IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received September 8, 2019, accepted September 29, 2019, date of publication October 8, 2019, date of current version October 31, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2946179

An Efficient Social Attribute Inference Scheme Based on Social Links and Attribute Relevance

JIAN MAO [1], (Member, IEEE), WENQIAN TIAN [2], YITONG YANG [1,3], AND JIANWEI LIU [1]

[1] School of Cyber Science and Technology, Beihang University, Beijing 100191, China
[2] School of Electronic and Information Engineering, Beihang University, Beijing 100191, China
[3] Shanghai Key Laboratory of Integrated Administration Technologies for Information Security, Shanghai 200240, China

Corresponding author: Jian Mao (maojian@buaa.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB0802400, in part by the National Natural Science Foundation of China under Grant U11733115, Grant 61571010, and Grant 61871023, and in part by the Opening Project of Shanghai Key Laboratory of Integrated Administration Technologies for Information Security under Grant AGK2019001.

ABSTRACT Social network is a critical component in mobile multimedia systems, where users share their videos, photos, and other media. However, the information (e.g., posts, user profiles, etc.) shared on the social network platforms usually reflects many users' personal (private) information, which could be mined and abused for malicious purposes. To address privacy concerns, many social network service providers adopted privacy-preserving mechanisms, e.g., anonymizing user identity, hiding users' profiles, etc. As a result, the attributes in user profiles are usually set up to be accessed only by friends to prevent privacy leakage. Several attacks have been proposed to infer the hidden attributes to Several the efficiency of current privacy-protecting mechanisms. Most of these solutions are based on the social links among users or their behaviors. In this paper, we systematically analyze the social features related to user privacy inference and found that there are relevances among social attributes, which has a great impact on inferring users' hidden attributes. According to our findings, we propose an efficient social attribute inference scheme based on social links and attribute relevance properties. We develop a relevance attribute inference method (ReAI) using random walks with restart. We analyze attribute relevance on inference performance and use Kulczynski measure to quantify attribute relevance as edge weights of attribute nodes in an improved social-attribute network. We evaluate our method and compare it with the traditional attribute inference method. The results show that our method performs better than the traditional method. We also use Kulczynski measure and Information Gain Ratio to evaluate the improvements. The results show that the bigger relevance between attributes contributes to higher improvements.

INDEX TERMS Social network, mobile multimedia platform, attribute inference, privacy preserving, attribute relevance.

I. INTRODUCTION

Online Social networks (OSNs) are important applications of mobile multimedia systems, which provide platforms for users to share their photos, music, and videos, etc. The advancement of mobile media has greatly boosted OSNs with attractive contents. It further amplified the problem that OSNs have huge amounts of personal information of their users, including social links, online behaviors and social attributes (e.g., gender, birthday, affiliation, education stage, etc.) [1]. Attribute information plays a significant role in social data applications, especially in personalized advertisements and product recommendations. As a result, many efforts have been devoted to analyzing and utilizing users' social attributes although they are hidden by users themselves or OSN service providers due to privacy concerns [2]. On the other side, this is also necessary to conduct attribute inference to analyze the effectiveness of the current privacy-preserving approaches and disclose the vulnerabilities of attribute protection policies adopted in OSNs.

The attribute inference attack discloses users' private social attributes from their public attributes and other auxiliary information, such as social links or online behaviors. There are two main categories of attribute inference approaches. One category of such approaches infers users' unknown attributes based on social links, e.g., friend or following

The associate editor coordinating the review of this manuscript and approving it for publication was Danda Rawat.

relationships. Bhagat *et al.* [3] infer attributes via majority voting by local neighbors. Yin *et al.* [4] propose link recommendation and attribute ranking using random walks with restart. They construct a social-attribute network (SAN) using the social graph to perform attribute inference. These methods are based on the assumption that people in the same community share the same interests and backgrounds. The other category infers unknown attributes from user behaviors [5], which assumes that people with similar interests and backgrounds will behave similarly. Gong and Liu [6] propose an attribute inference attack based on vote distribution using both social friends and behaviors. They construct a social-attribute-behavior network (SAB) performing random walks on the graph.

Most of these solutions are based on the social links among users or their behaviors [7], [8], which is just one aspect affected by users' social attributes. In fact, there are still some other new dimensions could be used to disclose users' social attributes.

A. OUR OBSERVATION

To disclose the new dimensions that can be used to improve the efficiency of the attribute inference, we identify that there are relevances among social attributes, which can be used to improve the effectiveness of inferring users' hidden attributes. For example, the attribute *gender* may have relevance to the attribute *job*, as the population distribution of job over gender is unbalanced: according to InfoQ's statistic [9], the proportion of male programmer to female programmer is around 9 : 1. Therefore, when predicting users' job, a male has more possibility to be a programmer than a female. That is to say, the relevance among different attributes does affect the population distribution of different attribute values resulting in performance influences on attribute inference approaches. To quantify the impact of the attribute relevance, we conduct a correlation analysis among different itemsets (attribute values). Empirically, in pattern evaluation of data mining, **Kulczynski** measure is a wildly used metric for presenting pattern relationships among itemsets [10] precisely. Accordingly, we use **Kulczynski** measure to analyze attribute relevance among different attribute values. We constructed a social graph and embedded the attribute relevance values into the social graph as edge weights among different attribute nodes. We observed that considering attribute relevance in attribute inference procedures would achieve better performances.

B. OUR WORK

In this paper, we propose a relevance attribute inference method (*ReAI*) using random walks based on a social graph. We first make correlation analysis among attribute itemsets using **Kulczynski** measure and then improve the SAN model by considering attribute relevance to form a social attribute-relevance network (SRAN). We add edges between attribute nodes in the SAN graph and define the weight of each edge in the SRAN graph, endowing them with different meanings. Secondly, we perform attribute inference based on the SRAN model. We use random walks with restart to make inference propagation for a target node. We use public information to quantify the closeness of attributes to represent the relevance of them and simplified the weights of edges to calculate the transition matrix. Thirdly, we evaluate and compare our method with the attribute inference method proposed by Yin *et al.* [4] using a Facebook dataset. The attribute ranking method they proposed can be used to infer attributes, which we call *TAI* in the following sections. We make several experiments to prove the effectiveness of ReAI. And we use *Information Gain Ratio* to evaluate the performance affected by attribute relevance. We observe that the closer relevance attributes have, the better our method performs than TAI.

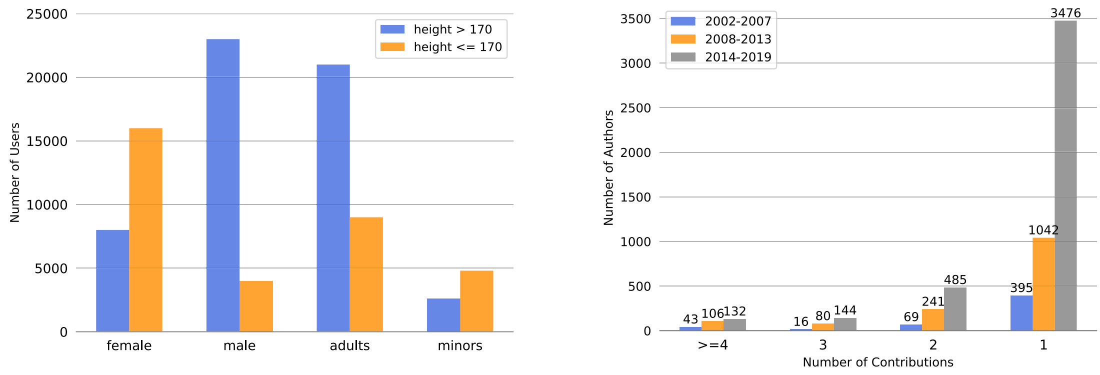
In summary, we made the following contributions in this paper.

