De-anonymizing Social Networks with Random Forest Classifier

Jiangtao Ma, Yaqiong Qiao, Guangwu Hu*, Yongzhong Huang, Arun Kumar Sangaiah, Chaoqin Zhang, Yanjun Wang and Rui Zhang

***Abstract*—Personal privacy is facing severe threats as social networks are sharing user data with advertisers, application developers and data mining researchers. Although these data are anonymized by removing personal information, such as user identity, nickname or address information, personal information still could not be protected effectively. In order to arouse the attention of people from academia and industry for privacy protection, we propose a random forest method to de-anonymize social networks. First, we convert the social network de-anonymization problem into a binary classification problem between node pairs. In order to partition large sparse social networks, we use the spectral partition method to partition large graphs into a number of small subgraphs. And then we use the features of the network structure to train the random forest classifier. As a result, candidate node pairs from anonymous network and auxiliary network can be classified as matched pair by the random forest classifier. Furthermore, we improve the efficiency of our solution through parallelizing proposed method. The experiments conducted on the real datasets show that our solution's Area Under the Curve (AUC) is 19% higher than baseline methods on average. Besides that, we test the robustness of the proposed algorithm by adding some noisy data, and the result demonstrates that our solution has good robustness.**

***Index Terms*—De-anonymization, graph partition, network structure, random forest, social network analysis**

This work was supported by the National Nature Science Foundation of China under Grant No.61402255, 61170292, 61373161, 61702462 and 61672470, in part by the National Key Research and Development Projects Intergovernmental Cooperation in Science and Technology of China under Grant No.12016YFE0100600 and 12016YFE0100300, in part by the Henan Province Science and Technology Department Foundation under Grant No.162102410076 and 162102310578, in part by the Henan Province Educational Committee Foundation under Grant No.16A520062 and 17A520064, in part by the Natural Science Foundation of Guangdong Province under Grant No.2015A030310492, and in part by the Fundamental Research Project of Shenzhen Municipality under Grant No.JCYJ20160301152145171.

J. Ma, Y. Qiao, Y. Huang, C. Zhang, Y. Wang, and R. Zhang are with the State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou450001, Henan, PRC (email: kitesmile2000@gmail.com; qiaoyq2016@gmail.com; 18600200718@163.com; zhangrui@ncwu.edu.cn; zhangcq@zzuli.edu.cn, 64502147@qq.com;). J. Ma is also with Zhengzhou University of Light Industry, Zhengzhou 450001, Henan, PRC. R. Zhang is also with North China University of Water Resources and Electric Power, Zhengzhou 450045, Henan, PRC.

G. Hu is with the School of Computer Science, Shenzhen Institute of Information Technology, 518172 Shenzhen, Guangdong, PRC (corresponding author, e-mail: hugw@sziit.edu.cn).

Arun Kumar Sangaiah is with the School of Computer Science and Engineering, VIT University, Vellore 632014, India (e-mail: arunkumarsangaiah@gmail.com).

I. INTRODUCTION

Many social networks such as Facebook and Twitter generate a large amount of data every day, which contain many useful information such as user profiles, social relationships, and daily life etc. Hence, social network analysis has attracted researchers' attention in recent years [1][2][3][4]. Social network data are provided to third parties for precise marketing, personal recommendation [5], academic research or data mining [6]. Nevertheless, the release of social data will lead to privacy disclosure and trigger public concerns. Removing user ID before publishing data is not enough to protect privacy [7]. Researchers have proposed a variety of data privacy protection methods and corresponding breach methods, such as k-degree anonymity [8] using the graph-based method, where an auxiliary graph is employed to de-anonymize social graphs. Unfortunately, there are three deficiencies. Firstly, most of the attacks only focus on one certain type of network, and the proposed method does not suit to other types of networks. Therefore, the proposed model is not universal for general social networks. Secondly, previous work assume the attacker's prior knowledge is limited. Some methods assume that an attacker has only one type of information, such as node degree. There are also other methods assuming that an attacker only has network topology information [9]. Some methods only require node attribute information, such as personal profiles, user behavior trajectories, and user-generated content information [10]. However, attackers usually own more personal information than we expected. Thirdly, the modeling of user's feature is more difficult, and automatically extracting features to de-anonymize social networks is challenging.

To overcome these shortcomings, we are aiming to build a comprehensive, automated attack model which can alert social network providers to prevent potential privacy exposures. We use this model to assess the effectiveness of structural anonymous methods, so that we can give suggestions to privacy protection researchers.

We are facing three main challenges. First, it is difficult to build a generic structure-based feature model. As it is challenging to extract features from network structural features such as node degrees, link relationships, and neighbor's subgraphs. Second, the sizes of social networks are large, and it is difficult to explore network structures in a sparse social network. Therefore, how to partition a large graph into small

subgraphs is a challenging problem. Third, how to match the nodes between anonymous graph and auxiliary graph with an efficient and automated way is also a challenging task. Since, existing de-anonymize social networks methods need labor-intensive work and need a lot of features. We propose an automatic method to de-anonymize social networks with desirable accuracy and efficiency only with network structures.

In this paper, we propose a random forest classifier to de-anonymize social networks. First of all, we convert the social network de-anonymization problem into a node matching problem between networks. Then, we utilize the spectral partition method to partition large graphs into a number of small subgraphs. Subsequently, the features of network structures (such as node degrees, node clustering coefficient and eigenvector centrality) are used to train the random forest classifier. Thereafter, the candidate node pairs of the anonymous network and the auxiliary network are classified as matched pairs by the classifier. Then, we parallelize the proposed algorithm to improve the efficiency of the solution. Finally, we test the robustness of the proposed algorithm by adding some noisy data. The experimental results demonstrate that the proposed algorithm is effective and robust. To summarize, the main contributions of this paper are as follows:

1. We propose a random forest classifier to de-anonymize social networks, which is able to identify the matched node pairs automatically and make the attack model more efficiently.
2. We extract node degree, node clustering coefficient, and eigenvector centrality features from network structures, and employ these features to train the random forest classifier.
3. We utilize the spectral method to partition large social networks into small subgraphs, which makes our proposed method to be parallelable with multiple processors.
4. We verified the effectiveness of our proposed algorithm with real dataset. Test result shows that our solution's Area Under the Curve (AUC) is 19% higher than the baseline methods on average. The robustness of the method also proved by adding noisy data in the test dataset.

The rest of this paper is organized as follows: Section II introduces related work. Section III defines some terminologies and formulates a de-anonymization problem. The proposed scheme for de-anonymizing social networks is described in Section IV, and the test of proposed algorithm is covered in Section V. Finally, we conclude the whole paper and outline possible future work in Section VI.

II. RELATED WORK

In recent years, de-anonymizing social networks has attracted significant attentions from researchers and social media marketing companies. Researchers proposed various methods to de-anonymize social networks, which can be divided into three categories: node information based, network structure based and knowledge graph based approaches.

