Bangalore University

University Visvesvaraya College of Engineering

Department of Computer Science and Engineering

K R Circle Bengaluru-560001

"PRESERVE-WHILE-SHARING: EFFICIENT TECHNIQUES FOR PRIVACY PRESERVING IN ONLINE SOCIAL NETWORK DATA SHARING"

Project Associates

K Thanujashree 18GAEC9025

Lakshmi Priya B 18GAEC9030

Nimitha J 18GAEC9042

Under the guidance of:

Dr Venkatesh

Associate Professor, Dept. of CSE, UVCE

Year 2021-22

INTRODUCTION

Online Social Networks (OSNs) have become one of the major platforms for social interactions.

CHALLENGES

Third party service integrated with OSN acquires user information and enrich their services.

User’s data privacy concerned threaten the usage of OSN by inference attacks.

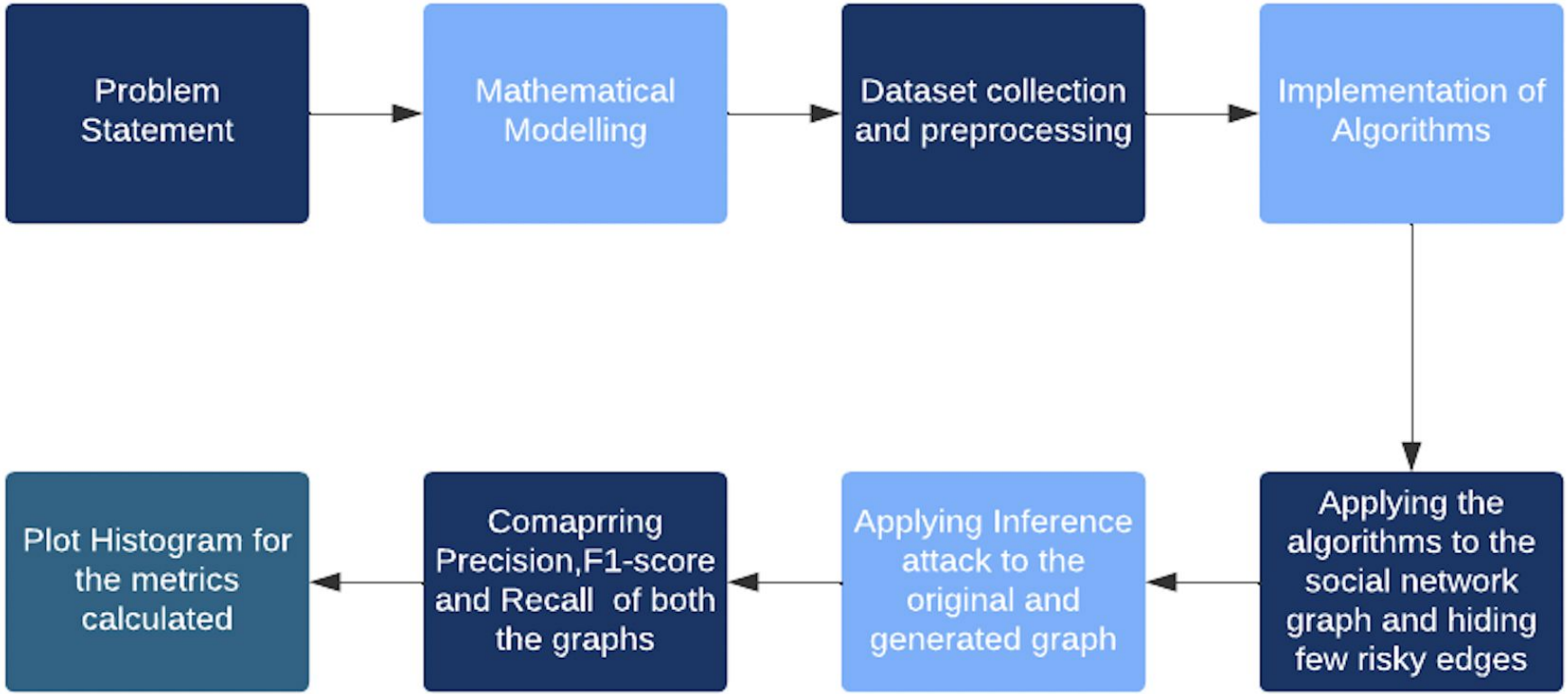
Papers

- Relationship Privacy Preservation in Publishing Online Social Networks
- Preservation of Centrality Measures in Anonymized Social Networks
- GASNA: Greedy Algorithm for Social Network Anonymization
- Privacy Preservation Based on R-Constrained Dummy Trajectory in Mobile Social Networks
- Link-Privacy Preserving Graph Embedding Data Publication with Adversarial Learning

