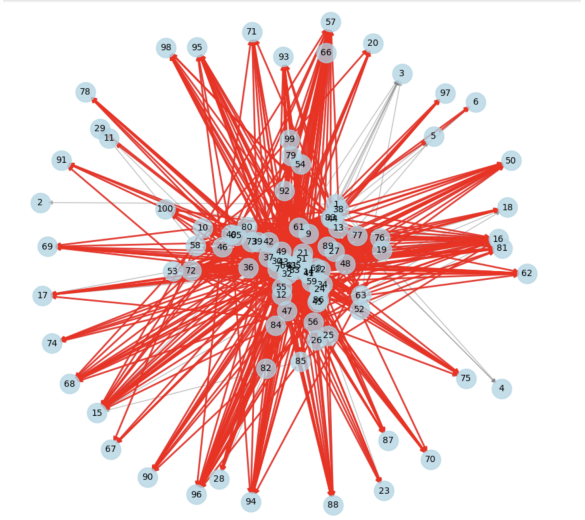


Figure 3: Original Graph for the Wiki-Vote Dataset: Visualization of the first 100 nodes before perturbation, highlighting the triangles in red. The clustering coefficient is 0.1409.

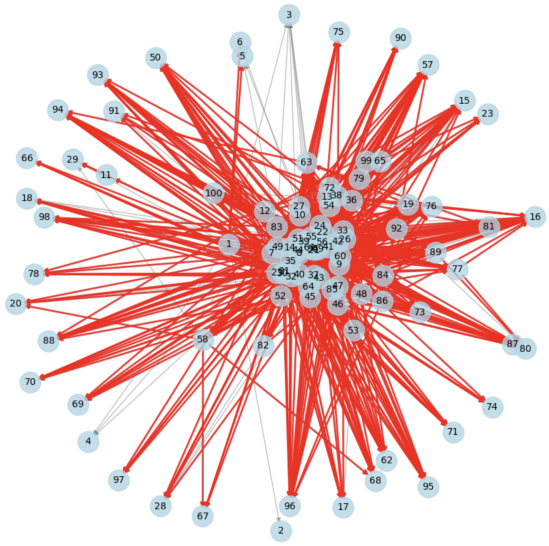


Figure 4: Graph After 10% Perturbation for the Wiki-Vote Dataset: Visualization of the first 100 nodes after edge modifications. The clustering coefficient increases to 0.1542, demonstrating the preservation and redistribution of triangles.

Comparing the graphs, indicates that the number of triangles did not change that much, just where the triangles located and the distribution changed, it shows that the perturbation preserved the clustering coefficient, while provided anonymity to the network.

6.2 Evaluating Graph structure through diameter

The diameter of a graph measures the longest shortest path between two nodes. It shows how far apart the most distant nodes are from each other in terms of the number of connections. A bigger diameter means the graph is more spread out, while a smaller diameter means the graph is more tightly connected.

For the Wiki-Vote dataset, the diameter stayed at 7 across all levels of change. This shows that the method worked well to keep the overall structure of the graph even with changes made to it. The method preserved the triangles, which helped maintain the overall connectivity.

In the Email-Eu-core dataset, the diameter stayed at 7 before and after the 5% and 10% changes. However, when the changes reached 100%, the diameter dropped to 6. This suggests that while smaller changes did not affect the diameter much, bigger changes caused parts of the graph to become less connected, reducing the diameter.

For the Soc-Epinions1 dataset, the diameter stayed at 15 no matter the level of perturbation. This indicates that trust-based networks are generally stable. Small changes in the graph do not have a major effect on the overall structure, which is why the diameter stayed the same.

In the Facebook Combined dataset, the diameter increased with higher levels of change. At 5% and 10% changes, the diameter was 9, but at 100%, it increased to 10. This makes sense, as social networks like Facebook are highly connected. When large changes are made, the graph spreads out more, which increases the diameter.

For the p2p-Gnutella08 dataset, the diameter stayed at 9 with 0%, 5%, and 10% changes, but decreased to 8 with 100%. This shows that with a large number of changes, the graph became more connected, causing the diameter to decrease.

Table 2: Graph Diameter for Different Datasets.