A. Node Information Based De-anonymization

Attackers only employ node information to de-anonymize social networks. Node information contains user's nicknames, profiles, user generated contents and user behaviors. Latanya [11] models user behaviors in social networks and utilizes usernames to correlate social network users. Mohotra et al. [12] propose a profile-based similarity method to match similar users across social networks, where a classifier is used for profile matching. They employ user's digital footprints such as usernames, nicknames, locations and photos to calculate user profile similarity. Tan et al. [13] find that about 50% users use the same username across different online social networks (OSNs). Based on this finding, Zafarani et al. [14] utilize username to de-anonymize user accounts by adding or deleting the prefix/postfix of usernames. Furthermore, Peritio et al. [15] estimate the uniqueness of username by modeling a Markov chain process. Similarly, Liu et al. [16] propose an unsupervised approach which takes the n-gram model to estimate the uniqueness of a username. Moreover, Iofciu et al. [17] de-anonymize users across OSNs by measuring the distance between user profiles based on their IDs and tags through string edit distance. In addition, Zhang et al. [18] use the Jaro-Winkler [19] method to de-anonymize user accounts among different OSNs with a language model [20]. This method first converts user profiles into a bag of word vectors and then calculates profile similarity by analyzing vector similarity through the cosine distance. Although above methods can achieve a good performance in some scenarios, the biggest challenge is the authenticity and integrality of user profile information. Such as when social networks are suffering Sybil attacks the user profile is not real. Therefore, node information based de-anonymization methods cannot achieve a good result when the veracity of profiles is not guaranteed.

B. Network Structure Based De-anonymization

Attackers also utilize the feature of network structure to de-anonymize users across social networks. Kazemi et al. [21] propose a graph matching method to map users between two social networks. However, this method cannot solve the cold start problem and causes higher time and space complexity. Fabiana et al. [22] use the bootstrap filter and graph segmentation method to de-anonymize scale-free social networks. Nonetheless, there is a user's maximum group in social networks, which makes the common neighbor threshold go erroneous.

Many the privacy protection models derive from k-anonymity [23], assuming that the attacker has limited prior knowledge. Unfortunately, if the attacker has more priori knowledge than imagined, such privacy protection method becomes fragile. For example, when k-degree anonymity is used for preventing attacks. The attack model that uses the equal number of users and the degree of vertex to de-anonymize users is invalid. But this method cannot prevent community re-identification [24]. Researchers have proposed similar method, such as k-neighborhood anonymity [25]. In addition, there are some de-anonymization methods based on clustering or aggregation, differential privacy [26] and random walk methods.

A community-enhanced de-anonymization algorithm is proposed to complete a two-stage matching process [20]. The de-anonymization is executed at community level, after which it is extended to the entire network. Due to the asymmetry of communities on both sides, their two-stage approach may meet new problems [27] when network structure is destroyed. Korula and Lattanzi [28] design a simple, local and effective algorithm to solve the de-anonymization problem and give the theoretical

guarantee of the algorithm performance. Nevertheless, in case merely a small number of seed nodes are provided, this method will not be effective. There are quite a few solutions that do not need seed users, such as [9] and [29]. These methods assume that the attacker has an auxiliary network that overlaps the anonymous network and the attacker is sure that the priori information is 100% correct. Such methods have following shortcomings. First, their social network model is unweighted and ignores the close relationship between users. Second, they barely utilize the local features of a network (such as node degree and common neighbors) and usually ignore the global features of the network, which caused low accuracy rate. Third, above methods are not able to extend to general social networks as they only utilize local features of networks. In contrast, our approach is based on the general features of social networks, so it is more versatile.

C. Construction Model-based De-anonymization

Researchers construct lots of labor-intensive models to de-anonymize social networks. To de-anonymize Google+ and Pokec social networks, Qian et al. [30] employ a knowledge graph to model priori knowledge of attackers to improve the accuracy of de-anonymization results. Hay et al. [31] use vertex refinement queries, subgraph queries, and hub print queries to de-anonymize social networks, but they have neither the actual ability to model an attacker nor the priori knowledge of the attacker owns. Wondracek et al. [32] point out that group relationships can identify individual users in social networks. Narayanan et al. [33] use the random forest method to predict the link relationships between nodes and consequently de-anonymize the Kaggle dataset. Such methods require constructing complex attack models, which demand lots of labor-intensive work. By contrast, our method is to automatically extract the structural features of the node from the network structure, so the efficiency is higher.

To analyze the de-anonymization and privacy problem, Narayanan and Shmatikov [20] design an account linking method for Twitter and Flickr, in which account-linking only based on network topology. Even if the overlapped information between target network and auxiliary network is little, the robustness of this method is good. Similar techniques are utilized to de-anonymize the Netflix dataset with the IMDB dataset. Narayanan et al. [33] employ the de-anonymization method to do link prediction for the Kaggle dataset and utilize the random forest method to do link prediction between nodes. Similarly, Sharad and Danezis [34] address the de-anonymization problem using the random forest method to match the node pairs automatically. Korula and Lattanzi [28] use the Erdös-Rényi (ER) random graph and preferential attachment model to link accounts from intense nodes (nodes with a large number of neighbor nodes). This method proposes a many-to-many mapping algorithm based on the number of unmatched users and common neighbors. In addition, it uses two control parameters to fine tune the algorithm performance. Actually, the ER random graph model is only mathematically meaningful and impractical in OSNs. Even the quantification is effective under the assumption of identified seeds, it is impractical for real-world de-anonymization attacks.

III. Problem Statement and Formulation

A. Problem Statement

To better understand the de-anonymization problem across OSNs, we take Figure 1 as an example. Suppose an attacker owns an auxiliary graph G^s, which could be used to de-anonymize the target graph G^t. The node pairs linked by the dotted line is the goal of solution. Once obtaining the mapping relationships between these nodes, we can get according node information. In short, our goal is to find the node pairs across social networks accurately and effectively.

We are facing following challenges for finding node pairs across large social networks. First of all, it is difficult to partition the large graph into subgraphs. Second, it is challenging to match similar subgraphs across social networks. Third, it is difficult to compute the similarity of the nodes across graphs, and it is challenging to automatically classify the node pairs. Besides all above challenges, it is challenging to parallel the proposed method efficiently.

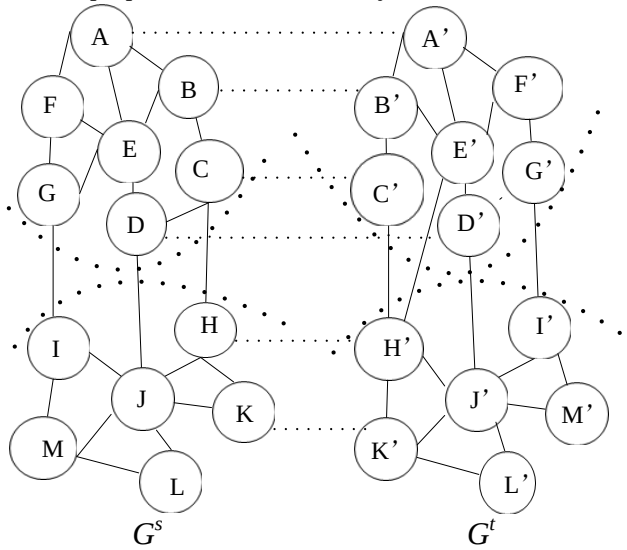

Figure 1. An illustration of de-anonymizing social networks. G^s can be used to de-anonymize G^t. G^s and G^t are partitioned by thick dotted lines. The thin dotted lines between nodes in different networks (e.g., A-A', B-B', C-C', D-D', H-H', K-K') are matched pairs. The goal of de-anonymization is to find out the node pairs among different OSNs with desirable performance and accuracy.