PROPOSED MODEL

- **Problem Statement:** To defend inference attacks we propose two algorithms that enrich the data utility while privacy is preserved.
- The first proposed algorithm is the "**Privacy preservation algorithm**" : The proposed algorithm identifies edges in social attribute networks that enable users to disclose information with a privacy guarantee.
- The second algorithm is "**Social relation based dkp**" which transforms and simplifies the social relation disclosure problem into a standard multidimensional knapsack problem. The second algorithm considers the influence of this social relation on other social actors.

Flow Chart of the Project

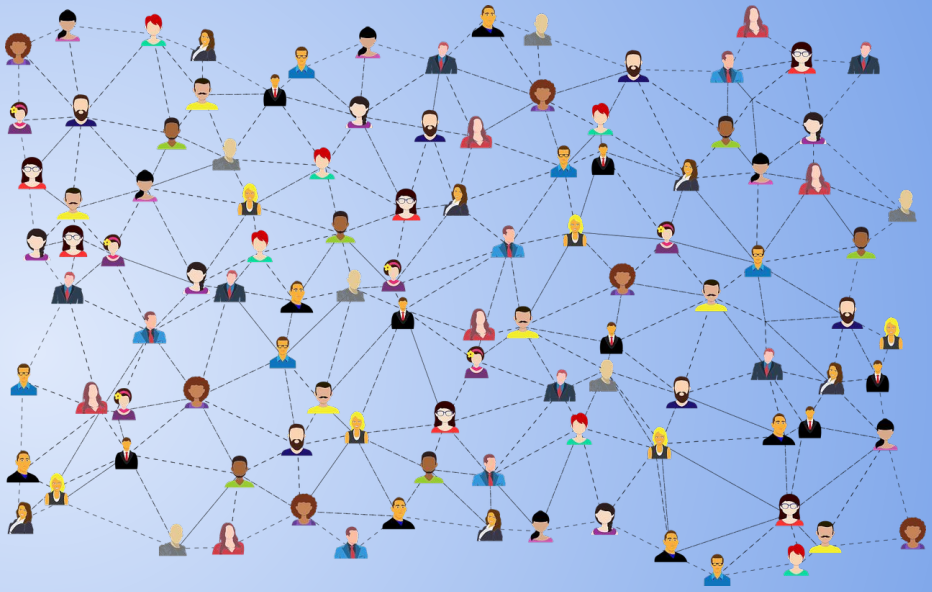


Problem formulation

- Social network publishing problem is a knapsack-like optimization problem.
- Bag capacity = Threshold disclosure.
- Items = Edges/ Links of Social Actor.
- Weight = Amount of Contribution of each edge/link for privacy disclosure.
- Profit = Utility of edge

PPA (Privacy Preserving Algorithm)

- We solve this problem by greedy approach.
- Initially the utility of all the edges is pre-computed.
- In every iteration weights of all secrets is computed and for every edge we will find , which is the ratio of edge utility to the total sum of ratios of weights of secrets to its threshold value. The edge with maximum and satisfying all privacy constraints is selected.



Step-1: Initialize

- S= $\{S_1, S_2, S_3 \ldots S_m\}$ (Secrets of a social actor)
- N = $\{N_1, N_2, N_3 \ldots N_n\}$ (Neighbor of a social actor)
- $\theta = \{\theta_1, \theta_2, \theta_3 \ldots \theta_m\}$ (Secret threshold list)

Step-2: Calculate Privacy using Jaccard Coefficient $p = \{p_1, p_2, p_3 \ldots p_n\}$

$$p_j(e_{u,v}) = |\mathbf{N_u} \cap \mathbf{N_v}| / |\mathbf{N_u} \ \mathbf{U} \ \mathbf{N_v}|$$

For every social relation existing between social actor u and social actor v :

P(Social actor u, Social actor v) =Total Common neighbors of u and v Total neighbors of u or v

Step-3: Initialize

- C = V_n (Vertex set of social actors)
- Sel = Ø
- $P_{max} = 0$
- l = [1, 2, 3,....,n]

Do Step-4 to Step-10 until l not equal to Ø

Step-4 : Initialize

- $\rho_{max} = -1$
- $S = -1$
- $W_{sel} = \emptyset$

Do Step- 5 to Step-8 for every i in l

Step-5: J = [1, 2, 3,....,n]