Dataset	Perturbation Level			
	0%	5%	10%	100%
Wiki-Vote	7	7	7	7
Email-Eu-core	7	7	7	6
Soc-Epinions1	15	15	15	15
facebook-combined	8	9	9	10
p2p-Gnutella08	9	9	9	8

6.3 Community Analysis

The Normalized Mutual Information (NMI) score measures how much the community structure changes after perturbations. A score close to 1 means the community structure remains similar, and a score closer to 0 means it changes more.

The NMI score is calculated by comparing the community partitions before and after perturbation. The Louvain method which is a community detection algorithm that optimizes modularity to identify communities in a network by iteratively grouping nodes that are more densely connected to each other than to the rest of the network, is used to detect communities, and the partitions of nodes into communities are compared using NMI to quantify how much the community structure has changed. [11]

In the Wiki-Vote dataset, the NMI score decreased from 1 to 0.84 after a 5% perturbation, dropped to 0.64 after a 10% perturbation, and then to 0.66 after 100%. This shows that small changes already affected the community structure, and larger changes reduced the structure's usefulness while increasing privacy.

For the Email-Eu-core dataset, the NMI score remained high, at 1 before any changes, and then dropped to 0.79 after a 5% change and 0.78 after 100%. This shows that the community structure remained mostly the same even with significant changes, balancing privacy and utility.

In the Soc-Epinions1 dataset, the NMI score dropped more drastically, from 1 to 0.53 after 5% and to 0.47 after 100%, indicating that larger changes significantly affected the structure.

For the Facebook Combined dataset, the NMI score stayed high, dropped from 1 to 0.97 after 5% and remaining at 0.93 after 100%. This indicates that the community structure remained stable even with significant changes, allowing for a balance between privacy and utility. Despite changes in the network's diameter and entropy, which showed more diversity in the graph structure, the community structure itself was largely unaffected by the perturbations. This suggests that, in highly connected networks like Facebook, while the overall graph may become more spread out and diverse, the community structure remains mostly the same. Moreover,the NMI score has some biased due to the network's size, so this could also be one reason that we observed an unexpected result.[12]

Finally, in the p2p-Gnutella08 dataset, the NMI score dropped from

1 to 0.19 after 5% and to 0.17 after 100%, showing that larger changes significantly disrupted the community structure.

Table 3: NMI Score for Different Datasets and Perturbation Levels.

Dataset	Perturbation Level			
	0%	5%	10%	100%
Wiki-Vote	1	0.84	0.64	0.66
Email-Eu-core	1	0.79	0.86	0.78
Soc-Epinions1	1	0.53	0.50	0.47
facebook-combined	1	0.97	0.98	0.93
p2p-Gnutella08	1	0.19	0.18	0.17

6.4 Evaluating Annonymity through Clustering Coefficient Entropy

Clustering coefficient entropy is a measure of the variation in clustering coefficients across a network. Higher entropy suggests that the network has a more diverse structure, with nodes having a wider range of clustering coefficients. This increased diversity can make the graph more difficult to analyze, enhancing privacy by making it harder for an adversary to identify specific nodes or their relationships. In the context of vertex refinement, as entropy increases, the anonymization process becomes stronger. This means that with higher entropy, the nodes are harder to find out their original identities because the structural patterns that would typically allow for re-identification become less predictable.

Entropy is important for evaluating anonymity in anonymized networks because a higher entropy value typically indicates that the anonymization has introduced more randomness or variability in the graph's structure. In contrast, lower entropy suggests that the graph's structure is more uniform, which could make re-identification easier. [13], [14].

For the Wiki-Vote dataset, entropy increased from 6.43 to 7.17 as the perturbation level went from 0% to 100%, showing that the changes made the structure more diverse. In the Email-Eu-core dataset, entropy showed only a small increase from 8.281 to 8.32, meaning the graph remained relatively stable even with more perturbation.

The Soc-Epinions1 dataset had a slight increase in entropy from 3.823 to 3.87, indicating that trust-based networks are more stable. In the Facebook Combined dataset, entropy rose slightly from 10.29 to 10.39, showing minor changes in its structure. Finally, for the p2p-Gnutella08 dataset, entropy increased from 1.77 to 4.07, suggesting that perturbations had a bigger impact on the network's structure.

These results indicate that increasing perturbation can lead to higher entropy, which generally enhances anonymity, but the structure of the network becomes more diverse, which may affect its utility for analysis.