B. Problem Formulation

The relationship between accounts across OSNs is same as the mapping relationship between vertices in social graphs. Details described in Definition 1. Table 1 shows the notations used in the paper.

Definition 1: Given anonymized social network $G^t=(V^t,E^t,W^t)$, $V^t=\{i|i$ is a node$\}$ represents the set of user accounts. $E^t=\{l^a_{i,j}|i, j\in V^t\}$ indicates the social relationships between user accounts, $W^t=\{w^t_{i,j}|i, j\in V^t, l^a_{i,j}\in E^t, w^t_{i,j}$ is a real number$\}$ is the weight of edge in E^t, and $w^t_{i,j}=1$ if G^t is an unweighted graph.

To de-anonymize G^t, we use an auxiliary graph $G^s=(V^s,E^s,W^s)$, which has overlapping users with G^t and can be constructed using multiple dataset sources, e.g., online social network, or various kinds of published data.

TABLE 1. SUMMARY OF THE NOTATIONS

Notation	Definition
$G^s=(V^s,E^s)$	Auxiliary graph
$G^t=(V^t,E^t)$	Anonymized (target) graph
$p\in V^s, q\in V^t$	Nodes of graph
$\|V^s\|=m$, $\|V^t\|=n$	Number of nodes in a graph
w	Weight of edge in a graph
d^v	Degree of node v
c^v	Closeness centrality of node v
b^v	Betweenness centrality of node v
e^v	Eigenvector centrality of node v
θ	Similarity threshold of two matched node pairs
$\delta(x, y)$	Information gain function of vector x and y
S^L	Set of left children in decision tree (unmatched pair)
S^R	Set of right children in decision tree (matched pair)
$\boldsymbol{A}$	Adjacency matrix of social network
$\boldsymbol{D}$	Degree matrix of social network
$\boldsymbol{L}$	Laplacian matrix of social network
|V|	The number of nodes in the graph
|E|	The number of edges in the graph

Our goal is to accurately find the node mapping between the anonymous graph G^t and the auxiliary graph G^s. This mapping relationship can be described as $\sigma : G^s \rightarrow G^t$, for $p\in V^s$, $q\in V^t$, if p,q is a node pair, $\sigma(p,q) = 1$, otherwise $\sigma(p,q) =0$. How can we find the set of node pairs in large social networks? What kind of features can be used to compute the similarity of two nodes? How can we find an automatic method to identify the matched node pairs and unmatched pairs? We address above issues in below five steps, first, we partition the large sparse graph into subgraphs. Second, we find the similar subgraphs across graphs. Third, we build a model to match similar nodes across subgraphs. Fourth, we propose a method to measure node similarity based on network structure features. Finally, we find a proper strategy to match the nodes with an automatic method. We will detail our solution in next section.

IV. SCHEME DETAILS

In this section, we describe our scheme in detail. We employ network structure to de-anonymize social network. First, we use the spectral segmentation method to partition large social graph into several small subgraphs and then we match similar partitioned subgraphs. Subsequently, the features of network nodes, such as node degree, clustering coefficient and eigenvector centrality are extracted from matched subgraphs, and these features are taken as the feature vectors of nodes. The matching of nodes between the anonymous network and the auxiliary network can be considered as a binary classification problem. If the node pair is matched, the class label is 1, otherwise, the class label is 0. Thereafter, we use the random forest classifier to classify matching nodes according to their feature vectors. The classification process is paralleled through many processors before outputting matched node pairs. The specific framework is shown in Figure 2.

A. Degree Centrality

Degree centrality's intuition is that if a node has larger number of node neighbors, the node has greater influence [35]. Degree centrality measures the degree of the node's direct influence to its neighbors [36], which means the bigger the node degree, the more neighbors are influenced by the node. However, nodes with same degree in the networks with different sizes have different degree of influences. For the purpose of comparison, we normalize degree centrality of node v_i in (1):

$$DC(i) = \frac{k_i}{k_{\max}} \tag{1}$$

where, $k_i = \sum_i a_{ij}$, and a_{ij} is the element of adjacency matrix $\boldsymbol{A}$ of social graph, n is the number of nodes in social graph, k_{max} is maximum node degree. In directed networks, in-degree and out-degree carries different meanings (in-degree means the popularity of a node, out-degree means the gregarious of a node). The in-degree and out-degree centrality can be measured separately according to different type of social networks. In large-scale social networks, the degree distribution follows power-law distribution. Individual popularity across different networks is similar, i.e., individuals in different networks have similar behaviors. We can utilize this feature to measure node features in social networks.

B. Node's Clustering Coefficient

Clustering coefficient [37] measures the degrees of nodes that tend to cluster together in a social graph. The node's clustering coefficient quantifies the closeness of its neighbors tending to form a complete graph. The local clustering coefficient c_i for a vertex v_i is given by the proportion of edges between the vertices within its neighbors divided by the number of edges of the complete graph constructed by the neighbors. In a directed graph, e_{ij} is distinct from e_{ji}, so for each neighbor N_i there are $k_i(k_i-1)$ edges that exist in the complete graph constructed by its neighbors (k_i is the number of neighbors of a vertex). Therefore, the node clustering coefficient in directed graphs is given as (2):

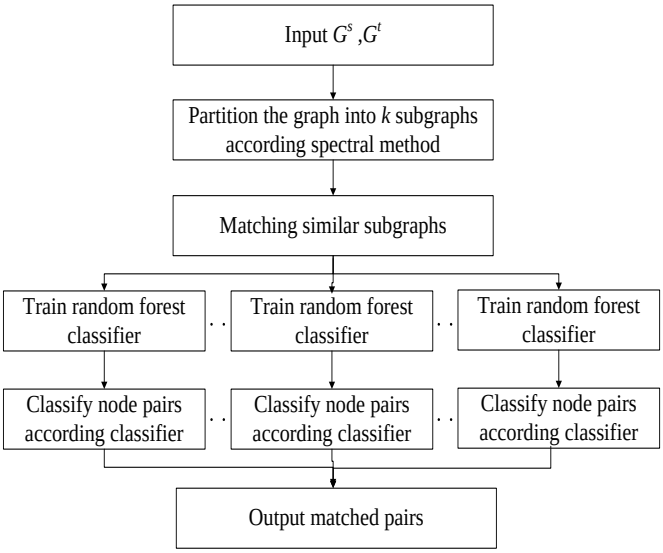

Figure 2. Framework of the proposed solution. The solution first partitions the graph into k subgraphs and then matches the similar subgraphs before training the forest classifier and classifying node pairs with k processors parallelly.

$$C_i = \frac{|\{e_{jk} : v_j, v_k \in N_i, e_{jk} \in E\}|}{k_i(k_i-1)} \tag{2}$$

For an undirected graph, $k_i(k_i-1)/2$ edges exist among the vertices within the neighbor's complete graph. Thus, the local clustering coefficient for undirected graphs is measured according to (3):

$$C_i = \frac{2|\{e_{jk} : v_j, v_k \in N_i, e_{jk} \in E\}|}{k_i(k_i-1)} \quad (3)$$

Therefore, node's clustering coefficient can be employed as node features for measuring node structure similarity.