Do Step-6 for every j in J

Step-6: Calculate w_j using the below formula

$$W_j \leftarrow |C \cap N_i \cap S_j| \,/\, |C \cap N_i|$$

Step 7 : Calculate ρ using below formula

$$\rho \leftarrow \frac{P_i}{\sum_{k=1}^{m} w_j / \theta_j}$$

Do Step- 8 if $\rho > \rho_{max}$

Step 8 :

- $\rho_{max} = \rho$
- s=i
- $W_{sel} = W_j$

Do Step 9 if $w_j <= \theta_j$ for every j in {1,2,3,...m}

Step-9:

- Sel = Sel or {s}
- C = C and N_s
- $p_{max} = p_{max} + p_s$

Step-10: remove s from l

Step-11 : Return p_{max}, Sel

Social relation based d-kp (S-dkp)

The fundamental cause of the high complexity is because self-privacy disclosure takes into account all of the relationships between qualities and social relationships. On the one hand, omitting the correlations and treating all public information as (conditionally) independent may reduce privacy protections because two public attributes/social relations may give more information than the sum of the information presented individually. However, it will also simplify the problem by determining the weight of each public attribute or social relationship.

Step-1: Initialize

- secret neighborhood setlist S= {S1, S2, S3…..Sm} ,
- item neighborhood setlist N = {N1, N2, N3…..Nn}, and
- secret threshold list θ = {θ1,θ2,θ3…..θm}

Step-2: Calculate Privacy using Jaccard Coefficient p = {p1,p2,p3…..pn}

$$p_j(e_{u,v}) = |\mathbf{N_u} \cap \mathbf{N_v}| / |\mathbf{N_u} \ \mathbf{U} \ \mathbf{N_v}|$$

Step-3: Initialize

- C = Vn (Vertex set of social actors)
- Sel = Ø
- pmax = 0
- l = [1, 2, 3,....,n]

Do Step-4 to Step-13 until l not equal to Ø

Step-4 : Initialize

- ρmax= MIN_INT
- S = -1
- Isel = Ø

Do Step- 5 to Step-13 for every i in l

Step-5:

J = N[i] (friends of i)

Do Step-6 to Step 8 for j in J

Step-6 : Calculate Information gain for node i and node j using below formula

- Ii = log(1/(nCr(n-1, len(N[i]))))
- Ij = log(1/(nCr(n-1, len(N[N[i][j]]))))

Step-7 : Calculate ρ using below formula

ρ = P[i][j]/(Ii+Ij)

Do step 8 if ρ> ρmax

Step-8 :

- s = N[i][j]
- ρ =ρ_{max}
- Isel = Ij

Step-9 :

- flag = False
- J =θ [s]

Do step-10 for every j in J , if $I_{sel} <$ log(θ[s][j])

Step-10 :

- Sel = Sel or {s}
- $p_{max} = p_{max} + sum(p_s)$

Step-11:

