Hindawi
Security and Communication Networks
Volume 2019, Article ID 8171263, 23 pages
https://doi.org/10.1155/2019/8171263

Research Article

SNI: Supervised Anonymization Technique to Publish Social Networks Having Multiple Sensitive Labels

A. Karimi Rizi,[1,2] M. Naderi Dehkordi,[1,2] and N. Nemat bakhsh[3]

[1]*Faculty of Computer Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran*
[2]*Big Data Research Center, Najafabad Branch, Islamic Azad University, Najafabad, Iran*
[3]*Shahid Ashrafi Esfahani University, Esfahan, Iran*

Correspondence should be addressed to M. Naderi Dehkordi; m_naderi_d@yahoo.com

Received 12 July 2019; Revised 29 September 2019; Accepted 16 October 2019; Published 6 November 2019

Academic Editor: Leandros Maglaras

In social networks, preserving privacy and preserving correlation among sensitive labels are a matter of trade-off. This paper presents a supervised anonymization technique, SNI (social network immunization), to publish social networks having multiple sensitive labels with correlation. SNI publishes all sensitive labels without distorting them. It publishes sensitive labels along with innovative labels named "partial sensitive labels" in an immune graph and multiple supplementary trees. These graph and trees, by itself or with the combination of other objects, supply correlation among sensitive labels for membership analysis. We present a framework along with an algorithm for extracting the immune graph and supplementary trees. These graph and trees minimize the membership error rate for membership analysis. The practical evaluation of the cancer code label of individuals also indicates the effectiveness of the SNI method.

1. Introduction

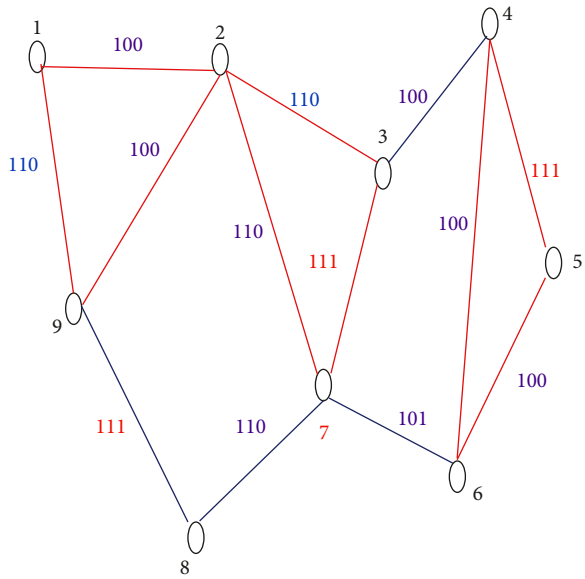
Now, the most important need for societal network analysts is access to raw data of social networks, and the most important need for individuals is privacy preservation when raw data are released. Publishing social networks for research purposes raises serious concerns for individual privacy.

The important issue in privacy-preserving data publishing (PPDP) is maintaining privacy while maintaining the utility of data. Data privacy and data utility are always a matter of trade-off. In the PPDP area, it is assumed that a data recipient may be an adversary. In this area, various methodologies are presented for publishing data in an adversary environment. Anonymization is the most common methodology in PPDP [1, 2].

Anonymization usually provides an opportunity for the data holder to anonymize the sensitive information of data owners before publishing raw data of social networks. When a social network site collects information of individuals and a research institution receives the information for analysis from that site, individuals are considered data owners, the site the data holder (publisher), and the research institution the data receiver.

In the age of social network analysis, most social network data are explicitly available [3]. Now, publishing social networks which have multiple sensitive labels with correlation has become an essential challenge. In accordance with Figure 1, we can model a social network as a simple graph $G(V, E, \mathrm{VL}, \mathrm{EL})$, where V is a set of vertices, $E \subseteq V \times V$ is a set of edges, VL is a set of labels on the vertices, and EL is a set of labels on the edges. Here, a social network is modeled as a graph in which individuals are stated as vertices and their features as labels on vertices or edges.

In accordance with Figure 1, each of the vertices belongs to an individual. For example, vertex 7 belongs to Ada. Each of the edges represents a connection between two individuals. For example, the edge 7-2 represents a two-way connection between Ada and Bob. Each of the labels on the edges states a type of connection. For example, the label "110" on the edge 7-2 states just the relationship of friendship for Ada and Bob, but the label "111" on the edge 4-5 states three relationships of friendship, classmate, and roommate for Ed and George. Each



Sensitive labels on vertices and weight edges

Job	*Disease*	*Weight edge*
J: janitor	B: bronchitis	F: friendship
M: mover	P: pneumonia	C: classmate
C: carpenter	F: flu	R: roommate
T: technician	G: gastritis	
MA: manager	GU: gastric ulcer	
A: accountant	D: dyspepsia	
L: lawyer		

The labelsof vertices 1–9
v(name, quasi-identifier, disease, job)
1(*Dell*, 35, M, 59000, B, T)
2(*Bob*, 59, M, 11000, P, L)
3(*Cathy*, 70, F, 30000, G, C)
4(*Ed*, 65, F, 25000, D, J)
5(*George*, 65, F, 25000, GU, A)
6(*Irene*, 59, M, 11000, P, L)
7(*Ada*, 59, M, 11000, P, MA)
8(*Harry*, 27, M, 13000, B, M)
9(*Fred*, 61, F, 54000, F, J)

The label of edges
e: (FCR)

1-2: (100)	4-5: (111)
1–9: (110)	4–6: (100)
2–9: (100)	5-6: (100)
2–7: (110)	6-7: (101)
2-3: (110)	7-8: (110)
3-4: (100)	8-9: (111)

Figure 1: Original graph (G).

of the labels on the vertices represents the personal profile of an individual. For example, the labels of vertex 7 (i.e., "59, *M*, 11000, *P*, MA") state Ada's personal profile. That is, Ada is a 59-year-old man with a zip code of 11000, disease name of pneumonia, and job title of manager.