C. Eigenvector Centrality

Eigenvector centrality [38] is an important indicator for measuring the importance of a node in a social graph. Node degree believes surrounding neighbor nodes have equal importance, but in fact the importance of nodes is unequal and the influence of their neighbors should be considered. The eigenvector centrality value of the node could be high if its neighbor's eigenvector centrality value is high. The eigenvector of a network is the vector of its adjacent matrix network corresponding to the largest eigenvalue. The importance of a node not only depends on the number of its neighbors (node degree) but also depends on the importance of its neighbors. The importance of the node v_i is described in (4):

$$EC(i) = x_i = \lambda^{-1}\sum_{j=1}^{n} a_{ij}x_j \quad (4)$$

where $x = [\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3, \cdots, \mathrm{x}_n]^T$. After several iterations, it will reach a steady state, which can be written as $\mathbf{x} = \lambda^{-1}Ax$ and $\boldsymbol{x}$ is the eigenvector corresponding to the eigenvalue λ of matrix $\boldsymbol{A}$. Eigenvector centrality emphasizes the surrounding environment of a node (the quantity and value of neighbors). The essence is that the importance score of a node is the sum of the importance scores of all its neighbors. The node can connect to many other important nodes to promote its importance. High node scores could be obtained by linking to a large number of low-value nodes or linking to a small number of high-value nodes. Eigenvector is a linear combination of nodes in a network, we can describe eigenvector with linear system of equations as it is a linear combination of all nodes in a network. The eigenvector corresponding to the maximum eigenvalue represents the importance of each node. Ranking based on network global features mainly considers the global information of a network. Although this is more accurate than other methods, it has high time complexity and does not work for large-scale networks. Therefore, we propose to divide large graph into small subgraphs.

D. Graph Partition with the Spectral Method

For large social networks, the cost of computational node's clustering coefficient and eigenvector centrality are expensive. Therefore, we propose to partition the large social network into a number of small graphs and then process them parallelly. In a real large-scale network, connected component can be employed to partition it. In this paper, we use the spectrum partitioning algorithm to partition the social graph [39], the idea of which comes from the spectral partition. The spectrum of the matrix is its eigenvalue and corresponding eigenvector.

Given a graph G=(V, E) with adjacency matrix $\boldsymbol{A}$, where an entry A_{ij} denotes an edge between node i and j, and degree matrix $\boldsymbol{D}$ is a diagonal matrix. Each diagonal entry of a row i, d_{ii} is the degree of node i. The Laplacian matrix $\boldsymbol{L}$ is defined as $\boldsymbol{L}$=$\boldsymbol{D}$-$\boldsymbol{A}$. The ratio-cut partition for graph G is defined as a partition of v into disjoint U and W, minimizing the ratio of the number of edges across this cut to the number of pairs of vertices that support such edges. This ratio can be described as in (5):

$$\min\left(\frac{|E(G)\cap(U\times W)|}{|U|\cdot|W|}\right) \quad (5)$$

The graph partition in our solution follows three main steps:

(1) For a given graph G = (V, E), the Laplacian matrix of the social graph can be calculated by $\boldsymbol{L}$ = $\boldsymbol{D}$-$\boldsymbol{A}$;

(2) After the eigenvalue decomposition of the matrix $\boldsymbol{L}$, the eigenvector matrix $\boldsymbol{Q}$ is constructed by taking the corresponding eigenvector of the k largest eigenvalues.

(3) K-means clustering algorithm is used to cluster the element of the matrix $\boldsymbol{Q}$, so that the similar vertices could be clustered together.

Thus, a large social graph can be divided into several small subgraphs. The auxiliary graph and the anonymous graph are divided into subgraphs by the same method. Corresponding subgraphs can be matched according to the k largest eigenvalues, and the corresponding nodes could be matched according to the structural similarity of the nodes in the matching subgraphs. In order to achieve social network de-anonymization, we will use the random forest classifier to identify the possible candidate matching node pairs of two subgraphs. As a result, matching nodes are classified into one class and the unmatched nodes are classified into another class.

E. Random Forest Classifier

Random forest [40] employs a random method to build a forest, which is composed by a lot of decision trees, and the decision trees in the random forest are unrelated. When new data are input to the random forest, each of the decision tree judges which label should the data belong to separately and classify the data according to the vote of total number of labels.

Random forest works well on large-scale datasets. It is able to handle high-dimensional data and does not need feature selection. After training, it can tell what features are important. The training speed is fast in random forests. During the training process, it is possible to detect the mutual influence between features, and it is easy to implement the parallelization of the classifying process. Although each decision tree in the random forest is weak, the final result of the random classifier is powerful.

Our de-anonymization model is expressed as a set of decision trees based on graph features, which uses random forest model to integrate the prediction result of each decision tree together. Thus, it leads to an overall performance improvement. We use bagging [41] and random node optimization methods [42] to train decision trees, and utilize $x \in v_p$ and $y \in v_q$ as features to make the split decision function in the process of making decision trees. The split decision is defined as (6):

$$\delta(\mathrm{x},\mathrm{y}) = \begin{cases} 0 & if\ x = y = 0 \\ \frac{x\cdot y}{|x||y|} & otherwise \end{cases} \quad (6)$$

We calculate $\delta(v_p[i], v_q[j])$ for each node feature pairs (v_p, v_q), where $i, j \in \{0, n-1\}$. For every feature pair $(v_p[i], v_q[j])$, the decision tree gets $\delta(v_p[i], v_q[j])$ and classifies the data to the left child or right child according to threshold τ. In the training process, every spilt node is given a threshold τ to classify $(v_p[i], v_q[j])$ for the maximum information gain. The average predicted results of the decision trees are the final result of a decision forest, which is defined in (7):

$$\mathrm{p}(\mathrm{c}|\mathrm{v}) = \frac{1}{T}\sum_{t=1}^{T} p_t(\mathrm{c}|\mathrm{v}) \quad (7)$$

Where $p(c|v)$ is the prior probability of the feature vector of node v (degree, clustering coefficient, eigenvector), T is the number of the decision trees, and $p_t(c|v)$ is the prediction of the single decision trees.