- We evaluate relevance among attribute values using **Kulczynski** measure and improve the traditional SAN model by considering attribute relevance, adding edges between attribute nodes to form a new SRAN model. And we define different weights of edges to endow different meanings for links in the graph.
- We design a relevance attribute inference method (ReAI) based on the improved SRAN model to perform attribute inference by utilizing random walks. And we use public information to quantify relevance between attribute nodes.
- We demonstrate the effectiveness of our ReAI method and compare it with TAI method on a Facebook dataset by making inference on different attribute pairs, and further, we evaluate the performance and improvement of attribute relevance using *Information Gain Ratio*.

Paper Organization: The rest of this paper is organized as follows. Section 2 introduces the relevance analysis and attribute inference problem definition. Section 3 presents our main algorithm. Section 4 presents the evaluation results and analysis. We discuss closely related work in Section 5 and conclude the paper in Section 6.

II. RELEVANCE ANALYSIS AND PROBLEM DEFINITION

There exists relevance among some attribute values. For example, intuitively the attribute *gender* has relevance to the attribute *body height*, as men are usually taller than women. So, if the height of a user is 164cm, the user is more likely to be a woman. Therefore, *height* may help to infer a user's *gender*. The attribute *age* also has relevance to the attribute *height*, as adults ($\geq$18 years old) are usually taller than minors ($<$18 years old). So *height* may also help to infer whether a user is an adult or a minor. We randomly choose 1 million users of an online social network **Pokec** to make statistics of their gender, age and body height [11]. We make an analysis of *gender* and *height* and an analysis of *age* and *height*, respectively. After filtering the invalid data, we collect 508,996 users' gender and body height and 371,188 users' age and body height.



(a) Statistics of Pokec user age, gender and body height

(b) Statistics of DBLP users, publications, and years

FIGURE 1. **Relevance investigation on Pokec and DBLP datasets.**

The result is shown in Figure 1(a). The figure shows that most men are taller than 170cm and most women are shorter than 170cm, which means that *height* has statistical relevance to *gender*. According to the statistics, most minors are shorter than 170cm and most adults are taller than 170cm. Therefore, *height* is also related to *age*. Besides, we conduct investigation on the DBLP dataset, which maintains authors' publication information. We choose several common items in the dataset, such as authors, journals to analyze the relevance among them. The investigation result is shown in Figure 1(b). According to the statistical result, it could be concluded that authors that contribute to the journal of "*Computers in Human Behavior*" in the year 2012-2015 are more likely to contribute to this journal in the year 2016-1019 than the authors who contributed to the journal during the year 2004-2007 and 2008-2011.This shows that the author's activity attribute and the year attribute do have some relevance.

Besides, we conduct investigation on the DBLP dataset, which maintains authors' publication information. We choose several common items in the dataset, such as authors, journals to analyze the relevance among them. Figure 1(b) shows the investigation result of the number of contributions to the journal of "*Computers in Human Behavior*" with respect to the publication years. According to the statistical result, authors have one contribution to the journal rather than three, the proportion of authors contributing from 2014 to 2019 is higher. It means that if an author has no less than four contributions to this journal, he is more likely to contribute to this journal earlier than the authors who have only one contributions to the journal. This discloses that the number of author's contributions and the publication years are relevant.

A. RELEVANCE ANALYSIS

The aforementioned statistics implicates that there are relevance among the social attributes that can be use to facilitate attribute inference.

Kulczynski measure (abbreviated as **Kulc**) is an efficient metric to quantify the intrinsic pattern relationships [10]. Given two itemsets A and B, the **Kulczynski** measure of A and B is defined as

$$Kulc(A, B) = \frac{1}{2}(P(A|B) + P(B|A)). \tag{1}$$

It denotes the average of two confidence measures of two relevant patterns, "$A \Rightarrow B$" and "$B \Rightarrow A$", where $P(A|B)$ denotes the confidence of pattern "$B \Rightarrow A$" ($confidence(B \Rightarrow A)$) and $P(B|A)$ denotes the confidence of pattern "$A \Rightarrow B$" ($confidence(A \Rightarrow B)$). For example, the relevance between the purchase of two items, *milk* and *coffee*, can be evaluated by calculating the purchase history of the two items. Suppose T is the set of whole transactions, T_c is the set of the transactions bought *coffee*, T_m is the set of transactions including *milk*, set $T_{m\overline{c}}$ is the set of transections that bought *milk* without item *coffee*, $T_{\overline{m}c}$ consists of the transactions that bought *coffee* without *milk*, and T_{mc} is the transaction set that bought both items. Obviously, $T_m = T_{m\overline{c}} + T_{mc}$ and $T_c = T_{\overline{m}c} + T_{mc}$ hold. The relevance of item *coffee* and *milk* measured by *Kulc* is calculated as follows.

$$\begin{aligned} Kulc(milk, coffee) &= \frac{1}{2}(P(milk|coffee) + P(coffee|milk)) \\ &= \frac{1}{2}(\frac{|T_{mc}|}{|T_c|} + \frac{|T_{mc}|}{|T_m|}), \end{aligned} \tag{2}$$

where $|T|$ represents the number of entries of the set T.

We conduct the **Kulc** measure to evaluate the relevance between attribute items. For example, attribute A has 3 different values, namely a_1, a_2, a_3, and attribute B has 2 different values, namely b_1, b_2. Let $N_{a_ib_j}$ denote the number of users that has attribute value a_i and b_j, N_{a_i} and N_{b_j} denote the number of users that has attribute a_i, b_j, respectively.

The relevance of attribute value a_i and b_j can be defined as

$$\begin{aligned} Kulc(a_i, b_j) &= \frac{1}{2}(P(a_i|b_j) + P(b_j|a_i)) \\ &= \frac{1}{2}(confidence(b_j \Rightarrow a_i) + confidence(a_i \Rightarrow b_j)) \\ &= \frac{1}{2}(\frac{N_{a_ib_j}}{N_{bj}} + \frac{N_{a_ib_j}}{N_{a_i}}). \end{aligned} \tag{3}$$

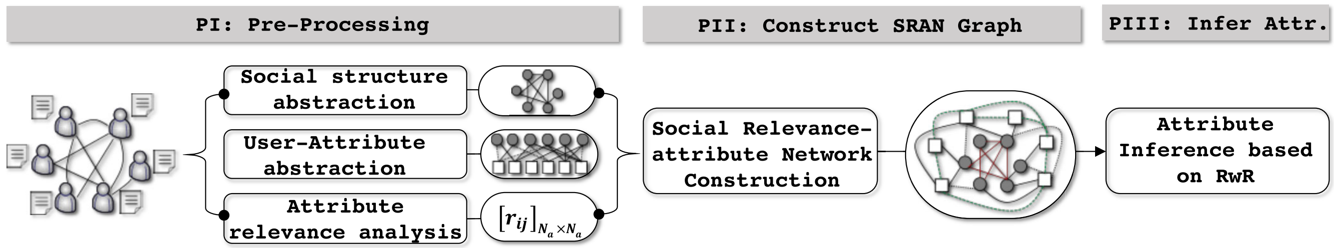


FIGURE 2. Overall architecture of our approach.

Similarly, we use **Kulc** measure to quantify relevance among attribute values and then take the relevance factor into consideration when we conduct attribute inference.

B. ATTRIBUTE INFERENCE PROBLEM

We use an undirected[1] graph $G_s = (V_u, E_u)$ to represent a social network, where nodes in V_u represent users and E_u represent the relationship between user nodes in V_u.