- **Step 13**: return Sel, p_{max}

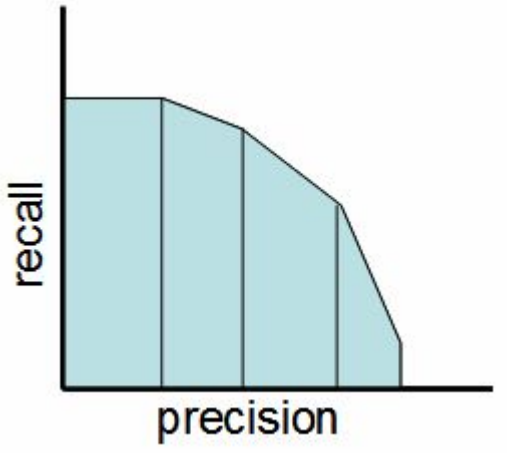
Inference Attacks

Link Prediction Algorithms used to apply inference attack

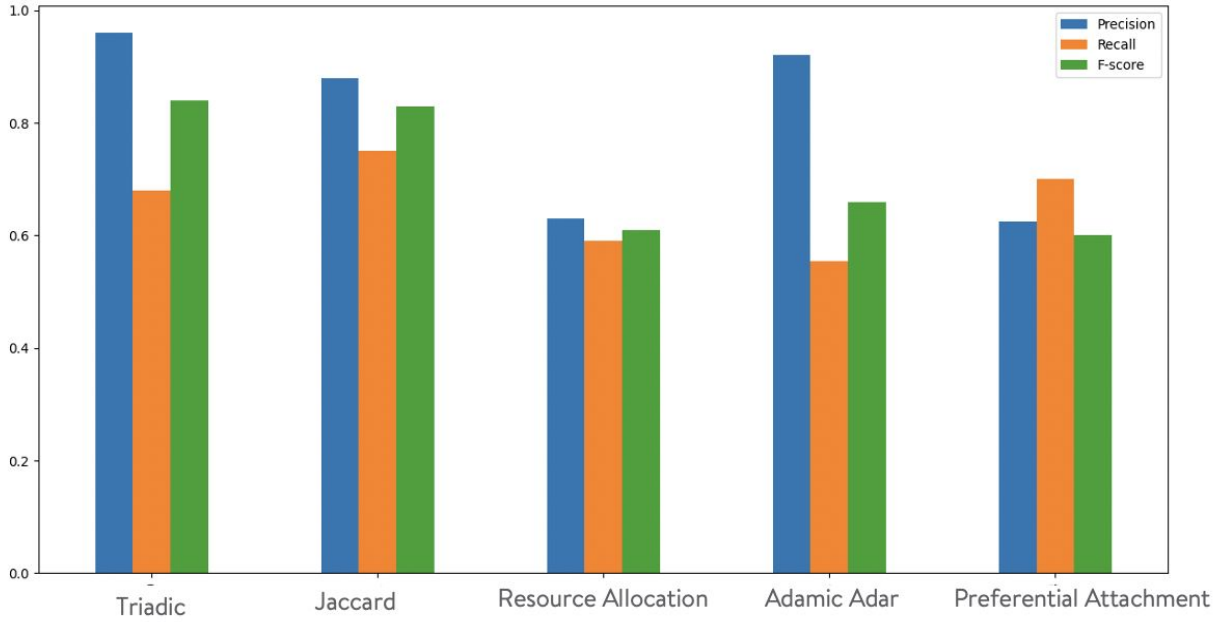
- **Triadic Closure** : $|N_x \cap N_y|$
- **Jaccard Coefficient** : $|N_x \cap N_y| / |N_x \cup N_y|$
- **Resource Allocation** : $\sum_{u \in Nx \cap Ny} 1 / N_u$
- **Adamic Adar** : $\sum_{u \in Nx \cap Ny} 1 / \log(N_u)$
- **Preferential Attachment**: $|N_x| . |N_y|$

Metrics to measure performance of inference attack

- Precision = TP / (TP+FP)
- Recall = TP / (TP+FN)
- F1 Score = (2 X precision X recall) / (precision +recall)