F. Node Matching Algorithm

We use the random forest classifier to classify the labels of node pairs between anonymous and auxiliary graphs. Degree centrality, clustering coefficient and eigenvector composes the feature vector $v[i]= [d_i^v, c_i^v, e_i^v]$, and the classifier predicts the label of the node pairs according to their feature vectors. If the label is 1, it means the node pair is matched. If the label is 0, it means the node pair is unmatched. θ is a threshold for cosine similarity of two vectors. If two vectors' cosine similarity is bigger than the threshold, the two vectors are more similar. The details of the proposed solution are described in Algorithm 1.

TABLE 2 ALGORITHM OF DE-ANONYMIZATION

Algorithm 1 Social Graph De-Anonymization with Random Forest Classifier

Input: anonymous graph $G^s(V^s, E^s)$, auxiliary graph $G^t(V^t, E^t)$,$|V^s|=m$, $|V^t|=n$, S^L is the set of classified to left child, S^R the set of classified to left child, $p \in V1$, $q \in V2$.
Output: a set of matched node pairs S^R.

```
for(i=0;i<m;i++)// initialize the feature vector of node v_p in G^s
 calculate v_p[i]= [d_i^v, c_i^v, e_i^v];
for(j=0;j<n;j++)// initialize the feature vector of node v_q in G^t
 calculate v_q[j]= [d_j^v, c_j^v, e_j^v];
for(i=0;i<m;i++)
 for(j=0;j<n;i++)
 while(p=breadfirstsearch(G^s))
 {
  while(q=breadfirstsearch(G^t))
  {
  if (cossim(v_p[i], v_q[j])>θ)//compute the cosine similarity of feature vector
    S^R=S^R ∪ (p,q);
   else
    S^L=S^L ∪ (p,q);
   V^t = V^t -q;
  }
  V^s = V^s -p;
 }
return S^R;
```

V. EVALUATION

In this section, we present datasets in part A, and describe the comparison method in part B. After that, we present our evaluation results in part C and give out some discussions in part D.

A. Datasets

MAG: Microsoft academic graph data (MAG) is extracted from [43]. We use papers published in artificial intelligence, deep learning, data mining, social network analysis, database area which range from year 2000 to 2016 to construct the cooperation graph. The papers published in odd years construct the anonymous graph, and those published in even years construct the auxiliary graph. The graph contains information that related to author, paper and author's cooperation.

DBLP: The DBLP dataset is extracted from the DBLP website. Similar to the MAG dataset, we use papers published in odd years to construct the anonymous graph, and papers published in even years to construct the auxiliary graph. The papers are mainly in artificial intelligence, deep learning, data mining, social network analysis, and database areas and range from year 2000 to 2016.

Mendeley: Mendeley [44] is a reference management software that offers social network functions. We extract the author's social relationships from social networks to construct the social graph. The graph built in January 2017 is an anonymous graph and the graph built in May 2017 is an auxiliary graph.

Arnetminer: ArnetMiner [45] is an academic social network that offers expert finding and paper recommendation services. Similar to the Mendeley dataset, we use same method to build the anonymous graph and the auxiliary graph.

MAG-Aminer: We employ MAG as an auxiliary graph to de-anonymize Aminer dataset.

DBLP-Mendeley: Similar to MAG-Aminer, DBLP is used as the auxiliary graph to de-anonymize Mendeley dataset.

Table 3 give the statistics of datasets.

TABLE 3 STATISTICS OF DATASETS USED IN THE EXPERIMENT

Statistics	|V|	|E|
Aminer	121,135	163,478
Mendeley	105,365	143,742
MAG	95,479	136,125
DBLP	90,182	127,353
MAG-Aminer	85,141	114,631
DBLP-Mendeley	82,962	104,336

B. Comparison Methods

We use true positive rate (TPR) and false positive rate (FPR) to evaluate the performance of proposed de-anonymization method. TPR measures the proportion of positives that are correctly identified in all matched node pairs. FPR is calculated as the ratio between the number of negative node pairs that wrongly categorized as positive to the total number of actual wrongly categorized as positive node pairs and matched node pairs. The receiver operating characteristic (ROC) curve is created by plotting TPR against FPR at various threshold settings.

We compare the proposed method with below existing state-of-the-art methods for social networks de-anonymization.

NS Method: This de-anonymization algorithm [20] is based only on network topologies. As a feedback-based attack method, it has self-enhanced learning ability and the identified nodes are added to the auxiliary information of the attacker, so that the auxiliary graph becomes larger. The method is robust even if the anonymous network and the auxiliary network have small overlaps. It works well in large social networks.

Random Forest: Sharad and Danezis [34] propose a generic automatic de-anonymization method for de-anonymizing social networks. This method employs a machine learning method (random decision forest) to match pairs of different anonymous subgraphs for evaluating the effect of anonymous techniques quickly. Being an automated method, it achieves a higher classification of positive rate and lower negative rate.

ADA: Ji et al. [46] design a joint similarity measure that takes into account the local and global topology features of the data as well as the information obtained from the auxiliary data, it also inherits useful information from the de-anonymization results. They propose a joint de-anonymization (DA) scheme, which ensures the accuracy of the de-anonymizing social graph data. They extended DA to an adaptive de-anonymize method (ADA) for resolving the de-anonymization problem where there are unknown overlaps between the auxiliary data and the anonymous data. By matching the two core subgraphs, ADA has achieved a high de-anonymizing accuracy and reduced the computational complexity.

Seed-and-Grow: Peng et al. [27] propose the Seed-and-Grow algorithm, which identifies users in anonymous graphs based on the graph structure. This algorithm first identifies a seed subgraph (which is pre-implanted by the attacker or by the user conspired), and then the attacker based on the existing user's social relationships to extend this seed subgraph. It has less assumptions than other methods, reduces the number of parameters and improves the recognition of validity and accuracy.

DDM: Fabiana et al. [22] propose a degree-driven graph matching (DDM) method, which is a rigorous mathematical analysis method for social network de-anonymization that based on the important features of power distribution in complex networks. Using the original graph partition method, they prove that nodes with large degrees must be identified as prior knowledge in order to successfully identify social network user accounts.

C. Experimental Results

We implemented our algorithm in C++ language, which is conducted on a cluster with Intel Xeon E5-2620 V3 CPU, NVIDIA Tesla K80 GPU, Intel Xeon Phi 7120P, 128 GB main memory, 1T SSD, 6T SAS disk, and CentOS release 6.4.

1) Analysis of Degree Distribution in Datasets

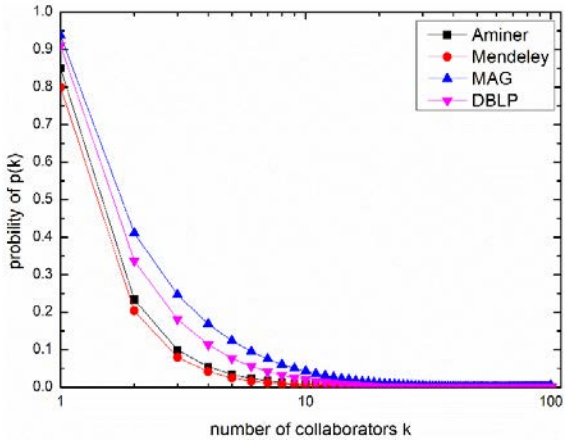

Figure 3 Probability of scientist's collaborators in four of the datasets studied in the experiment.