For each node v in V_u, we use a attribute vector $\vec{a_v}$ to represent attribute values of v. An attribute may have multiple values, e.g., gender could be female or male. So each dimension of $\vec{a_v}$ represent a attribute value, $\vec{a_v} = [a^v_{11}, \ldots, a^v_{1m_1}, \ldots, a^v_{i1}, \ldots, a^v_{im_i} \ldots, a^v_{n1}, \ldots, a^v_{nm_n}]$, where n denotes the classes of attribute and $m_{i,i\in n}$ denotes the number of different values of the ith attribute. We denote the number of all distinct attribute values $N_a = |\vec{a_v}|$ as the dimension of $\vec{a_v}$. The jth entry of $\vec{a_v}$ equals to 1 when v has the jth attribute value, -1 when v does not have it and 0 when it is unknown. Then we can denote the node attribute matrix as $A = [\vec{a_1}, \vec{a_2}, \ldots, \vec{a_N}]$ for all nodes, where $N = |V_u|$. Attribute inference is to infer the unknown values in attribute vector $\vec{a_v}$ for each $v \in V_u$, that is, for $\forall a^v_{ij} = 0 \in A$, determine the value $a^v_{ij} \in \{-1, 1\}$.

The description above constructs a traditional attribute inference problem. As we mentioned in Section I, we concentrate on the attribute relevance among these attribute values and the corresponding influences on the attribute inference. To describe the relevance among the attribute values precisely, we define an attribute adjacency matrix $R = \{r_{ij}\}_{N_a \times N_a}$. It is an $N_a \times N_a$ matrix, where $N_a = \Sigma^n_{i=1} m_i$, and an entry $r_{ij} \in R$ represents the relevance degree between the ith attribute value and the jth attribute value. Note that in the attribute adjacency matrix, we consider the relevance between different attributes and the relevance of attribute values belong to one attribute does not been considered, which means that the corresponding entry in R equals to ∞.

*Definition 1 (**Attribute Inference Problem**):* Given $T = (G_u, A, R, V_t)$ as a snapshot of a social network G_u with nodes' attribute matrix A, attribute adjacency matrix R and a list of target users V_t. **Attribute inference** is to determine the attribute vectors $\vec{a_v}$ for all $v \in V_t$, i.e., to replace all the 0-value entries of $\vec{a_v}$ with 1 or -1.

[1] Our model and algorithm can also generalized to directed graph

To solve this problem, there are several challenges should be addressed.

- **Challenge I**: How to quantify the relevance among the attribute values?
- **Challenge II**: How to aggregate the attribute relevance information into the traditional attribute inference structures?
- **Challenge III**: How to conduct the attribute inference efficiently based on the newly developed data structure?

C. APPROACH OVERVIEW

Our goal is to design an attribute inference framework considering attribute relevance to solve the attribute inference problem for real-world, large-scale networks. As shown in Figure 2, to address the aforementioned challenges, our approach consists of three major phases: *Pre-processing*, *Construct SRAN Graph*, and *Infer Attribute*.

Phase I: Pre-processing takes as input the social data, and there are three components, *social structure abstraction* (PI-ss), *user-attribute abstraction* (PI-ua), and *attribute relevance analysis* (PI-ar). PI-ss is to extract the social links among the users and outputs the social node (user) graph G_s. PI-ua is to establish the mapping between users (social node) and the social attribute values and outputs the attribute matrix A. PI-ar measures the relevance between two attribute values and outputs the attribute adjacency matrix R.

Phase II: Construct SRAN Graph takes as inputs the social graph G_s, attribute matrix A and attribute adjacency matrix R, and outputs a social relevance attribute network (SRAN) graph. SRAN graph has two kinds of nodes, social nodes (labeled as circle nodes) and attribute nodes (labeled as rectangle nodes), where social nodes represent users and attribute nodes are attribute values included in the target social network. Accordingly, we defined three types of edges to describe relationships among these nodes. Specifically, social edges represent the social links between two social nodes; a user-attribute edge between a social node and an attribute node is determined by whether the social node has this attribute value; an attribute relevance edge is weighted by the correlation between two attribute values, (i.e., attribute node in our SRAN), which is quantified by *PI-ar* in Phase I.

Phase III: Infer Unknown Attribute takes the SRAN Graph obtained from Phase II as the inputs, conducts random

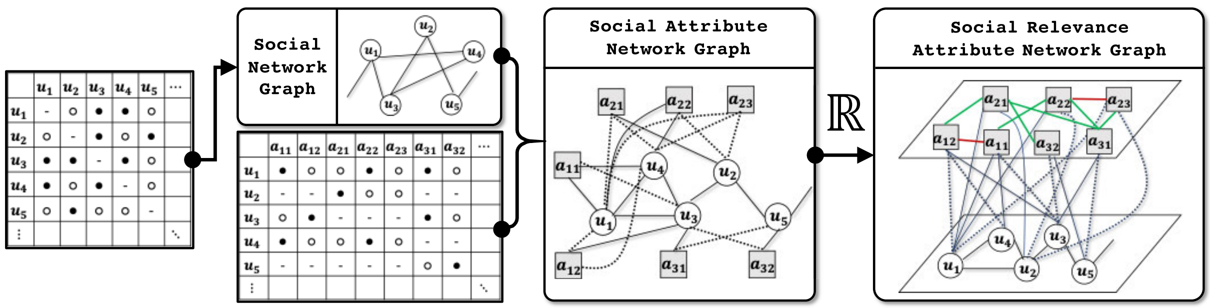


FIGURE 3. SRAN graph construction procedure.

walk with restart (RwR) to perform relevance-based attribute inference and outputs all the unknown user-attribute links of the target users in V_t.

III. RELEVANCE ATTRIBUTE INFERENCE

In this section, we present our attribute inference approach based on social links and attribute relevance. We describe the design of the social relevance attribute network (SRAN) graph in Section III-A, and illustrate social link and relevance based attribute inference algorithm in Section III-B. A brief analysis of the time complexity of the proposed algorithm is discussed in Section III-C.

A. SRAN GRAPH CONSTRUCTION

As shown in Figure 3, the first step to construct a social relevance attribute network graph is to extract the social network graph and output a social attribute network graph, which is combined with the user-attribute relationships. We establish the social relevance attribute network by aggregating the attribute relevance properties into the social attribute network.

Specifically, the social relevance attribute network (SRAN) graph is defined as $\mathbb{G} = (V, E, W)$, where V denotes nodes in the graph, E denotes edges and W denotes edge weights. Apart from social (user) nodes,[2] we define every attribute value as an *attribute node*. In an SRAN graph $\mathbb{G}$, the vertex set V includes user node set V_u and attribute node set V_a, and $V = V_u \cup V_a$. For each attribute value of a user, we create a edge between the user node and the attribute node if the user has the attribute value. Differing from the traditional SAN graph, we create a new type of edges between two attribute values, which represent their relevance values. In graph $\mathbb{G}$, the edge set E includes user links E_{uu}, user-attribute links E_{ua} and attribute links E_{aa}, where $E = E_{uu} \cup E_{ua} \cup E_{aa}$.

Before defining the weights, we define a capacity C_u of a node u to represent its total proximity with other nodes. Then we divide a user node's capacity C_u into two parts, which can be represented as $C_u = \lambda_1 C_{uu} + (1 - \lambda_1)C_{ua}$, where C_{uu} denotes the proximity to all neighboring user nodes of u and C_{ua} denotes the proximity to all neighboring attribute nodes of u. And λ is a parameter that controls the trade-off between friend links and attribute values. Similarly, the capacity of an attribute node can be defined as $C_a = \lambda_2 C_{aa} + (1 - \lambda_2)C_{au}$. Therefore, the weights of all edges in $\mathbb{G}$ are defined as follows. The weight from a user node u to another user node v denoted by $w(u, v)$ is defined as follows.

$$w(u,v) = \begin{cases} \dfrac{\lambda_1 \cdot C_{uv}}{\sum\limits_{(u,v)\in E_{uu}} C_{uv}} & \text{if}(u,v)\in E_{uu},\ |N_u(u)|>0 \ \text{and}\ |N_a(u)|>0; \\ \dfrac{C_{uv}}{\sum\limits_{(u,v)\in E_{uu}} C_{uv}} & \text{if}(u,v)\in E_{uu},\ |N_u(u)|>0 \ \text{and}\ |N_a(u)|=0; \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

where $N_u(u)$ denotes the user neighbors of u, $N_a(u)$ denotes the attribute neighbors of u. C_{uv} denotes the proximity of user node u to v, and $\frac{C_{uv}}{\sum_{(u,v)\in E_{uu}} C_{uv}}$ denotes the probability of v get close to u among all its neighbors. If C_{uv} is large, it means node v is more close to node u than other neighbors, so they may have more attribute values in common.