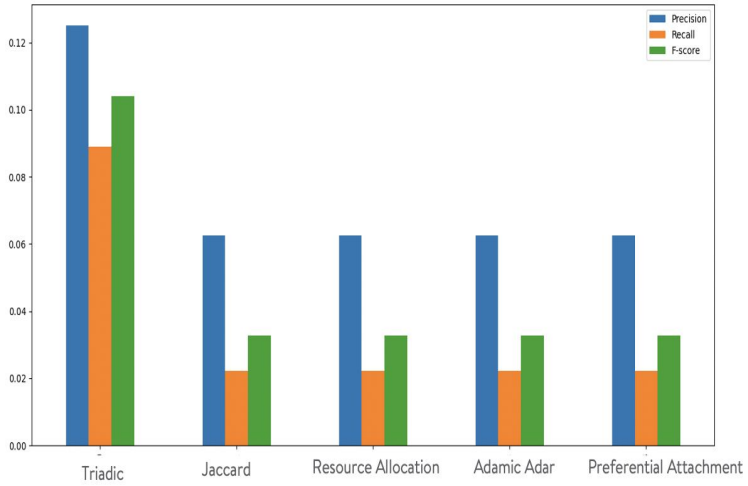


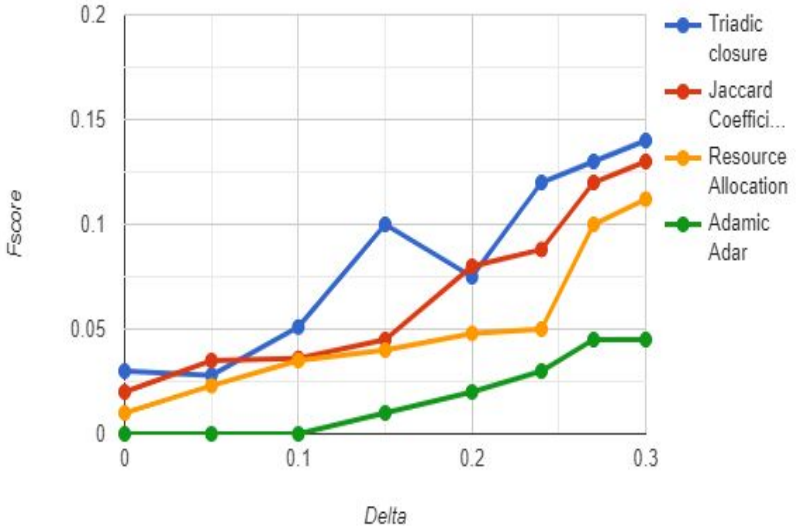
Results



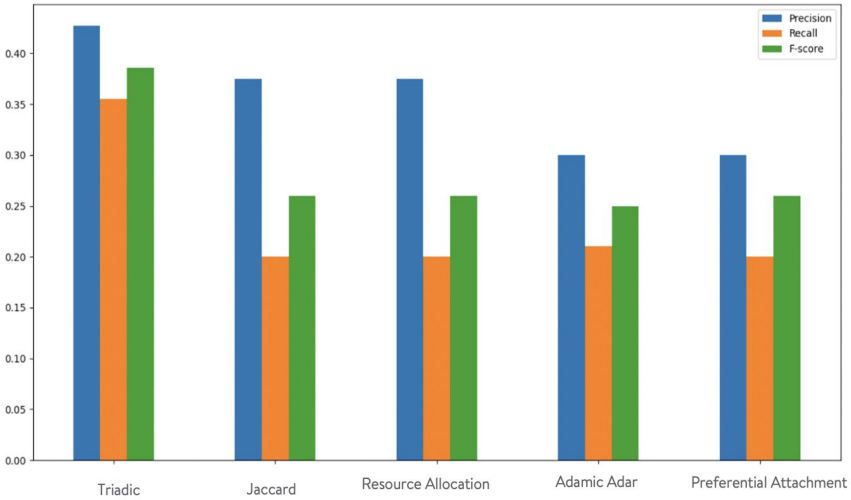
Performance of Inference attacks before PPA or SDKP