The labels on the vertices are typically distributed in four classes named "*explicit identifier label*," "*quasi-identifier label*," "*sensitive label*," and "*nonsensitive label*." The explicit identifier label is a set of labels, such as name and social security number, containing information that explicitly identifies data owners; the sensitive label, like the disease name and job title, is in fact an label whose values are confidential and the privacy of data owners; quasi-identifier labels like age, sex, and zip code are a set of labels through which the privacy of data owners can be threatened and their sensitive values are disclosed; and the nonsensitive label contains all labels that do not fall into the previous three classes. The four classes of labels are distinct.

1.1. Main Challenges. Clearly, to publish the original graph, explicit identifier labels must be removed. Typically, about a target victim, an adversary can have each of three background knowledge levels or combination of them: 1, the quasi-identifier of a target victim (i.e., the adversary knows Ada is a 59-year-old man with a zip code of 11000); 2, the target vertex degree (i.e., the adversary knows the vertex degree or the number of connection of Ada is 4); and 3, the weight edge of a target victim (i.e., the adversary knows that four weight labels on Ada's connections are the friendship and classmate relationships and two weight labels on Ada's connections are the roommate relationship). Here, we enumerate three main challenges, not all possible challenges: 1, vertex linkage challenge (VLC); 2, label linkage challenge (LLC); and 3, label correlation challenge (LCC).

1.1.1. VLC. Obviously, removing the explicit identifier label, like name, is not sufficient, singly. For example, according to Figure 1, if an adversary identifies Ada has four friends (vertex degree) and only two of his friends are friends (weight edge), then the adversary can identify the vertex and personal profile of Ada in Figure 1. In other words, there is a neighborhood attack [4] or vertex linkage challenge (VLC). In fact, VLC occurs when the target victim is linked to a specific vertex in the published graph. Clearly, to solve VLC, we can insert some additional edges among vertices [4] or insert additional vertices in the graph [5]. For example, by adding an additional edge (edge 6-8), the graph in Figure 2 presents that the vertex 7 (Ada) is not unique. Visibly, this action hides the vertex 7 (Ada) among two vertices 2 and 6. Each of the vertices 2, 6, and 7 belongs to Ada.

Note, VLC leads to the disclosure of sensitive labels on vertices. To solve VLC, many existing methods can be used, like k-anonymity and l-diversity in social networks [4].

1.1.2. LLC. Label linkage challenge (LLC) is studied in two modes: 1, LLC on edge and 2, LLC on vertex.

(1) LLC on Edge. In accordance with Figure 2, assume labels on the edges are distributed into two classes named "sensitive label" and "quasisensitive label." The sensitive label, like the roommate relationship, is in fact a label whose values can be confidential and the privacy of data owners. The quasisensitive label, like the friendship and classmate relationships, is in fact labels whose values are not confidential but are labels through which the privacy of data owners can be threatened. For example, if we find that Ada has four friends (vertex degree) and only three of his friends are classmates (weight edge), then we can identify the roommates of Ada in Figure 2. In this mode, LLC occurs when the sensitive values on the edge of a target victim are estimated, like the recent example. Clearly, according to Figure 3, to solve LLC on edge, we can insert a few additional edges, like edge 7–9, among some vertices or change the values of a few labels on some edges [5, 6]. Visibly, this action hides the sensitive labels on edges of Ada among other edges.

Note that LLC on edge leads to the disclosure of sensitive labels on edges. To solve LLC on edge, many existing methods can be used, like differential privacy for edge weights in social networks [6].

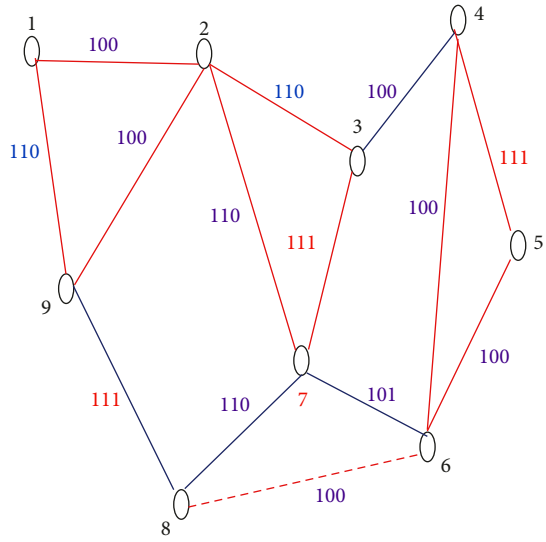


Figure 2: k-anonymity graph (AG).

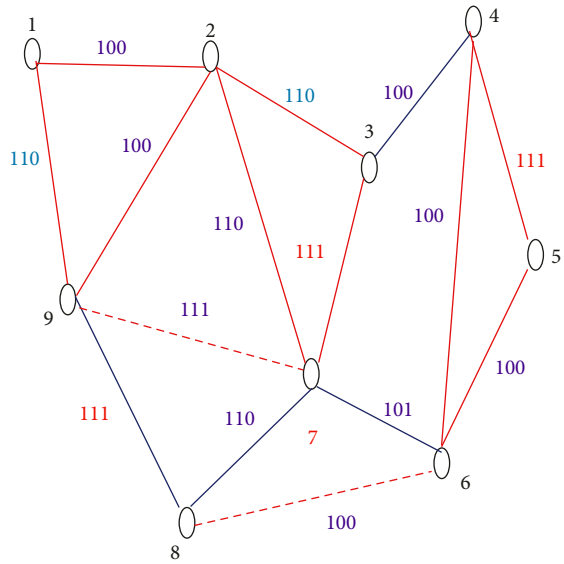


Figure 3: Anonymized graph (AG).