The weight from a user node u to an attribute node a denoted by $w(u, a)$ is defined as follows.

$$w(u,a) = \begin{cases} \dfrac{(1-\lambda_1) \cdot C_{ua}}{\sum\limits_{(u,a)\in E_{ua}} C_{ua}} & \text{if}(u,a)\in E_{ua},\ |N_a(u)|>0 \ \text{and}\ |N_u(u)|>0; \\ \dfrac{C_{ua}}{\sum\limits_{(u,a)\in E_{ua}} C_{ua}} & \text{if}(u,a)\in E_{ua},\ |N_a(u)|>0 \ \text{and}\ |N_u(u)|=0; \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

where C_{ua} denotes the proximity of user node u to attribute node a, and $\frac{C_{ua}}{\sum_{(u,a)\in E_{ua}} C_{ua}}$ denotes the preference that user u has to attribute a among all the attribute values of u. Large C_{ua} means u obtain more confidence to have value a. For example, in hometown inference, the cities where a person has been are listed in attribute values. Intuitively a long-time-stay city contributes more to the person than a short-time-stay city.

Considering the attribute relevance, we endow different weights to edges between attribute values. The weight

[2] In this paper, we use "social node" and "user node" exchangeably.

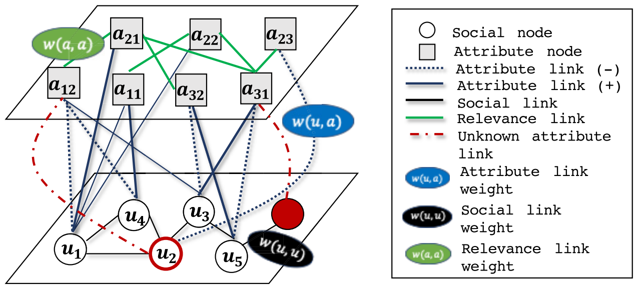


FIGURE 4. Social relevance attribute network graph.

between two attribute nodes, a and a', is defined as follows.

$$w(a,a')=\begin{cases}\dfrac{\lambda_2\cdot C_{aa'}}{\sum\limits_{(a,a')\in E_{aa}} C_{aa'}} \\ \qquad \text{if}(a,a')\in E_{aa}, |N_a(a)|>0 \ \text{ and } \ |N_u(a)|>0; \\ \dfrac{C_{aa'}}{\sum\limits_{(a,a')\in E_{aa}} C_{aa'}} \\ \qquad \text{if}(a,a')\in E_{aa}, |N_a(a)|>0 \ \text{ and } \ |N_u(a)|=0; \\ 0 \quad \text{otherwise.}\end{cases} \tag{6}$$

where $N_a(a)$ denotes the attribute neighbors of a, $N_u(a)$ denotes the user neighbors of a. And $C_{aa'}$ denotes the proximity of attribute node a to a', and $\frac{C_{aa'}}{\sum_{(a,a')\in E_{aa}} C_{aa'}}$ denotes relevance degree of a to a' among all its neighbors. Large $C_{aa'}$ means a is more relevant to a' than other attribute neighbors, which implies the high probability of co-occurrence in a user attribute lists.

The weight from a attribute node a to a user node u denoted by $w(a, u)$ is defined as follows.

$$w(a,u)=\begin{cases}\dfrac{(1-\lambda_2)\cdot C_{au}}{\sum\limits_{(a,u)\in E_{ua}} C_{au}} \\ \qquad \text{if}(a,u)\in E_{ua}, |N_u(a)|>0 \ \text{ and } \ |N_a(a)|>0; \\ \dfrac{C_{ua}}{\sum\limits_{(a,u)\in E_{ua}} C_{au}} \\ \qquad \text{if}(a,u)\in E_{ua}, |N_u(a)|>0 \ \text{ and } \ |N_a(a)|=0; \\ 0 \quad \text{otherwise.}\end{cases} \tag{7}$$

where C_{au} denotes the proximity of attribute node a to user node u, and $\frac{C_{au}}{\sum_{(a,u)\in E_{ua}} C_{au}}$ denotes the importance that attribute value a contributes to different user node u among users who have a. Large C_{au} means users who are similar to u are more possible to have attribute a than those who are similar to other user nodes neighboring to a.

Therefore, we can build a new social attribute-relevance network (SRAN) model, shown in Figure 4, using the nodes, edges and weights we discussed above. We introduce our attribute inference method based on the model in the following part.

B. ATTRIBUTE INFERENCE ALGORITHM BASED ON SRAN GRAPH VIA RANDOM WALK WITH RESTART

After we construct the new social relevance-attribute network(SRAN) graph, then the problem is to find the "closest" attribute node to some targeted user node through the graph. Intuitively, we propose a random walk with restart based method on SRAN to perform attribute inference. As the stationary probability of random walk is considered as the proximity between two nodes. Starting from the vary targeted user node and it converges to the stationary probability that represent the proximity form the targeted node to other nodes. Therefore, we can find the "closest" attribute node to the targeted user node, which considered as the user's predicted attribute.

So First we endow capacities for nodes in graph $\mathbb{G}$. The capacities propagate through the whole graph and converge to a stationary situation, where we can find the "closest" attribute node to the target node.

The random walk with restart process is defined by Equation (8) as follows.

$$P_t = (1-\alpha) * T * P_{t-1} + \alpha * PV_0 \tag{8}$$

where P_t denotes the node capacity vector at the ith iteration, T is the transition matrix for graph $\mathbb{G}$, PV_0 is the personalized vector toward the targeted user, and α is the restart probability.

For the initial capacity vector P_0 for all nodes in $\mathbb{G}$, we can randomly initialize it, and then normalize it to the interval [0,1] for later calculation.

In the SRAN graph $\mathbb{G}$, to calculate the proximity for a special node, we need to define the personalized vector. For a targeted user node u, the personalized vector PV_0 is defined in Equation (9) as follows.

$$PV_0(i) = \begin{cases}1 & i=u; \\ 0 & \text{otherwise.}\end{cases} \tag{9}$$

In section II, we have defined edge weights for various edges as Equation (4)-(7). And we use these edge weights to calculate the transition matrix T. Each element of transition matrix is computed as Equation (10)

$$T_{ij} = \begin{cases} w(u,v) & \text{if} \quad (i,j)\in E_{uu}; \\ w(u,a) & \text{if} \quad (i,j)\in E_{ua}, i\in V_u \text{and } j\in V_a; \\ w(a,a') & \text{if} \quad (i,j)\in E_{aa}; \\ w(a,u) & \text{if} \quad (i,j)\in E_{ua}, i\in V_a \text{and } j\in V_u; \\ 0 & \text{otherwise.}\end{cases} \tag{10}$$

For simplicity, in computing weights of graph $\mathbb{G}$, we take a node's capacity into two equal part for separately neighboring user nodes and neighboring attribute nodes, where $\lambda_1 = \lambda_2 = 0.5$. And we take equal division for a node's capacity to another except for attribute node to attribute node. For example, a user node u's capacity $C_u = 0.5C_{uu} + 0.5C_{ua}$, and a neighboring user node v obtain $C_{uv} = C_{uu}/|N_u(u)|$ capacity from u.