7 CONCLUSION

The problem in this paper was to find a way to anonymize social network graphs while keeping important features, like the clustering coefficient, intact. This is important for ensuring that the data can still be used for meaningful analysis after anonymization.

Table 4: Clustering Coefficient Entropy for Different Datasets.

Dataset	Perturbation Level			
	0%	5%	10%	100%
Wiki-Vote	6.43	6.51	6.54	7.17
Email-Eu-core	8.281	8.29	8.289	8.32
Soc-Epinions1	3.823	3.827	3.83	3.87
facebook-combined	10.29	10.33	10.36	10.39
p2p-Gnutella08	1.77	1.94	2.1	4.07

The proposed approach used perturbation techniques to add and remove edges in the graph, while focusing on preserving triangles, which are keys to maintaining the clustering coefficient.

The experiments showed that the approach worked well at smaller perturbation levels like 5% and 10%, where the clustering coefficient was mostly preserved. However, at higher levels like 100% perturbation, the graph structure changed more, which lowered its usefulness for community analysis but improved privacy. Different datasets responded differently to perturbations according to their size and structure.

In conclusion, the proposed method successfully balances privacy and utility, though a trade-off remains. The experiments demonstrated that while privacy can be enhanced through perturbations, the utility of the data for community analysis may decrease, especially at higher levels of perturbation. Future work could focus on refining perturbation strategies to better maintain the graph's utility, particularly for larger and more complex networks, ensuring that privacy is preserved without compromising analytical values.

REFERENCES

[1] P. Hay, M. T. Baroni, and E. S. G. Sha, "Anonymizing Social Networks," *Technical Report TR-07-19*, University of Massachusetts Amherst, 2007. Available at: https://groups.cs.umass.edu/wp-content/uploads/sites/17/2022/03/hay-et-al-tr0719.pdf.

[2] L. Backstrom, C. Dwork, and J. Kleinberg, "Wherefore art thou R3579X? anonymized social networks, hidden patterns and structural steganography," *World Wide Web Conference*, 2007.

[3] L. Sweeney, "k-anonymity: a model for protecting privacy," *International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems*, 2002.

[4] N. Li and T. Li, "t-closeness: privacy beyond k-anonymity and l-diversity," *In Proceedings of the International Conference on Data Engineering (ICDE)*, 2007.

[5] S. Rad, "Social Network Analysis Repository," *GitHub*, 2024. Available at: https://github.com/sadaf-rad/Social-Network-Analysis.

[6] J. Leskovec, "Wikipedia vote network," *Stanford Large Network Dataset Collection (SNAP)*, https://snap.stanford.edu/data/wiki-Vote.html, accessed 2024.

[7] J. Leskovec, "Epinions social network," *Stanford Large Network Dataset Collection (SNAP)*, https://snap.stanford.edu/data/soc-Epinions1.html, accessed 2024.

[8] R. Leskovec, "Email-Eu-core network," *Stanford Large Network Dataset Collection (SNAP)*, https://snap.stanford.edu/data/email-Eu-core.html, accessed 2024.

[9] J. Leskovec, L. Backstrom, and J. Kleinberg, "Mapping the Evolution of Social Networks," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, 2007. Available at: https://snap.stanford.edu/data/ego-Facebook.html.

[10] Stanford Network Analysis Project, "The p2p-Gnutella08 Dataset," *Stanford Large Network Dataset Collection*, 2008. Available at: https://snap.stanford.edu/data/p2p-Gnutella08.html.

[11] V. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, 2008. Available at: https://arxiv.org/abs/0803.0476, accessed 2024.

[12] A. Radicchi, S. Fortunato, and C. Castellano, "Universality of the size distribution of communities in large networks," *arXiv:1501.03844*, 2015. Available at: https://arxiv.org/abs/1501.03844.

[13] A. Author1, B. Author2, and C. Author3, "Title of the article," *Journal Name*, vol. XX, no. YY, pp. ZZZ-ZZZ, 2024. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0378437124006794.

[14] E. C. Kenley and Y. R. Cho, "Entropy-Based Graph Clustering: Application to Biological and Social Networks," *Proceedings of the 2011 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pp. 122-129, 2011. Available at: https://ieeexplore.ieee.org/document/6137324.