(2) LLC on Vertex. In accordance with Figure 2, besides solving VLC, the risk of attack still exists. For example, each of the three vertices 2, 6, and 7 in Figure 2 has four friends and their quasi-identifier and disease labels are equal to "59, *M*, 11000," and "*P*." Obviously, if we have the quasi-identifier of Ada, then we can identify Ada's disease name with 100% confidence. In this mode, LLC on vertex occurs when the sensitive values on the vertex of a target victim are estimated with a high confidence level of the published graph, like the recent example.

Another example is that, according to Figure 2, if we find that Ada has four friends, then we can find out that the disease name of Ada is "*P*." According to Figure 2, since just three vertices 2, 6, and 7 have four friends and their quasi-identifier and disease labels are the same, each of these vertices belongs to Ada.

At first glance, to solve LLC on vertex, we can apply each of the relational models for privacy preserving, like Anatomy [7] (or alpha anonymization [8]). We can conceive the set of labels on all vertices as a relational table and use Anatomy to protect labels. For example, Anatomy obtains the graph in Figure 3 as the input and returns data in Figure 4 and Tables 1 and 2 as the output for publishing.

Any table in which each group of tuples contains at least l separate sensitive values has the l-diversity property [9]. For example, each of Tables 1 and 3 has the 4-diversity property. In accordance with Figure 4 and Tables 1 and 3, Anatomy first partitions each of the original labels of vertices into buckets supplying the need of l-diversity; then, it extracts a graph called quasi-identifier graph (QIG) and a table called sensitive table (ST).

Visibly, this action hides the sensitive labels of vertices, in two separate tables (Tables 1 and 3). Note that the count label in Tables 1 and 3 refers to the frequency of disease labels and job labels in the original graph (Figure 1) and Group-ID and Group-ID2 refer to bucket identifiers in Tables 1 and 3.

Obviously, solving this challenge (LLC on vertex) results in weakening VLC because if an adversary can identify a target vertex, then he could deduce the sensitive values with a low percentage of confidence. Therefore, solving LLC on vertex is preferred rather than solving VLC.

Note that, in this work, we focus on the new challenge (i.e., LLC on vertex) and propose a solution to disable that.

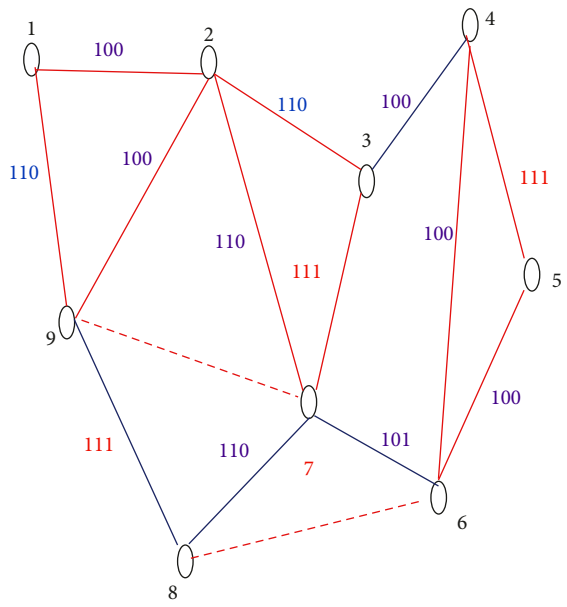
1.1.3. LCC. In accordance with Figure 4, by solving LLC on vertex, unfortunately, an important challenge named "label correlation challenge" (LCC) is found out. In Anatomy, LCC refers to that the correlation among labels extremely weakens. Preserving the correlation among labels and privacy is an important issue in publishing social networks which have multiple sensitive labels with correlation. Since, in real life, there is a correlation among multiple sensitive labels, for example, in medical data, the sensitive labels like disease, physician, symptoms, and treatment are correlated and breaching one can easily breach the other labels.

Multiple sensitive labels without correlation are a simple scenario in which the data publisher can use any conventional privacy model, like Anatomy, which supports static data. In fact, LCC occurs when we use a conventional privacy model which supports static data.

Finally, note that each of the two challenges VLC and LLC on edge leads to the disclosure of sensitive labels on vertices or edges. To solve each of these two challenges, many methods have been suggested. These methods usually edit the vertices, edges, or labels of the original graph (by removing, adding, or generalizing) [3–6, 8, 10–17]. Obviously, editing a vertex, edge, or label may intensify (or harden) LCC for some analyses such as membership analysis.

In this article, only focusing on two challenges LCC and LLC on vertex, we propose a new method which can almost supply correlation among labels for some analyses like membership analysis and solve LLC on vertex, as much as possible (expert's desired thresholds).