Therefore, the Equation (4)(5)(7) can be simplified as follows.

$$w(u,v)=\begin{cases}\dfrac{1}{2\times|N_u(u)|} & \text{if}(u,v)\in E_{uu},\ |N_u(u)|>0 \ \text{ and } \ |N_a(u)|>0;\\ \dfrac{1}{|N_u(u)|} & \text{if}(u,v)\in E_{uu},\ |N_u(u)|>0 \ \text{ and } \ |N_a(u)|=0;\\ 0 & \text{otherwise.}\end{cases} \tag{11}$$

$$w(u,a)=\begin{cases}\dfrac{1}{2\times|N_a(u)|} & \text{if}(u,a)\in E_{ua},\ |N_a(u)|>0 \ \text{ and } \ |N_u(u)|>0;\\ \dfrac{1}{|N_a(u)|} & \text{if}(u,a)\in E_{ua},\ |N_a(u)|>0 \ \text{ and } \ |N_u(u)|=0;\\ 0 & \text{otherwise.}\end{cases} \tag{12}$$

$$w(a,u)=\begin{cases}\dfrac{1}{2\times|N_u(a)|} & \text{if}(a,u)\in E_{ua},\ |N_u(a)|>0 \ \text{ and } \ |N_a(a)|>0;\\ \dfrac{1}{|N_u(a)|} & \text{if}(a,u)\in E_{ua},\ |N_u(a)|>0 \ \text{ and } \ |N_a(a)|=0;\\ 0 & \text{otherwise.}\end{cases} \tag{13}$$

As for $w(a,a')$, in our method we need consider the relevance between attributes. Therefore we cannot use equal division to compute it. As we have mentioned in the last section, we use **Kulc** measure to quantify the relevance of attributes. In the measurement, we use the priori knowledge (public information) about the attribute values to compute **Kulc** measure and then obtain $w(a,a')$. That is to say we first use public attribute values of users to obtain population distribution and then we use Equation(3) to compute relevance degree between those attribute values, and then use relevance degree to define the weights between attribute nodes. In this paper, we compute the number of users who are known to have both attribute value a and a' denoted by $n_p(aa')$, and compute the number of users who are known to have attribute value a denoted by $n_p(a)$,and compute the number of users who are known to have attribute value a' denoted by $n_p(a)$. According to Equation(3) the relevance degree of a to a' among all its neighbors can be denoted as

$$\begin{aligned} Kulc(a,a') &= \frac{1}{2}(P(a|a')+P(a'|a))\\ &= \frac{1}{2}(confidence(a'\Rightarrow a)+confidence(a\Rightarrow a'))\\ &= \frac{1}{2}\left(\frac{n_p(aa')}{n_p(a')}+\frac{n_p(aa')}{n_p(a)}\right). \end{aligned} \tag{14}$$

So $w(a,a')$ can be denoted as follows.

$$w(a,a')=\begin{cases}\dfrac{Kulc(a,a')}{2} & \text{if}(a,a')\in E_{aa},\ |N_a(a)|>0 \ \text{ and } \ |N_u(a)|>0;\\ Kulc(a,a') & \text{if}(a,a')\in E_{aa},\ |N_a(a)|>0 \ \text{ and } \ |N_u(a)|=0;\\ 0 \text{ otherwise.}\end{cases} \tag{15}$$

Therefore, we can use the simplified equations above to compute the transition matrix T.

Algorithm 1 shows our relevance attribute inference method(ReAI) using random walk with restart. After we construct the new SRAN graph, we choose a targeted user node u for attribute inference. We first initialize the personalized vector PV_0, the transition matrix T and the node capacity vector P_0. Then we iteratively update the node weight vector P using Equation (8). It dose not stop until P converges. Finally, we choose the attribute value with the max stationary probability to be the predicted attribute value for the target node u.

Algorithm 1 Relevance Attribute Inference (ReAI)

Input: SRAN graph $\mathbb{G}$, Targeted user u, Parameter α
Output: The best attribute value for u
1: Initialize personalized vector PV_0 through Equation (9)
2: Initialize transition matrix T through Equation (10)
3: Randomly initialize and normalize node capacity vector P_0
4: $t = 1$
5: **repeat**
6: $P_t = (1-\alpha) * T * P_{t-1} + \alpha * PV_0$
   $t = t + 1$
7: **until** P_t converge
8: **return** The corresponding attribute value that gained max stationary probability in final P_t

Although in this paper, we only consider friendship and public attribute information to infer private attributes, actually we can also add user behaviors to form behavior nodes and construct social behavior relevance-attribute (SBRA) network and perform ReAI algorithm on SBRA graph. For simplicity, we only introduce in detail our method on SRAN graph and conduct experiments on SRAN graph.

C. TIME COMPLEXITY

We assume that the new SRAN graph has M edges and has N user nodes and the ReAI algorithm takes t iterations. The major process of our method is to compute the transition matrix T and do iterations to update P. The transition matrix only needs to be computed once. The transition matrix is a sparse matrix with M non-zero entries. Before computing T, we need to compute the number of users who are known to have attribute value a and attribute value a' together and

TABLE 1. Complexity comparison of attribute inference methods.

Algorithms	Time Complexity
Lindamood et al. [12]	$O(N)$
TAI [4]	$O(iter\|E\|)$
VIAL [6]	$O(d \cdot m)$
ReAI	$O(tM + N)$

m is the number of links in the SBA network.
d is the number of iterations.
iter is the iteration numbers.
$|E|$ is the edge count of the augmented graph G.
N is the number of public traits a person has.

TABLE 2. Categories of four attributes.

Attribute	The number of distinct values
gender	2
birthday	40
education type	3
work employer	145
work position	63
work location	49

separately. To compute the numbers, we need to traverse all the user nodes. So the time complexity of it is $O(N)$. And then to compute T, for each node in G we need to go through its neighbors. Therefore, the time complexity of computing T is $O(M)$. Using T in each iteration to compute the node weight vector P for one targeted user, the time complexity is $O(M)$. Therefore, the time complexity of t iterations is $O(tM)$. To sum up, the total time complexity of ReAI algorithm for one targeted user is $O(M + N + tM) = O(tM + N)$. We compared the time complexity of several algorithms, as you can see in Table 1, our work is preety efficient.

IV. EVALUATION

We conducted a series of experiments to evaluate the ReAI method proposed in this paper. We compare our method to the attribute inference (TAI) method in [4], which does not take attribute relevance into consideration. We evaluate and compare the two methods and use *Information Gain Ratio* to analyze the efficiency of the ReAI method.

A. DATA PREPARATION

In our experiments, we use a public dataset of Facebook generated by Mcauley and Leskovec [13]. It contains users, friend relationship, and various attributes (e.g., gender, birthday, education type, etc.), which is suitable for our experiments. The dataset consists of 4039 user nodes and 88234 social links. In the following experiments, we choose six types of attributes, namely, gender, birthday, education type, employer, work position and work location, to perform attribute inference. We summarize the distinct values of each attribute in Table 2 as follows.

B. EXPERIMENT SETUP

To evaluate the effectiveness of our ReAI method, we conduct seven groups of experiments. In the following experiments, we select two attribute types as one group and evaluate the effect of their relevance on attribute prediction results. We choose seven groups shown in Table 3 as follows.

TABLE 3. Classified attribute groups of experiments.