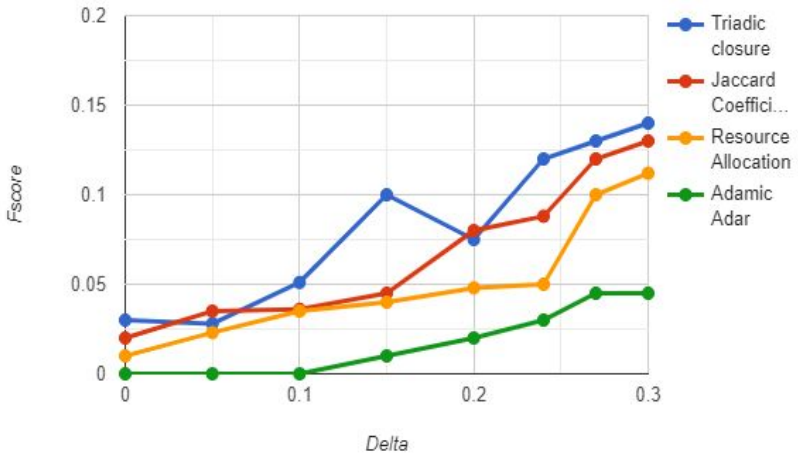
PROTECTION PERFORMANCE AFTER APPLYING PPA



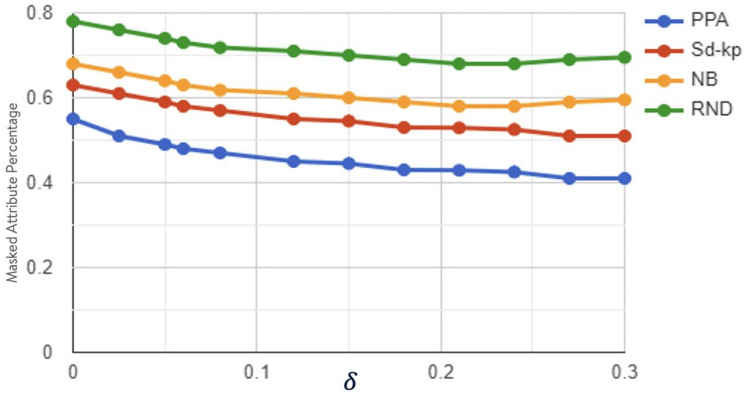


PROTECTION PERFORMANCE AFTER APPLYING SDKP

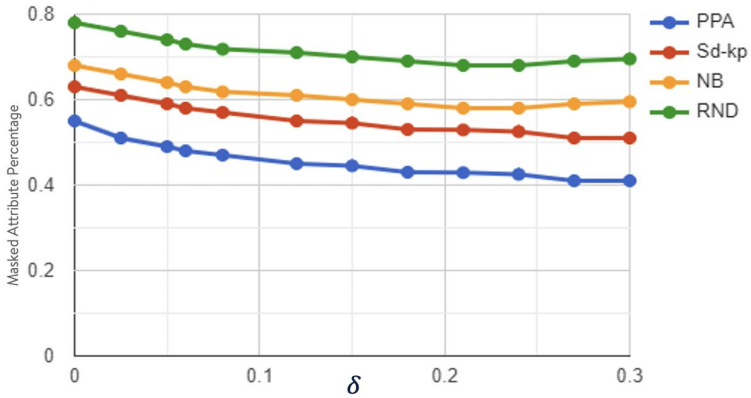




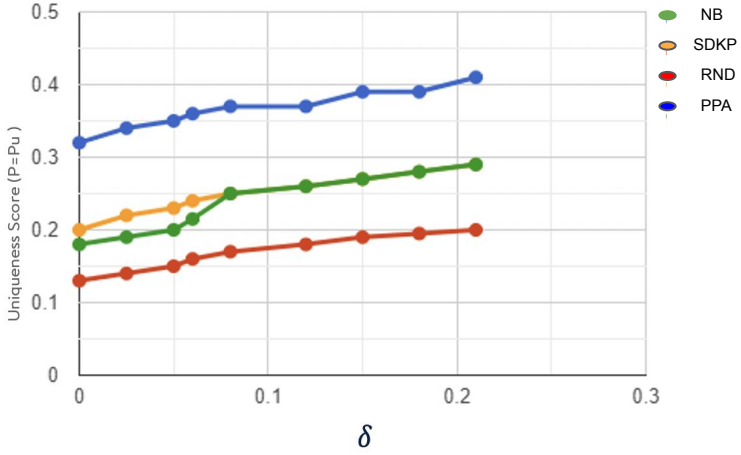
Utility Performance - Attribute Disclosure results



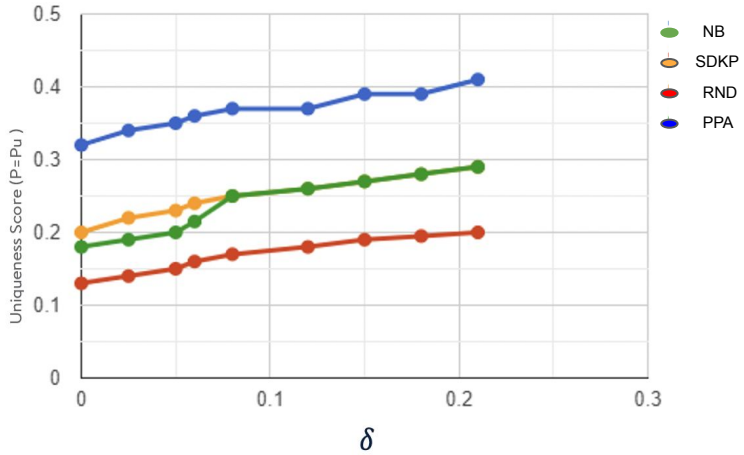
Masked Attribute - delta (Dataset-100)



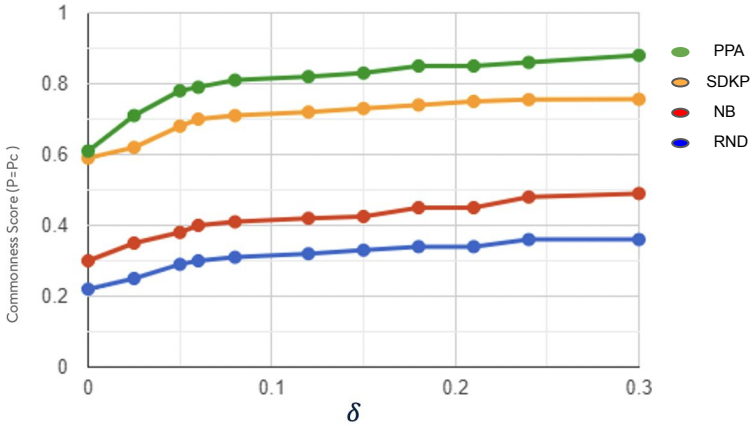
Masked Attribute - delta(Dataset-500)