1.2. Membership Analysis. If the set of labels on vertices is conceived as a table, then it is immediately said that the



The labels of vertices 1–9
v(quasi-identifier, Group-ID, Group-ID 2)
1(35, M, 59000, 2, 1)
2(59, M, 11000, 2, 1)
3(70, F, 30000, 1, 1)
4(65, F, 25000, 2, 1)
5(65, F, 25000, 1, 2)
6(59, M, 11000, 1, 2)
7(59, M, 11000, 1, 1)
8(27, M, 13000, 1, 2)
9(61, F, 54000, 2, 2)

The label of edges
e: (FCR)
1-2: (100) 4-5: (111)
1–9: (110) 4–6: (100)
2–9: (100) 5-6: (100)
2–7: (110) 6-7: (101)
2-3: (110) 7-8: (110)
3-4: (100) 8-9: (111)
6–8: (100) 7–9: (111)

Figure 4: Quasi-identifier graph (QIG).

Table 1: Second sensitive table (SST).

Group-ID2	Job	Count
1	T	1
1	L	1
1	C	1
1	J	1
2	A	1
2	L	1
1	MA	1
2	M	1
2	J	1

Table 2: Result of the query F on Figures 11 and 12.

v	Q	P_1	S_1	F
8	27, *M*, 13000	RI	F	1
8	27, *M*, 13000	RI	P	3
8	27, *M*, 13000	RI	B	2

conventional models, like Anatomy, can be used to protect labels. Suppose we use Anatomy to protect labels. In accordance with Figure 4 and Tables 1 and 3, Anatomy typically uses Group-ID and Group-ID2 to reassociate the relationship between quasi-identifier and two sensitive labels (disease and job).

Table 3: First sensitive table (FST).

Group-ID	Disease	Count
1	P	2
1	B	1
2	B	1
2	P	1
2	F	1
2	D	1
1	GU	1
1	G	1

$Group_2$ $Group_1$

Anatomy supplies the privacy of data owners but loses the correlation among labels and affects the accuracy of some data analyses like membership analysis which defeats Anatomy. Membership analysis is studied in two cases: 1, one sensitive label and 2, multiple sensitive labels.

1.2.1. One Sensitive Label. Suppose, by using query *A*, we would like to find out the personal profile of individuals who are in the category of "SD" (stomach disease).

A:

Select distinct vertex
From original graph
Where disease in {SD}

Note that, in all queries, like query *A*, to remove the redundant vertices, the clause "*distinct*" has used. In accordance with the taxonomic tree in Figure 5, only "GU," "*D*," and "*G*" are in the category of "SD."

If Figure 1 (original graph) is available, then the result of running the query *A* is shown in Figure 6 which only has 3 valid vertices.

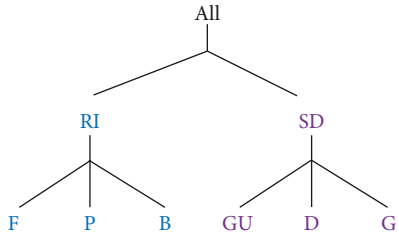
If Figure 4 and Table 3 are available, then the easiest way is to run query *B*.

B:

Select distinct vertex
From Figure 4 and Table 3
Where Group-ID of Table 3 = Group-ID of Figure 4
Disease in ("GU," "D," "G")

The result of running the query *B* is shown in Figure 7 in which 3 vertices are valid and 6 vertices are invalid. So, we distribute all vertices of a resultant graph into two distinct classes: 1, valid vertex and 2, invalid vertex. Any vertex in Figure 7 whose equivalent can be found in Figure 6 is a valid vertex; otherwise, it is an invalid vertex. For example, the vertices 3-5 in Figure 7 are valid and the vertices 1-2 and 6-9 are invalid. Note that isolating valid vertices and invalid vertices of the resultant graph is difficult without having the original graph or any other information.

To determine the accuracy and error of the membership analysis, we define the membership accuracy rate (MA) and the membership error rate (ME). Let NV be the number of



RI: respiratory infection *P: pneumonia*
SD: stomach disease *B: bronchitis*
GU: gastric ulcer *D: dyspepsia*
F: flu *G: gastritis*

Figure 5: Taxonomic tree of disease.

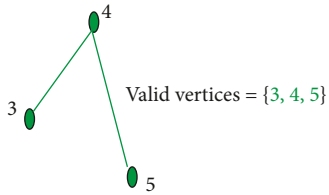
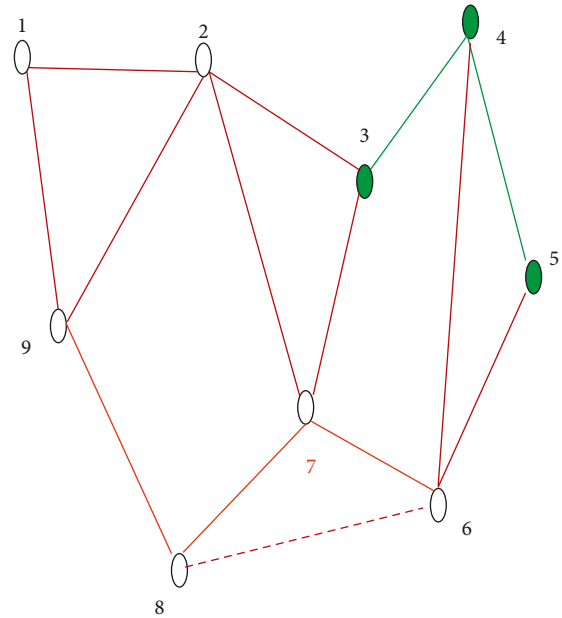


Figure 6: Result of running the query *A*.