Attribute groups	Dataset	Parameter
gender–education type	Facebook	$\alpha = 0.15$
gender–work employer		
gender–work position		
education type–work employer		
education type–work location		
education type– work position		
birthday–education type		

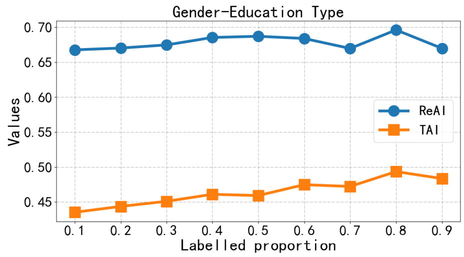


FIGURE 5. Attribute inference: gender and education type.

To evaluate the effectiveness, we use **precision** as the evaluation metrics. In our experiments, precision is defined as the number of correctly inferred user nodes proportioned to the number of all predicted nodes that have attribute values.

We select part of user nodes from whole Facebook user nodes as our test (targeted) user nodes. For test user nodes, we remove their attribute links and use them as ground truth. Other user nodes with known attributes are used as training user nodes for inference propagation, as well as the prior knowledge to compute weights between attribute edges in Equation (7).

In our experiments, we analyze the effect of the attribute relevance to the precision with regard to the different proportion of training user nodes to all user nodes. The parameter in our experiment is α in Equation (8). We set $\alpha = 0.15$ empirically. We set the number of iterations $t = 50$.

C. COMPARISON RESULTS

We conduct several experiments on both methods. The comparison results are shown as follows. The horizontal axis denotes the proportion of training user nodes to all user nodes, while the vertical axis denotes the precision of attribute inference. Under each proportion, we test 10 times to compute the average accuracy, which is shown as follows.

Observation: Through the evaluation using seven groups of attributes, we can see that the ReAI method obtains higher precision than the TAI method. It means our method performs better attribute inference. As in Figure 5, Figure 6 and Figure 7, we evaluate the precision between *gender* and *education type*, *work employer* or *work position*. The ReAI method obtains about 25 percent more precision than the TAI method when labeled proportion is 0.1 over *gender-education type*.

As in Figure 8, Figure 9 and Figure 10, we evaluate the precision between *education type* and *work employer*, *work location* and *work position*. The ReAI method obtains about 18 percent more precision than the TAI method when labeled

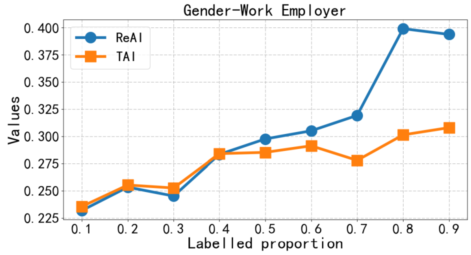


FIGURE 6. Attribute inference: gender and work employer.

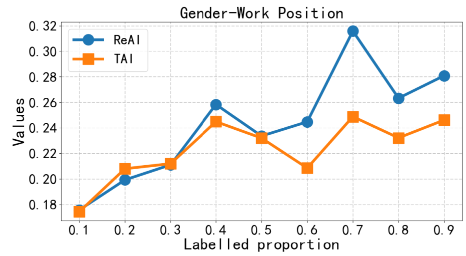


FIGURE 7. Attribute inference: gender and work position.

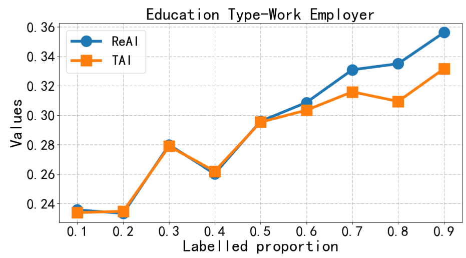


FIGURE 8. Attribute inference: education type and work employer.

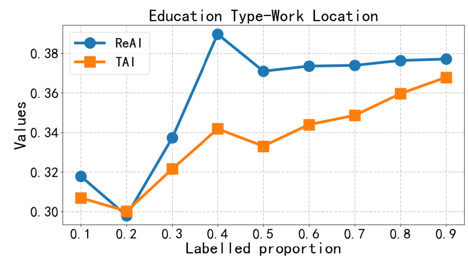


FIGURE 9. Attribute inference: education type and work location.

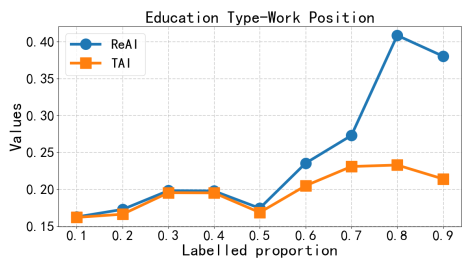


FIGURE 10. Attribute inference: education type and work position.

proportion is 0.8 over *education type-work position*, and obtains about 5 percent more precision when labeled proportion is 0.4 over *education type-work location*, and 2.5 percent more precision over *education type-work employer*.

And in Figure 11, we evaluate the performance over birthday and education type. We can see that the ReAI method performs better than the TAI method, and the ReAI method obtains about 20 percent more precision than the TAI method when labeled proportion is 0.2 over *birthday-education type*.

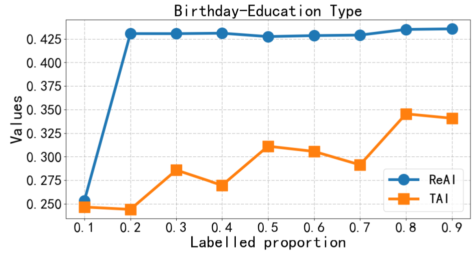


FIGURE 11. Attribute inference: birthday and education type.

Conclusion: Our ReAI method takes attribute relevance into consideration, and it performs better than the TAI method. However, we can see that the experiments in different attribute pairs have different improvements. The bigger relevance attributes have, the more improvements ReAI method obtains. when there is no relevance or much lower relevance between attributes, the result of ReAI method and TAI method would be similar. It means our ReAI method on different types of attributes with higher degree of relevance would perform much higher precision of attribute inference than TAI method.

D. ATTRIBUTE RELEVANCE ANALYSIS

In Decision Tree, *Information Gain Ratio* can be used to select effective features, especially for discrete features. In another aspect, *Information Gain Ratio* can be also used for feature relevance evaluation and we explain it as follows. For a dataset S and two features, A and B, to classify S, $Entropy(A)$ denotes the information entropy of itemsets classified by feature A, defined as Equation(16)

$$Entropy(A) = -\sum_{i=1}^{m} p(i)\log(p(i)). \tag{16}$$

where m denotes the number of class that dataset S can be classified by A, and $p(i)$ denotes the probability of items that belong to class i in feature A.

After being classified by feature A, the itemsets are classified by feature B then. And $Gain(A\_B)$ denotes the information gain after the operations, defined as Equation(17). The information gain for A and B is symmetric, where $Gain(A\_B) = Gain(B\_A)$.

$$Gain(A\_B) = Entropy(A) - \sum_{i=1}^{n} \frac{|B_i|}{|S|} Entropy(A\_B_i). \tag{17}$$

where n denotes the number of classes that itemsets can be classified by B, B_i denotes the set of the items that belong to class i, $|B_i|$ denotes the number of items of class i, $|S|$ denotes the number of total samples, and $Entropy(A\_B_i)$ denotes the information entropy of itemsets classified by A in itemset B_i,

TABLE 4. GainRatio of attribute pairs of Facebook dataset.

Attribute Pair	GainRatio
gender-education type	0.0008
gender-work employer	0.1153
gender-work position	0.0697
education type-work employer	0.0799
education type-work position	0.0497
education type-work location	0.0503
birthday-education type	0.0158
birthday-work employer	0.4726
birthday-work position	0.4168
birthday-work location	0.3291

defined as Equation(18).

$$Entropy(A\_B_i) = -\sum_{j=1}^{m} p_i(j)\log(p_i(j)) \quad (18)$$

where $p_i(j)$ denotes the probability of items in B_i that belong to class j in feature A.