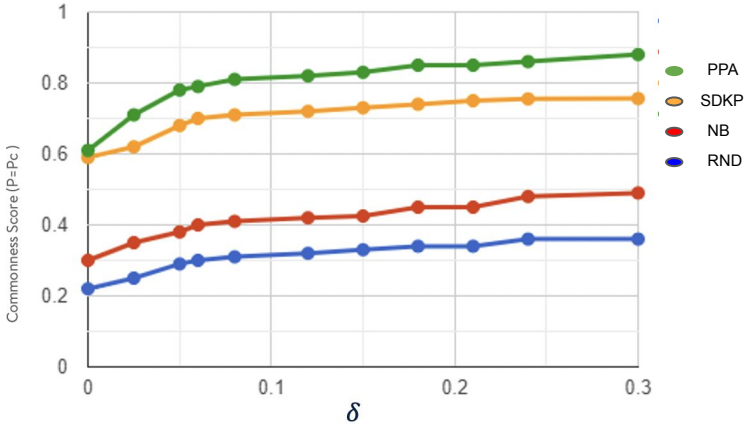
Uniqueness Score - delta (dataset-100)



Uniqueness Score - delta (dataset-500)

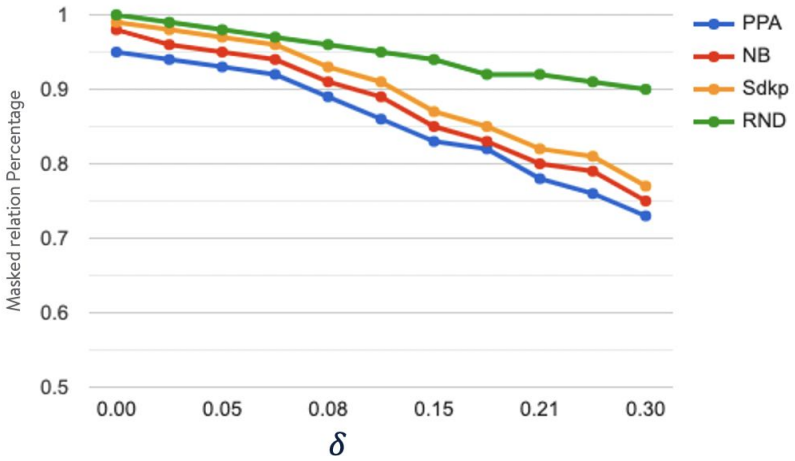


Commonness Score - delta (dataset-100)

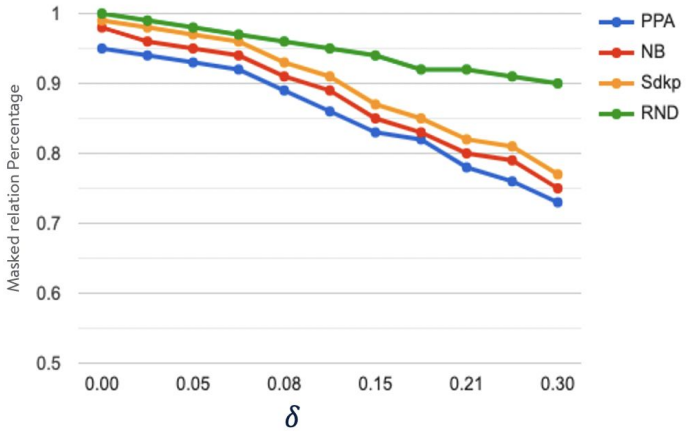


Commonness Score - delta (dataset-500)