Valid vertices = {3, 4, 5}
Invalid vertices = { 1, 2, 6, 7, 8, 9}

Figure 7: Result of running the query *B*.

valid vertices and NI be the number of invalid vertices, then the MA can be stated by the following equation:

$$\text{MA} = \frac{\text{NV}}{\text{NV} + \text{NI}}. \quad (1)$$

But the MA criterion is exactly opposite to the ME criterion obtained using the following equation:

$$\text{ME} = \frac{\text{NI}}{\text{NV} + \text{NI}} = 1 - \text{MA}. \quad (2)$$

The smallest value of the ME is 0, when we have the best performance, and the largest value of the ME is 1, when we have the lowest performance. Consequently, the MA and ME for the query *B* are 0.33 and 0.67 (MA = 3/9 and ME = 6/9). Note that Anatomy extremely reduces the accuracy of such membership analysis because it randomly groups sensitive values [10]. In other words, Anatomy is an unsupervised anonymization technique.

1.2.2. Multiple Sensitive Labels. Suppose the disease name and job title are sensitive labels and we would like to find out the personal profile of individuals who are in categories of "SD" and "WC" (white-collar), by query *C*.

C:

Select distinct vertex
From Figure 1
Where disease in {SD} and job in {WC}

In accordance with Figure 8, only the titles of "MA," "L," and "A" are in the category of "WC." If Figure 1 is available, then the result of running the query *C* is just the vertex 5. But if Figure 4 and Tables 1 and 3 are available, then the easiest way is to run query *D*.

D:

Select distinct vertex
From Figure 4 and Tables 1 and 3
Where Group-ID of Table 3 = Group-ID of Figure 4 and
Group-ID2 of Table 1 = Group-ID2 of Figure 4 and
Disease in ("GU," "D," "G")
And job in ("MA," "L," "A")

The result of the query *D* is Figure 9 whose vertex 5 is only valid. In accordance with equations (1) and (2), the MA and ME for the query *D* are 0.11 and 0.89 (MA = 1/9 and ME = 8/9). In fact, in Anatomy, as the number of sensitive labels increases, the MA extremely decreases. Accordingly, Anatomy weakens the correlation among multiple sensitive labels.

1.3. Motivation. To solve VLC, any conventional anonymization methods can be applied, like *k*-anonymity [4, 18, 19], *l*-diversity [20], and differential privacy [6] in social networks.

At first glance, to solve LLC on vertex, each of the relational models can be applied, like personalized privacy [21], MNSA [22], SLOMS [23], Anatomy [7], generalization [2], anatomization with slicing [24], a novel approach for personalized privacy [25], effective privacy preserving [26], and a privacy-preserving model for 1 : M data [27]. Unfortunately, conventional models disconnect or reassociate the relationships among labels, or change the structure of the graph. The relationship disconnection results in information loss, and the reassociation reduces the correlation among sensitive labels for some analyses like membership analysis; but changing the structure results in varying the properties of the graph.

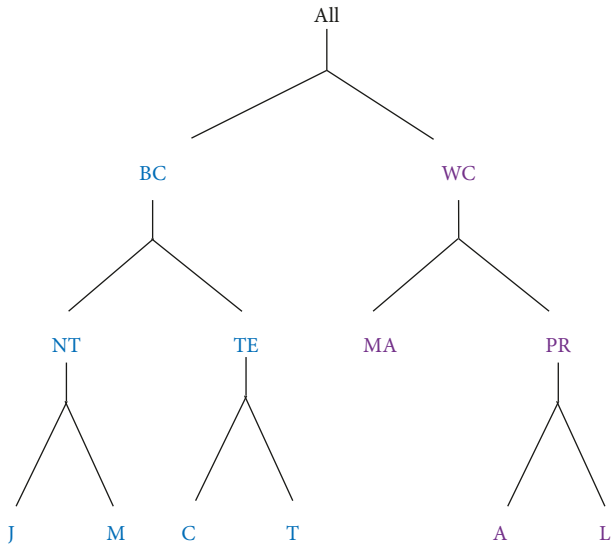


Figure 8: Taxonomic tree of job.

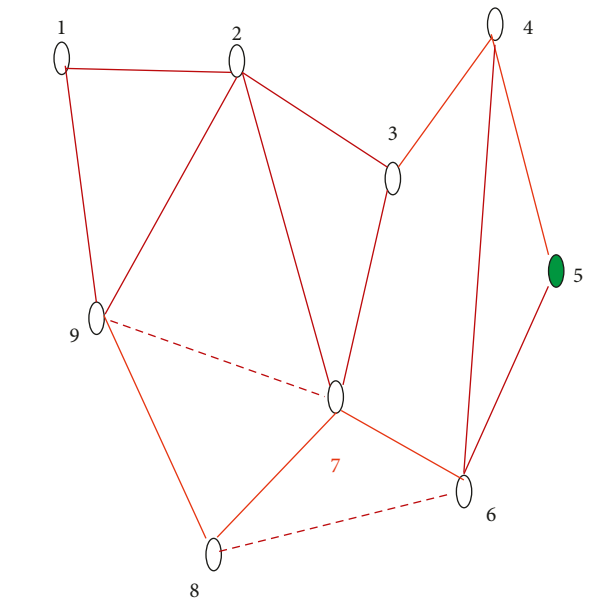


Figure 9: Result of running the query D.

Conceptually, according to Figure 5, a set of the internal nodes in the taxonomy tree whose values can be published are named "attack threshold." For example, in Figure 5, the nodes like "RI" and "SD" are an attack threshold. In other words, they are the defined categories of diseases.

Quantitatively, an attack threshold quantity for a sensitive label denotes the maximum probability of disclosing original values of that label. For example, for the disease label, an attack threshold of 0.5 states that the maximum probability of disclosing the disease name of any individual is equal to 0.5.