We investigate the degree distribution of four datasets as shown in Figure 3. We can see that the degree of four datasets generally follows a heavy-tailed distribution, which is consistent with Newman's conclusion [47]. Thus, the degree distribution is an important feature of de-anonymizing social networks in our solution.

2) Parallel Scalable

Since our solution can be parallelized in node mapping in different subgraphs, we parallel our algorithm with multi-processors. We vary the number *n* of processors from 4 to 20 for Aminer, Mendeley, MAG, DBLP, MAG-Aminer, and DBLP-Mendeley datasets. The algorithms generated up to 300 patterns to be verified. As shown in Figure 4, our method scales well with the increase of processors. The improvement is 3 times when *n* increases from 4 to 20 for the Mendeley dataset. With 20 processors, our method takes 301, 270, 241, 199, 132, 112 seconds on Aminer, Mendeley, MAG, DBLP, MAG-Aminer, and DBLP-Mendeley, respectively. Therefore, our solution is parallel scalable, and it is 3.7 times faster on average when *n* increases from 4 to 20 on real-world networks.

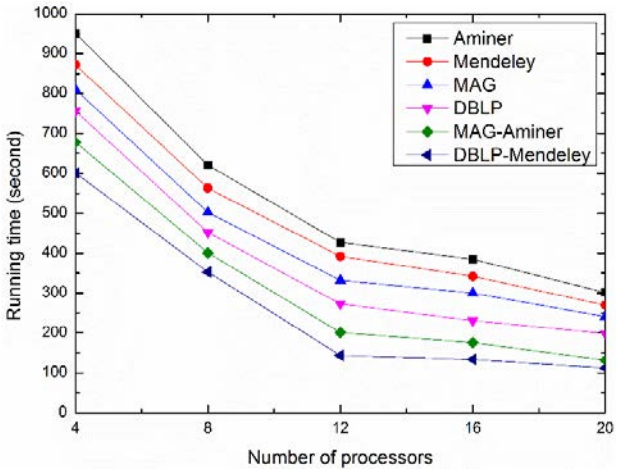

Figure 4 The running time of our solution with different processors in 6 datasets.

3) Comparison to State-of-the-art Algorithm

Figure 5 and figure 6 shows the ROC curves of the six algorithms on the Aminer and MAG data, from which we can clearly see that our method achieves the best performance. Figure 5 shows the performance of our solution for de-anonymizing the Aminer dataset. Under the false positive of 0.7%, our algorithm can de-anonymize 84% graph nodes. The DDM, ADA, Seed-and-Grow, Random Forest, and NS methods can de-anonymize 67%, 62%, 55%, 47%, 37% user nodes, respectively. Even for very small false positive (0.5%), our method can still de-anonymize 81% of the nodes. Figure 6 displays the ROC of our solution for de-anonymizing the MAG dataset. Under the false positive of 0.7%, our algorithm can de-anonymize 84% user nodes. The DDM, ADA, Seed-and-Grow, Random Forest, and NS methods can de-anonymize 69%, 56%, 53%, 44%, 29% user nodes, respectively. Even for very small false positive (0.5%), our method can still de-anonymize 82% of the nodes.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2756904, IEEE Access

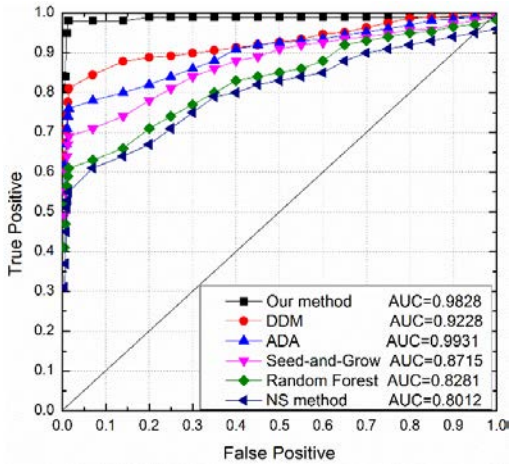

Figure 5. Receiver operating characteristic curve de-anonymizing with the Aminer dataset.

4) De-anonymization Across Networks

In this section, we compare our method to state-of-the-art methods with MAG-Aminer and DBLP-mendeley datasets. These two datasets are different from the other four datasets mentioned above. The two datasets come from different networks. They are not a snapshot of same network, and the network structures are different. The AUC of the experiments are displayed in Figure 7. It is clear to see that our method achieves the best AUC from Figure 7. The AUC of our solution is 54% and 19% higher than NS and DDM, respectively. This result shows that our method works well across networks.

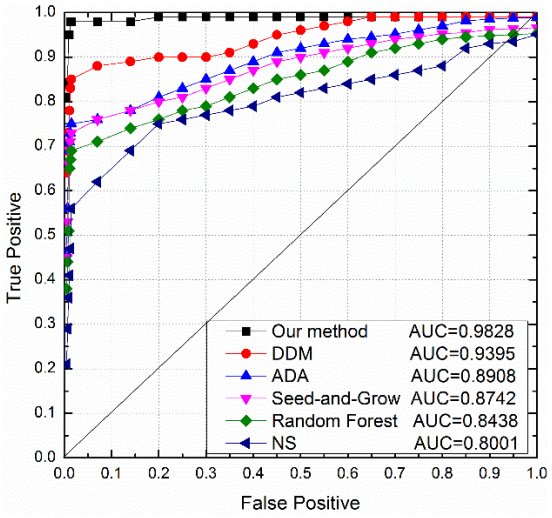

Figure 6. Receiver operating characteristic curve de-anonymization with the MAG dataset.

5) Robustness Evaluation

We examine the robustness of the proposed method by adding noisy data (delete or add edges to perturb the edge data). We can see the degree distribution follows power-law from the results of the experiment in part C of Section V. The Barabási-Albert(BA) [48] preferential attachment model describes this feature. Therefore, we use the BA model to add edges to simulate the noisy data in the dataset. Results are shown in Figure 8.

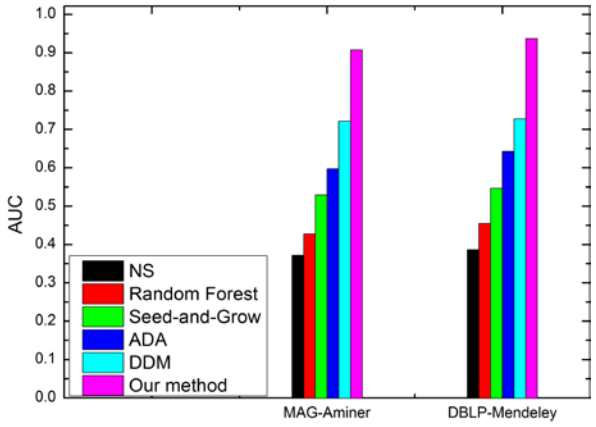

Figure 7. AUC de-anonymizing in MAG-Aminer and DBLP-mendeley datasets with different methods

In the experiment, we add noise to the anonymous graph and the auxiliary graph, respectively. In order to add p to the anonymous data, we randomly add $(p/2)\times|E|$ links of to the anonymous graph and delete the existing link for $(p/2)\times|E|$ (in this case a node may become an isolated node). We use the PA model to add edges. For example, in Figure 8, 20% noise means that we add 10% of edges and delete 10% of existing edges to the anonymous graph. We can see that the proposed de-anonymization scheme is robust to noisy data. Even if we change 25% links in the anonymous graph, we achieve AUC of 80.2%, 80.9%, 80.6%, 80.1%, 81.7% on Aminer, Mendeley, MAG, DBLP, MAG-DBLP, and Aminer-Mendeley datasets, respectively. Note that when 25% links are changed, the structure of the anonymous graph is significantly changed. In practice, if the majority of the published anonymous graph's structure is changed, the usefulness of data will be reduced dramatically. Therefore, the data publisher will not change the structure dramatically for data integrity. In summary, our solution is robust to the noisy data.