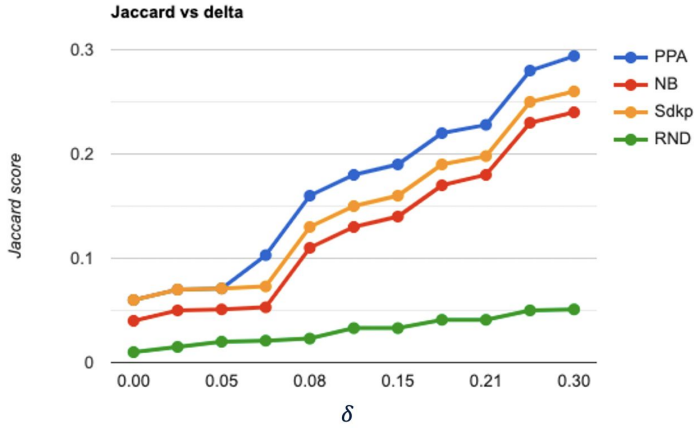
Social Relation Disclosure Results



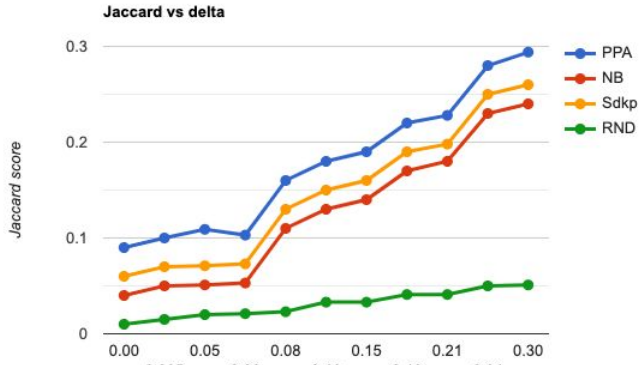
Masked Relation - delta (dataset-100)



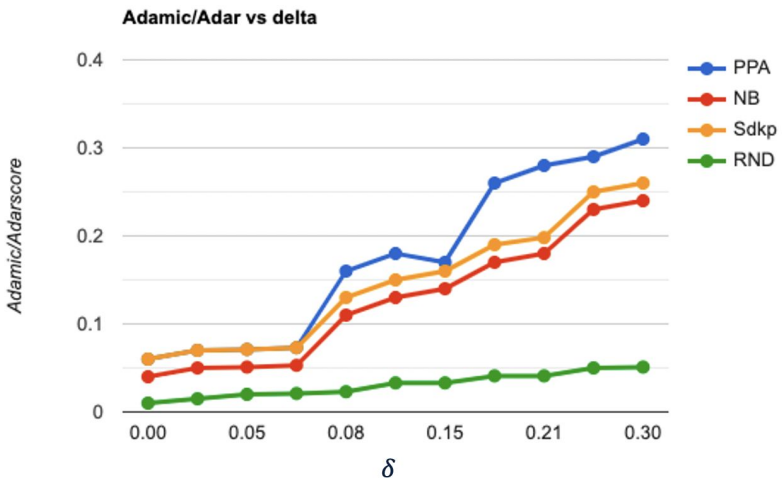
Masked Relation - delta (dataset-500)



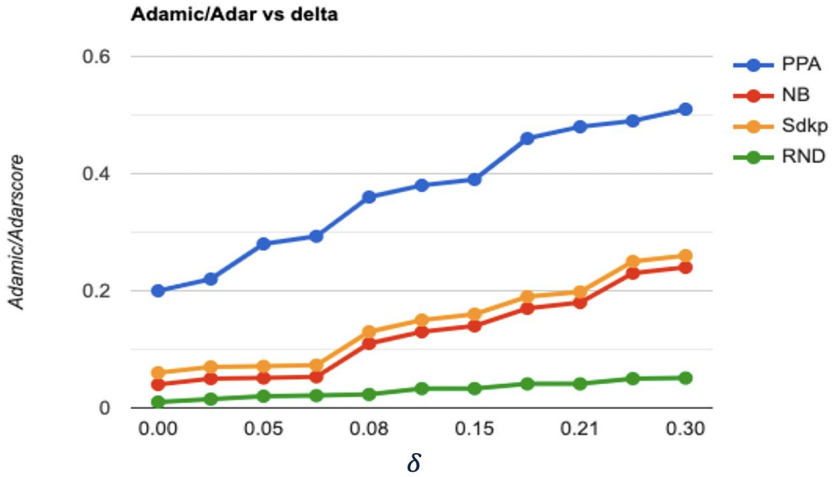
Jaccard Score - delta (dataset-100)



Jaccard Score - delta (dataset-500)



Adamic/Adar Score - delta (dataset-100)



Adamic/Adar Score - delta (dataset-500)

Conclusions and Future Enhancement

- We investigated the online social network data sharing with defense against the inference attack and formulated the optimization problem which maximizes the utility with privacy guarantee and user privacy concerns.
- The PPA algorithm always has the best performance compared to that of sd-kp in terms of utility.
- In terms of computational complexity SDKP outperforms PPA.
- **Future Work** : Implement our algorithms on larger dataset and solve privacy leakage problem in terms community information.

THANK YOU!