In real life, since an adversary may find out the categories of sensitive labels of individuals (i.e., the categories of the disease for patients), it is not necessary that we protect the categories of sensitive labels (i.e., disease categories), but protecting the original values of sensitive labels (i.e., disease name) is necessary. Hence, by decreasing sensitivity of sensitive labels to an expert's desired attack thresholds, we want to propose a new method to publish social networks having multiple sensitive labels with correlation, without distorting original labels or disconnecting their correlation.

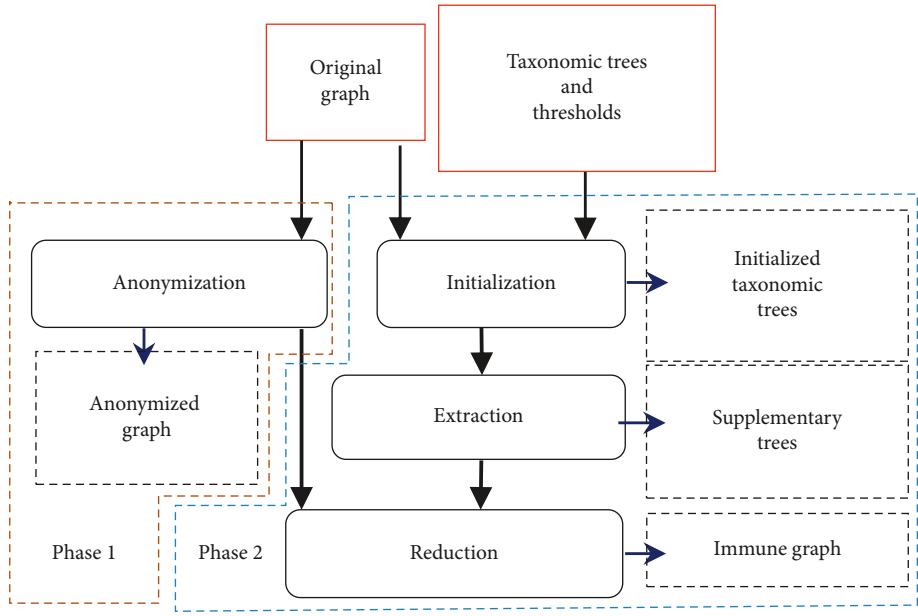
In fact, for some analyses like membership analysis, we want to increase the correlation among sensitive labels of published graphs and decrease the membership error rate, by reducing the sensitivity of the sensitive values up to the expert's desired attack threshold. Thus, focusing on LCC, we propose a new method which can almost supply correlation among labels for membership analysis and solve LLC on vertex.

1.4. Rationale of SNI. In this article, we propose a supervised technique named "SNI" which solves two challenges, LLC on vertex and LCC. Construction of SNI can be understood from the framework in Figure 10. In this framework, it is assumed that sensitivity of the sensitive labels can be decreased up to the attack thresholds, and also each sensitive label has a taxonomic tree and an attack threshold.

As in Figure 10, if an original graph (G) contains multiple sensitive labels, SNI will produce one *immune graph* (IG) and multiple *supplementary trees* (STs) to publish the original graph. For example, it produces Figures 11–13 instead of publishing Figure 1 which contains two sensitive labels (disease and job). Figure 11 shows an IG, and Figures 12 and 13 show two STs. SNI, for each sensitive label, produces an immune label named "*partial sensitive label*," like partial disease and partial job labels in Figure 11. From the view point of Anatomy, any partial sensitive label plays the role of a Group-ID, and from the view point of the expert (or data owners), it is a label with less sensitivity that can be published.

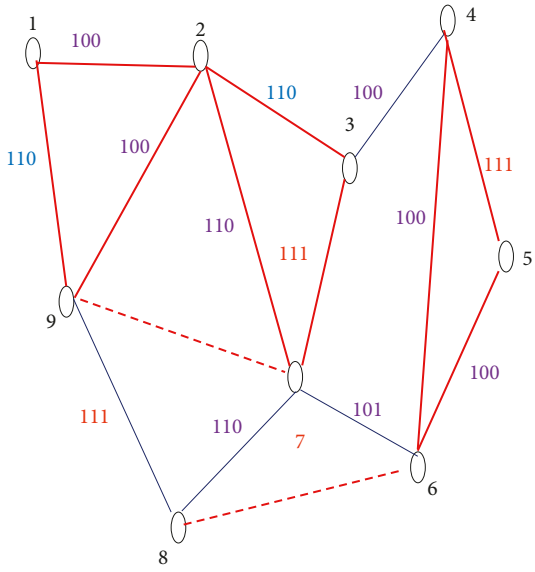
In order to reduce information loss, SNI presents the quasi-identifier and sensitive labels in separate objects (graph and trees). SNI uses the partial sensitive label to reestablish the relationship between quasi-identifier and sensitive labels. In accordance with Figure 10, the framework contains 2 phases.

1.4.1. Phase 1. This phase to solve VLC and LLC on edge has been considered. In this phase, we can apply each of the algorithms, like k-anonymity, l-diversity, and differential privacy, and extract an anonymized graph. Note that the algorithm that is used in phase 1 is not allowed to edit or change original values of sensitive labels on vertices. For example, if Figure 1 is the input of phase 1, then its output will be Figure 3.



Phase 1: solving VLC and LLC of edges
Phase 2: solving VLC and LLC of vertices

Figure 10: SNI framework.