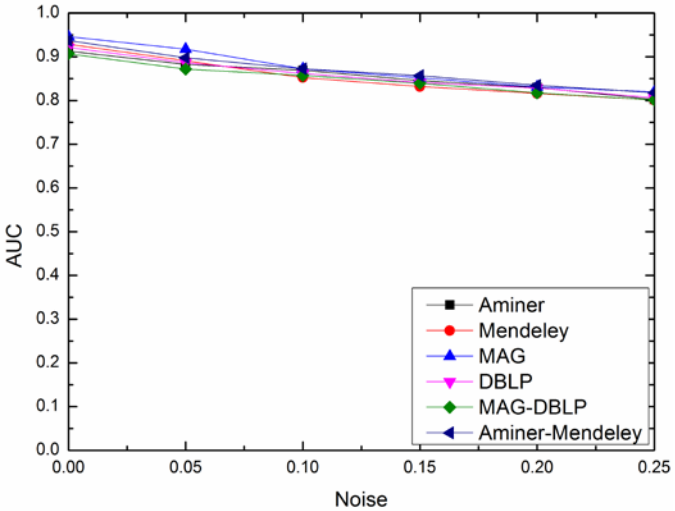

Figure 8. The AUC of the proposed method with different noise rates

D. Discussions

Random forests are able to handle high-dimensional data and do not need to select features. Nonetheless, the random forest

classification will be overfitting to large noisy data. For attributes with different data values, more values imply greater impact on the random forest. Therefore, the random forest classification could not be credible in high-dimensional data.

Classifier training and classification vary for different tasks. The algorithms (whether being anonymous algorithms or training algorithms) predict whether the test node matches the label, so it not only learns the features of the data, but also learns the features of anonymous methods. It simply assumes that the attack is invalid if the node degree is randomized based on the degree distribution of the node's neighborhood. The success of classification is not based on the invariance of node degree but rely on the inconsistency of anonymous strategy. The strategy function can be learned and used to attack its anonymous schemes. Besides degree, clustering coefficient and eigenvector, we can also employ global features, such as betweenness centrality and closeness centrality, to match node pairs. Designing a more efficient graph partition method is the key to reduce time complexity. In addition, how to find a proper strategy to choose good auxiliary graph is another problem to be considered.

VI. CONCLUSION

In this paper, we propose a valid and robust solution to solve the social network de-anonymization problem. We transform the de-anonymization problem into machine learning classification problem and then solve the problem with random forest classifier. Structural features such as degree centrality, clustering coefficient and eigenvector are used to classify node pairs. To de-anonymize large-scale social graph, we partition large graph into subgraphs with spectral partition method. Our solution does not need any seed node pair in the whole de-anonymize process. The validity of the algorithm is verified on the real datasets. Noise data are added during the de-anonymize process, and the result shows that proposed algorithm has good robustness. We speculate that the use of node embedding and network structure embedding methods may be more suitable for training the random forest classifier, and we will do corresponding research in the future work. Additionally, user profile attribute may be a useful de-anonymization feature, which will be one of the key focus point in the future work.

REFERENCES

[1] R. B. Shapiro and P. N. Ossorio, "Regulation of Online Social Network Studies," *Science (80-.).*, vol. 339, no. 6116, p. 144 LP-145, Jan. 2013.

[2] S. Aral and D. Walker, "Identifying Influential and Susceptible Members of Social Networks," *Science (80-.).*, vol. 337, no. 6092, p. 337 LP-341, Jul. 2012.

[3] D. Centola, "The Spread of Behavior in an Online Social Network Experiment," *Science (80-.).*, vol. 329, no. 5996, p. 1194 LP-1197, Sep. 2010.

[4] L. Cui, F. R. Yu, and Q. Yan, "When big data meets software-defined networking: {SDN} for big data and big data for {SDN}," *{IEEE} Netw.*, vol. 30, no. 1, pp. 58–65, 2016.

[5] L. Cui, L. Sun, X. Fu, N. Lu, and G. Zhang, "Exploring {A} Trust Based Recommendation Approach for Videos in Online Social Network," *Signal Process. Syst.*, vol. 86, no. 2–3, pp. 207–219, 2017.

[6] R. Gross and A. Acquisti, "Information Revelation and Privacy in Online Social Networks," in *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society*, 2005, pp. 71–80.

[7] T. Jung, X.-Y. Li, and M. Wan, "Collusion-tolerable privacy-preserving sum and product calculation without secure channel," *IEEE Trans. Dependable Secur. Comput.*, vol. 12, no. 1, pp. 45–57, 2015.

[8] K. Liu and E. Terzi, "Towards Identity Anonymization on Graphs," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, 2008, pp. 93–106.

[9] S. Ji, W. Li, M. Srivatsa, and R. Beyah, "Structural Data De-anonymization: Quantification, Practice, and Implications," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014, pp. 1040–1053.

[10] M. Zabielski, R. Kasprzyk, Z. Tarapata, and K. Szkółka, "Methods of Profile Cloning Detection in Online Social Networks," in *MATEC Web of Conferences*, 2016, vol. 76, p. 4013.

[11] R. Zafarani and H. Liu, "Connecting Users Across Social Media Sites: A Behavioral-modeling Approach," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013, pp. 41–49.

[12] A. Malhotra, L. Totti, W. M. Jr., P. Kumaraguru, and V. Almeida, "Studying User Footprints in Different Online Social Networks," in *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2012, pp. 1065–1070.

[13] S. Tan, Z. Guan, D. Cai, X. Qin, J. Bu, and C. Chen, "Mapping Users across Networks by Manifold Alignment on Hypergraph," in *28th AAAI Conference on Artificial Intelligence*, 2014, pp. 159–165.

[14] R. Zafarani and H. Liu, "Connecting Corresponding Identities across Communities," in *International Conference on Weblogs and Social Media, Icwsm 2009, San Jose, California, Usa, May*, 2009.

[15] D. Perito, C. Castelluccia, M. A. Kaafar, and P. Manils, "How Unique and Traceable Are Usernames?," in *Privacy Enhancing Technologies: 11th International Symposium, PETS 2011, Waterloo, ON, Canada, July 27-29, 2011. Proceedings*, S. Fischer-Hübner and N. Hopper, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1–17.