We defined *Information Gain Ratio* as Equation(19), which can be used to evaluate the relevance of A and B.

$$GainRatio(AB) = \frac{Gain(A\_B)}{\sqrt{Entropy(A) * Entropy(B)}}. \quad (19)$$

If the itemsets classified by B is more similar to that classified by A, $Entropy(A\_B_i)$ is closer to 0 and the $GainRatio(AB)$ is larger. That is to say, the bigger $GainRatio(AB)$ is, more similar the feature A is to the feature B and more bigger the probability using A to infer B is.

We use the *Information Gain Ratio* to calculate the attribute relevance shown as Table 4. **GainRatio** represent the similarity that two attribute features can classify items. The bigger **GainRatio** is, the more similarity they have in classification and the easier they can be replaced by each other. Combining the **GainRatio** values and the evaluation results in the previous subsection. If two attribute features have less **GainRatio**, our ReAI method obtains higher improvements than the TAI method.

Therefore, the bigger relevance attributes have, the more improvements ReAI method obtains. When there is no relevance or much lower relevance between attributes, the result of the ReAI method and the TAI method would be similar. So it means our ReAI method on different types of attributes with a higher degree of relevance would perform much higher precision of attribute inference than the TAI method.

V. RELATED WORK

There are several research efforts focused on the attribute inference problem. We can simply classify them into four categories. We give a brief introduction to these methods and then describe the methods related to ours in detail.

a: COMMUNITY-BASED ATTRIBUTE INFERENCE METHODS

According to homogeneity, similar users seem to get together and have more relationships. So, they seem to form communities as high density of their links. Users in one community have a higher probability to have the same attribute value, which produces community-based attribute inference methods. Misolve *et al.* [14] proposed an attribute inference method based on community detection. They infer private attributes of users according to the public attributes of users in the same community. They conducted experiments on a Facebook dataset to infer departments of users etc., which obtain a high accuracy. Mo and King [17] proposed a community-based graph semi-supervised learning to predict vertices labels in online social networks. Traud *et al.* [15] compared community structure to partitions based on given categories of Facebook to examine the effect of common attributes at the dyad level.

b: MACHINE LEARNING BASED ATTRIBUTE INFERENCE METHODS

Classifiers can be used to classify users into different values of attribute [18]. Users with public attribute values are used to train the classifier and then make a prediction on users with private attribute values. Besides, clustering algorithms are also used to cluster users with similar features into the same group. Neville and Jensen [19] proposed iterative classification algorithm (ICA) framework and naive Bayes classifier to infer attributes. Top nodes with the highest confidence are added to the next Bayes model training. It iteratively infers the private attributes of users. He *et al.* [20] used Bayesian networks to model social networks, and discussed the effect on attribute inference by prior probability, influence strength and society openness. Bhagat *et al.* [3] utilized K-Nearest Neighbor algorithm based on ICA framework to infer attributes on the LiveJournal dataset. Lindamood *et al.* [12] improved the Naive Bayes algorithm to predict attributes. Mo *et al.* [21] proposed a semi-supervised learning framework to infer attributes. Thomas *et al.* [16] used multi-label classification methods to infer attributes using friendships and wall posts, and they proposed multi-party privacy to defend against attribute inference.

c: RANDOM WALK BASED ATTRIBUTE INFERENCE METHODS

Random walk makes labels propagate through the network and stop at the most "close" node. The vote distribution-based methods are similar to it. They both conduct label propagation through the graph using a transition matrix and finally choose the attribute value with the most proximity. Bhagat *et al.* [3] propose a local iterative algorithm to infer attribute by choosing the value occurring the most frequently in local neighbors of a user node, which can be called majority voting by local neighbors. Macskassy and Provost [22] propose a relational neighbor model and propose two algorithms, iterative relational neighbor and probabilistic relational neighbor to perform attribute inference, which shows good performance on nine different datasets. Mo *et al.* [21] propose a graph-based model for attribute inference using friendship, group membership and network relationship for similarity computed as the transition matrix to perform label propagation. Yin *et al.* [4] use random walk with restart to

TABLE 5. Comparison of attribute inference methods.

Approach	Social Features					Category			Performance	
	Social Link	Attribute Link	Behavior Link	Community	Attribute Relevance	I	II	III	Scalability	Practical
Mislove et al. [14]	√	√		√		√			○	●
Traud et al. [15]	√	√		√		√			●	○
Lindamood et al. [12]	√	√					√		●	○
Thomas et al. [16]	√	√					√		○	○
Mo et al. [17]	√	√		√				√	○	●
TAI [4]	√	√						√	○	○
VIAL [6]	√	√	√					√	○	○
ReAI (Ours)	√	√			√			√	○	○

○ represents that the algorithm performs well; ● represents that the algorithm works conditionally.
Category I: community-based attribute inference methods.
Category II: machine learning attribute inference methods.
Category III: random walk attribute inference methods.

attribute ranking based on the social-attribute network (SAN). They model attributes as nodes and build links between user nodes and attribute nodes. But they don't consider attribute relevance in the inference process. And in this paper, we treat it as traditional attribute inference method for comparison. Gong and Liu [6] propose an attribute inference attack based on vote distribution via social friends and user behaviors. They construct a social-behavior-attribute (SBA) network regarding attributes and behaviors as nodes and building links between user nodes with them. They compute a vote dividing matrix for all edges and use random walk with restart to divide vote capacity from the targeted user node to different attribute nodes. And then they aggregate the voting capacity for each attribute node and choose the top capacity attribute nodes for inferred attributes. It performs better than methods using only friends or behaviors. Rossi *et al.* [23] propose a relational similarity machine framework for attribute inference. They compute the similarity of the targeted node with labeled and unlabeled neighbors and non-neighbors. And they use the similarity score between two user nodes to iteratively update the probability vector for attribute types.

d: OTHER METHODS

Zheleva and Getoor [24] use friendship and group membership to infer attributes. Zamal *et al.* [25] utilize users' and their neighbors' tweets to perform attribute inference. Gupta *et al.* [26] inferred users' interests on Facebook pages via sentiment-oriented mining. Dong *et al.* [27] use mobile communication to infer gender and age considering features of them and connections between the attribute values of a node. Gong *et al.* [28] extend the traditional link prediction problem to SAN graph for attribute inference. Zhong *et al.* [29] demonstrate the list of locations where users checked in can be used to infer attributes. Qian *et al.* [30] use a knowledge graph to represent arbitrary prior knowledge of attackers and compute node structural similarity and attribute similarity to private attribute inference based the knowledge graph. Jia *et al.* [31] used a pairwise Markov Random Field to model a social network and used improved Loopy Belief Propagation to compute the posterior probability to infer attributes. We compare our work with several classic methods mentioned above in Table 5.

VI. CONCLUSION

In this paper, we propose a relevance-attribute inference (ReAI) method using random walk with restart. We first analyze the relevance between attributes and use **Kulc measure** to quantify them. We improve the SAN graph by adding attribute links between attribute nodes constructing the new SRAN graph and use prior knowledge to compute the edge weights of attribute links to represent the degree of relevance between different attributes. And then we perform the random walks with restart process based on SRAN graph to find the "closest" attribute node for the given targeted node. Finally, we conduct various groups of experiments to evaluate the effectiveness of our ReAI method using real-world Facebook dataset and to compare our ReAI method with the traditional attribute inference method. We use **Kulc measure** and **GainRatio** evaluate our improvements. According to our experiments, our approach achieves better performance when attributes have relevances. In addition, our method can be used to enhance the existing solutions.

ACKNOWLEDGMENT

(Wenqian Tian and Yitong Yang contributed equally to this work.)

REFERENCES

[1] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 4, pp. 577–590, Jul./Aug. 2016.