The labels of vertices 1–9
v(quasi-identifier, partial disease, partial job)
1(35, M, 59000, RI, BC)
2(59, M, 11000, RI,WC)
3(70, F, 30000, SD, BC)
4(65, F, 25000, SD, BC)
5(65, F, 25000, SD, WC)
6(59, M, 11000, RI, WC)
7(59, M, 11000, RI, WC)
8(27, M, 13000, RI, BC)
9(61, F, 54000, RI, BC)

The labels of edges
e: (FCR)
1-2: (100) 4-5: (111)
1–9: (110) 4–6: (100)
2–9: (100) 5-6: (100)
2–7: (110) 6-7: (101)
2-3: (110) 7-8: (110)
3-4: (100) 8-9: (111)
6–8: (100) 7–9: (111)

Figure 11: Immune graph (IG).

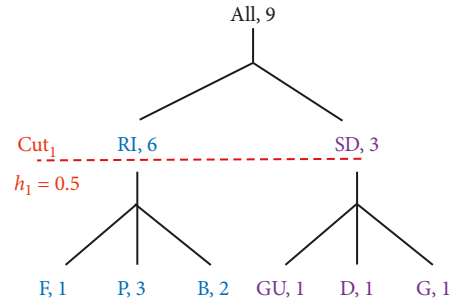


Figure 12: Supplementary tree of disease (ST_1).

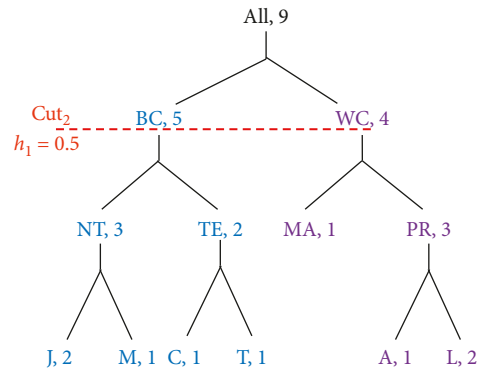


Figure 13: Supplementary tree of job (ST_2).

1.4.2. Phase 2. This phase which is the main focus of this article has 3 activities: 1, *initialization*; 2, *extraction*; and 3, *reduction*.

(1) Initialization. In this activity, for any leaf x in a taxonomic tree, a field F is considered so that $x[F]$ refers to the frequency of the leaf x in the original graph. However, this activity receives an original graph and multiple taxonomic trees as the input (i.e., Figures 3, 5, and 8) and just updates $x[F]$ for all leaves x in taxonomic trees. For example, in Figure 12, $P[F]$ (frequency of the leaf "P" or "*pneumonia*") is 3 because the frequency of "P" in Figure 1 equals 3 (disease values of vertices 2, 6, and 7).

(2) Extraction. In this activity, for any internal node y of any taxonomic tree, two fields (F, R) are considered so that $y[F]$ and $y[R]$ refer to the sum of $x[F]$ of all leaves x in the subtree of y and the maximum of $x[R]$ of all leaves x in the subtree of y, respectively. In other words, if the subtree of y contains n leaves (i.e., $x_1, \ldots, x_n$), then

$$y[F] = \sum_{i=1}^{n} x_i[F], \tag{3}$$

$$x_i[R] = \frac{x_i[F]}{y[F]}, \tag{4}$$

$$y[R] = \mathrm{Max}\left(x_1[R], \ldots, x_n[R]\right), \tag{5}$$

where $x_i[F]$ and $x_i[R]$ are the frequency and relative frequency of the leaf x_i in the subtree of y. For example, for the internal node "RI" (*respiratory infection*) in Figure 12 which contains three leaves "P," "F," and "B" (that is, *pneumonia*, *flu*, and *bronchitis*), according to equations (3)–(5), we have

$$\begin{aligned} \mathrm{RI}[F] &= P[F] + B[F] + F[F] = 3 + 2 + 1 = 6, \\ P[R] &= \frac{3}{6} = 0.5, \\ F[R] &= \frac{1}{6} = 0.17, \\ B[R] &= \frac{2}{6} = 0.33, \\ \mathrm{RI}[R] &= \mathrm{Max}\left(\frac{3}{6}.\frac{1}{6}.\frac{2}{6}\right) = \frac{3}{6} = 0.5. \end{aligned} \tag{6}$$

Extraction, for any taxonomic tree, finally sets a permitted generalization limit named "Cut." For example, for Figure 12, it determines some internal nodes, like "RI" and "SD," named "Cut_1." Obviously, any internal node y whose $y[R]$ is smaller or equal to the attack threshold ($h_1 = 0.5$) can be inserted in Cut_1.

(3) Reduction. This activity firstly receives an anonymized graph (output of phase 1 like Figure 3) and all updated taxonomic trees (output of extraction activity like Figures 12 and 13) and produces an IG whose vertices contain three labels (*quasi-identifier label*, *partial sensitive label*, and *nonsensitive label*). For example, if Figures 3, 12, and 13 are the input of reduction, then its output is Figure 11.

However, SNI can publish Figure 11 (IG) with Figures 12 and 13 (ST_1 and ST_2). Obviously, the output of SNI depends not only on the frequency of sensitive values but also on the structure of taxonomic trees. Figures 11–13 solve VLC, LLC, and LCC because Figure 11 contains partial sensitive values, and real sensitive values can only be estimated from Figures 12 and 13.