[16] J. Liu, F. Zhang, X. Song, Y.-I. Song, C.-Y. Lin, and H.-W. Hon, "What's in a Name?: An Unsupervised Approach to Link Users Across Communities," in *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, 2013, pp. 495–504.

[17] T. Iofciu, P. Fankhauser, F. Abel, and K. Bischoff, "Identifying Users Across Social Tagging Systems," in *International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July*, 2010.

[18] Y. Zhang, J. Tang, Z. Yang, J. Pei, and P. S. Yu, "COSNET: Connecting Heterogeneous Social Networks with Local and Global Consistency," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 1485–1494.

[19] W. Cohen, P. Ravikumar, and S. Fienberg, "A comparison of string metrics for matching names and records," in *Kdd workshop on data cleaning and object consolidation*, 2003, vol. 3, pp. 73–78.

[20] A. Narayanan and V. Shmatikov, "De-anonymizing Social Networks," in *Security and Privacy Symposium on IEEE*, 2009, pp. 173–187.

[21] E. Kazemi, S. H. Hassani, and M. Grossglauser, "Growing a Graph Matching from a Handful of Seeds," *Proc. VLDB Endow.*, vol. 8, no. 10, pp. 1010–1021, 2015.

[22] C. Fabiana, M. Garetto, and E. Leonardi, "De-anonymizing scale-free social networks by percolation graph matching," in *2015 IEEE*

Conference on Computer Communications (INFOCOM), 2015, pp. 1571–1579.

[23] L. Sweeney, "k-anonymity: a model for protecting privacy," *Int. J. Uncertainty, Fuzziness Knowledge-Based Syst.*, vol. 10, no. 5, pp. 557–570, 2002.

[24] C.-H. Tai, P. S. Yu, D.-N. Yang, and M.-S. Chen, "Structural Diversity for Privacy in Publishing Social Networks," in *Proceedings of the 2011 SIAM International Conference on Data Mining*, pp. 35–46.

[25] G. Wang, Q. Liu, F. Li, S. Yang, and J. Wu, "Outsourcing privacy-preserving social networks to a cloud," in *2013 Proceedings IEEE INFOCOM*, 2013, pp. 2886–2894.

[26] D. Proserpio, S. Goldberg, and F. McSherry, "Calibrating Data to Sensitivity in Private Data Analysis: A Platform for Differentially-private Analysis of Weighted Datasets," *Proc. VLDB Endow.*, vol. 7, no. 8, pp. 637–648, 2014.

[27] W. Peng, F. Li, X. Zou, and J. Wu, "A Two-Stage Deanonymization Attack against Anonymized Social Networks," *IEEE Trans. Comput.*, vol. 63, no. 2, pp. 290–303, 2014.

[28] N. Korula and S. Lattanzi, "An Efficient Reconciliation Algorithm for Social Networks," *Proc. VLDB Endow.*, vol. 7, no. 5, pp. 377–388, 2014.

[29] P. Pedarsani, D. R. Figueiredo, and M. Grossglauser, "A Bayesian method for matching two similar graphs without seeds," in *2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2013, pp. 1598–1607.

[30] J. Qian, X. Y. Li, C. Zhang, and L. Chen, "De-anonymizing social networks and inferring private attributes using knowledge graphs," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, 2016, pp. 1–9.

[31] M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weis, "Resisting Structural Re-identification in Anonymized Social Networks," *Proc. VLDB Endow.*, vol. 1, no. 1, pp. 102–114, 2008.

[32] G. Wondracek, T. Holz, E. Kirda, and C. Kruegel, "A Practical Attack to De-anonymize Social Network Users," in *Security & Privacy*, 2010, pp. 223–238.

[33] A. Narayanan, E. Shi, and B. I. P. Rubinstein, "Link prediction by de-anonymization: How We Won the Kaggle Social Network Challenge," in *International Joint Conference on Neural Networks*, 2011, pp. 1825–1834.

[34] K. Sharad and G. Danezis, "An Automated Social Graph De-anonymization Technique," in *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, 2014, pp. 47–58.

[35] T. Opsahl, F. Agneessens, and J. Skvoretz, "Node centrality in weighted networks: Generalizing degree and shortest paths," *Soc. Networks*, vol. 32, no. 3, pp. 245–251, 2010.

[36] L. C. Freeman, "Centrality in social networks conceptual clarification," *Soc. Networks*, vol. 1, no. 3, pp. 215–239, Jan. 1978.

[37] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, p. 440, 1998.

[38] M. E. J. Newman, "The mathematics of networks," *new palgrave Encycl. Econ.*, vol. 2, no. 2008, pp. 1–12, 2008.

[39] A. Buluç, H. Meyerhenke, I. Safro, P. Sanders, and C. Schulz, "Recent Advances in Graph Partitioning," in *Algorithm Engineering: Selected Results and Surveys*, L. Kliemann and P. Sanders, Eds. Cham: Springer International Publishing, 2016, pp. 117–158.

[40] A. Liaw and M. Wiener, "Classification and regression by randomForest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[41] C. Strobl, J. Malley, and G. Tutz, "An introduction to recursive partitioning: rationale, application, and characteristics of classification and regression trees, bagging, and random forests.," *Psychol. Methods*, vol. 14, no. 4, p. 323, 2009.

[42] R. J. Yang and G. L., "Experience with approximate reliability-based optimization methods," *Struct. Multidiscip. Optim.*, vol. 26, no. 1, pp. 152–159, 2004.

[43] A. Sinha, Z. Shen, Y. Song, H. Ma, D. Eide, B. P. Hsu, and K. Wang, "An overview of microsoft academic service (mas) and applications," in *Proceedings of the 24th international conference on world wide web*, 2015, pp. 243–246.

[44] R. D. L. Zaugg Holt, E. West Richard, Tateishi Isaku, "Mendeley: Creating Communities of Scholarly Inquiry Through Research Collaboration," *TechTrends*, vol. 55, no. 1, pp. 32–36, 2011.

[45] J. Tang, "AMiner: Toward Understanding Big Scholar Data," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, 2016, p. 467.

[46] S. Ji, W. Li, M. Srivatsa, J. S. He, and R. Beyah, "Structure Based Data De-Anonymization of Social Networks and Mobility Traces," in *Information Security: 17th International Conference, ISC 2014, Hong Kong, China, October 12-14, 2014. Proceedings*, S. S. M. Chow, J. Camenisch, L. C. K. Hui, and S. M. Yiu, Eds. Cham: Springer International Publishing, 2014, pp. 237–254.

[47] M. E. J. Newman, "Scientific collaboration networks. I. Network construction and fundamental results," *Phys. Rev. E*, vol. 64, no. 1, p. 16131, 2001.

[48] A.-L. Barabási and R. Albert, "Emergence of Scaling in Random Networks," *Science (80-.).*, vol. 286, no. 5439, p. 509 LP-512, Oct. 1999.