[2] Y. Huang, Z. Cai, and A. G. Bourgeois, "Search locations safely and accurately: A location privacy protection algorithm with accurate service," *J. Netw. Comput. Appl.*, vol. 103, pp. 146–156, Feb. 2019.

[3] S. Bhagat, G. Cormode, and I. Rozenbaum, "Applying link-based classification to label blogs," in *Proc. Int. Workshop Social Netw. Mining Anal.* San Jose, CA, USA: Springer, Aug. 2007, pp. 97–117.

[4] Z. Yin, M. Gupta, T. Weninger, and J. Han, "A unified framework for link recommendation using random walks," in *Proc. IEEE Int. Conf. Adv. Social Netw. Anal. Mining*. Odense, Denmark, Aug. 2010, pp. 152–159.

[5] X. Zheng, Z. Cai, J. Yu, C. Wang, and Y. Li, "Follow but no track: Privacy preserved profile publishing in cyber-physical social systems," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1868–1878, Dec. 2017.

[6] N. Z. Gong and B. Liu, "You are who you know and how you behave: Attribute inference attacks via users' social friends and behaviors," in *Proc. 25th USENIX Secur. Symp. (USENIX Secur.)*, Austin, TX, USA, Aug. 2016, pp. 979–995.

[7] X. Zheng, Z. Cai, J. Li, and H. Gao, "Location-privacy-aware review publication mechanism for local business service systems," in *Proc. 36th Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, May 2017, pp. 1–9.
[8] X. Zheng, Z. Cai, G. Luo, L. Tian, and X. Bai, "Privacy-preserved community discovery in online social network," *Future Gener. Comput. Syst.*, vol. 93, no. 1, pp. 1002–1009, 2019.
[9] InfoQ. (May 11, 2017). *InfoQ*. [Online]. Available: https://www.infoq.cn
[10] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. Amsterdam, The Netherlands: Elsevier, 2011.
[11] L. Takac and M. Zabovsky, "Data analysis in public social networks," in *Proc. Int. Sci. Conf. Int. Workshop Present Day Trends Innov.* Lomza, Poland: Dubnica Technological Institute, May 2012, pp. 1–6.
[12] J. Lindamood, R. Heatherly, M. Kantarcioglu, and B. M. Thuraisingham, "Inferring private information using social network data," in *Proc. ACM 18th Int. Conf. World Wide Web*, Madrid, Spain, Apr. 2008, pp. 1145–1146.
[13] J. J. Mcauley and J. Leskovec, "Learning to discover social circles in ego networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 539–547.
[14] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel, "You are who you know: Inferring user profiles in online social networks," in *Proc. 3rd ACM Int. Conf. Web Search Data Mining*, New York, NY, USA, Feb. 2010, pp. 251–260.
[15] A. L. Traud, P. J. Mucha, and M. A. Porter, "Social structure of Facebook networks," *Phys. A, Stat. Mech. Appl.*, vol. 391, no. 16, pp. 4165–4180, Aug. 2012.
[16] K. Thomas, C. Grier, and D. M. Nicol, "unFriendly: Multi-party privacy risks in social networks," in *Proc. Int. Symp. Privacy Enhancing Technol. Symp.* Berlin, Germany: Springer, Jul. 2010, pp. 236–252.
[17] M. Mo and I. King, "Exploit of online social networks with community-based graph semi-supervised learning," in *Proc. Int. Conf. Neural Inf. Process.* Sydney, NSW, Australia: Springer, Nov. 2010, pp. 669–678.
[18] Z. He, Z. Cai, and J. Yu, "Latent-data privacy preserving with customized data utility for social network data," *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 665–673, Jan. 2018.
[19] J. Neville and D. Jensen, "Iterative classification in relational data," in *Proc. AAAI Workshop Learn. Stat. Models Relational Data*, Austin, TX, USA, Aug. 2000, pp. 13–20.
[20] J. He, W. W. Chu, and Z. Liu, "Inferring privacy information from social networks," in *Proc. IEEE Int. Conf. Intell. Secur. Inform.*, San Diego, CA, USA, May 2006, pp. 154–165.
[21] M. Mo, D. Wang, B. Li, D. Hong, and I. King, "Exploit of online social networks with semi-supervised learning," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, Barcelona, Spain, Jul. 2010, pp. 1–8.
[22] S. A. Macskassy and F. Provost, "A simple relational classifier," in *Proc. ACM 2nd Int. Workshop Multi-Relational Data Mining KDD*, Washington, DC, USA, Aug. 2003, pp. 64–76.
[23] R. A. Rossi, R. Zhou, and N. K. Ahmed, "Relational similarity machines," Aug. 2016, *arXiv:1608.00876*. [Online]. Available: https://arxiv.org/abs/1608.00876
[24] E. Zheleva and L. Getoor, "To join or not to join: The illusion of privacy in social networks with mixed public and private user profiles," in *Proc. ACM 18th Int. Conf. World Wide Web*, Madrid, Spain, Apr. 2009, pp. 531–540.
[25] F. Al Zamal, W. Liu, and D. Ruths, "Homophily and latent attribute inference: Inferring latent attributes of Twitter users from neighbors," in *Proc. ICWSM*, vol. 270, 2012, pp. 1–4.
[26] P. Gupta, S. Gottipati, J. Jiang, and D. Gao, "Your love is public now: Questioning the use of personal information in authentication," in *Proc. 8th ACM SIGSAC Symp. Inf., Comput. Commun. Secur.*, Hangzhou, China, May 2013, pp. 49–60.
[27] Y. Dong, Y. Yang, J. Tang, Y. Yang, and N. V. Chawla, "Inferring user demographics and social strategies in mobile social networks," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Aug. 2014, pp. 15–24.
[28] N. Z. Gong, A. Talwalkar, L. Mackey, L. Huang, E. C. R. Shin, E. Stefanov, E. Shi, and D. Song, "Joint link prediction and attribute inference using a social-attribute network," *Trans. Intell. Syst. Technol.*, vol. 5, no. 2, p. 27, 2014.
[29] Y. Zhong, N. J. Yuan, W. Zhong, F. Zhang, and X. Xie, "You are where you go: Inferring demographic attributes from location check-ins," in *Proc. 8th ACM Int. Conf. Web Search Data Mining*, Shanghai, China, Feb. 2015, pp. 295–304.
[30] J. Qian, X.-Y. Li, C. Zhang, and L. Chen, "De-anonymizing social networks and inferring private attributes using knowledge graphs," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun.*, San Francisco, CA, USA, Apr. 2016, pp. 1–9.
[31] J. Jia, B. Wang, L. Zhang, and N. Z. Gong, "AttriInfer: Inferring user attributes in online social networks using Markov random fields," in *Proc. ACM 26th Int. Conf. World Wide Web*. Perth, WA, Australia, Apr. 2017, pp. 1561–1569.

JIAN MAO received the B.S. and Ph.D. degrees from Xidian University, Shanxi, China. She is currently an Associate Professor with the School of Cyber Science and Technology, Beihang University, Beijing, China. Her research interests include social network security, web security, cloud security, and mobile security.

WENQIAN TIAN received the B.S. and M.S. degrees in electronic and information engineering from Beihang University, Beijing, China. Her research interests include web security, mobile security, and privacy analysis.

YITONG YANG received the B.S. degree in electronic and information engineering from Beihang University, Beijing, China, in 2018, where she is currently pursuing the master's degree with the School of Cyber Science of Technology. Her research interests include learning-based security analysis and social network privacy preserving.

JIANWEI LIU received the B.S. and M.S. degrees in electronic and information from Shandong University, Shandong, China, in 1985 and 1988, respectively, and the Ph.D. degree in communication and electronic system from Xidian University, Shanxi, China, in 1998. He is currently a Professor with the School of Cyber Science and Technology, Beihang University, Beijing, China. His current research interests include wireless communication networks, cryptography, and network and information security.