Assume we would like to disclose Ada's disease name and job title through quasi-identifiers of Ada, by using Figures 11–13. If we can understand that each of the vertices 2, 6, and 7 in Figure 11 belongs to Ada, then VLC is solved; if edge 7-9 is a noise edge, then LLC on edge is disabled. But since we do not obtain accurate information on disease and job labels and can only obtain information with partial sensitivity through partial disease and partial job labels in Figure 11, then LLC on vertex is weakened. It is obvious that we should use Figures 12 and 13 for more accurate information. We can find that Ada's disease name is in category "RI." By referring to Figure 12, we find that his disease name is likely to be 0.5 "P" and 0.33 "B" and 0.17 "F." We through the partial job label can find that Ada's job title is in category "WC." By referring to Figure 13, we find that his job is likely to be 0.25 "A" and 0.25 "M" and 0.5 "L." Hence, LLC on vertex is disabled. Since SNI supplies the correlation among sensitive labels, by preserving the correlation among partial sensitive labels, LCC is partly solved.

1.5. Contribution. This paper presents a systematic study of the SNI technique. First, we formalize the new methodology based on the attack thresholds and taxonomic trees.

Second, we prove that SNI significantly supplies the privacy preservation at the vertex level (PPVL) and the privacy preservation at the individual level (PPIL). In PPVL, if there are several vertices with the same quasi-identifier, then each of them is considered separately and its sensitive values are estimated. But, in PPIL, for all vertices with the same quasi-identifier which may belong to one individual, one sensitive value is estimated. For example, vertices 2, 6, and 7 in Figure 6 contain the quasi-identifier "59, M, 11000." In PPVL, each of the vertices 2, 6, and 7 is considered separately and its sensitive labels are estimated. In PPIL, the sensitive labels are once estimated for three vertices 2, 6, and 7.

Third, we prove that the maximum probability of disclosing of a sensitive value for any individual is smaller than or equal to an attack threshold, so SNI will disable LLC on vertex.

Fourth, we prove that SNI increases the MA and decreases the ME, significantly. To carry out this, we consider two cases for the clauses in queries: 1, the clauses above the permitted generalization limits and 2, the clauses below the permitted generalization limits.

Fifth, we develop an algorithm that produces an IG and multiple STs. Finally, we prove by extensive experiments that SNI significantly outperforms Anatomy, and it almost solves LCC.

The rest of this paper is organized as follows: First, we will provide a background for anonymization methods. Second, we will formalize SNI based on the attack threshold

and taxonomic tree. Third, we will prove that SNI compared with Anatomy provides more correlation for data in membership analysis. Fourth, we will present an algorithm for SNI which carries out the major actions on taxonomic trees. Fifth, using heavy experiments, we will demonstrate that SNI compared with Anatomy supplies a more considerable utility for membership analysis. In the end, we will finish the paper with directions for future work.

2. Background

To solve VLC or LLC on edge, each of the anonymization conventional models can be used to protect social networks. These models usually edit vertices, edges, or labels on the original graph. For example, in 2012, Song et al. [14] presented a model named "sensitive label privacy on social networks." This model transforms the original graph into a graph whose vertices are sufficiently indistinguishable. It does not allow an adversary, by using the information about the neighborhood of a vertex, to disclose sensitive labels of that vertex.

In 2014, Yuan et al. [13] presented a model named "personalized privacy on social networks." This model applies two techniques generalizing labels and editing vertices or edges (adding the noisy edge or vertex) to supply the privacy-preserving service.

In 2014, Sun et al. [28] presented the k-NMF-anonymity model which preserves original vertices (no deletion) and adds new vertices.

In 2017, Li et al. [6] presented a model named "differential privacy method for edge weights in social networks." In this method, a certain amount of random noise is added to the answer of the query set.

In 2018, Baktha and Tripathy [8] presented the alpha anonymization model. In fact, this model extends the lossy-join approach [29] to achieve (α, k)-anonymity in relational data [30].

In 2019, a novel technique named "a novel graph-modification technique for user privacy preserving on social networks" has been presented [5]. This technique is based on edge addition.

Usually, such models change the original structure of the graph, by adding, removing, or modifying vertices, edges, or labels. Unfortunately, changing the structure may result in varying the properties of the graph.

At first glance, to solve LLC on vertex, each of the relational models can be used. For example, Samarati and Sweeney [2] proposed the generalization technique. The disadvantage of generalization is inflicting loss to the data.

Sweeney suggested the concept of k-anonymity [19]. A group of tuples with a quasi-identifier value is named "QI-group." Any table in which cardinality of any QI-group is at least k has a k-anonymity property. k-Anonymity always is not enough because several records in a QI-group belonging to one person still have a potential privacy threat for that person.

Machanavajjhala et al. suggested the concept of l-diversity [9]. Since some sensitive values are naturally more abundant than other values in a QI-group, it is not always possible to obtain the l-diversity property for any table.

Wang et al. proposed the confidence bounding method [31]. In this method, a maximum confidence level is specified to infer any value of the sensitive attribute in a QI-group.

Xiao and Tao proposed the Anatomy technique to maintain the confidentiality of data, which, despite the utility of data for aggregative analysis, lacks the necessary utility of data mining [7]. The advantage of Anatomy is that it does not change the sensitive attribute and quasi-identifier values at all but only weakens the association between sensitive and quasi-identifier attributes.

Xiao and Tao introduced another method called personalized privacy preservation, which allows the data owner, without notice to the utility of data, to set an attack threshold named the guarding node on a taxonomic tree belonging to a sensitive attribute [21]. Unfortunately, the data owners do not have access to the distribution of sensitive values in a QI-group or the entire published table when setting their attack threshold.

In 2013, Han et al. presented the SLOMS method to publish data with several sensitive attributes [23]. This method partitions horizontally the sensitive attributes in several tables, and each table buckets the sensitive attribute to provide the l-diversity while also generalizing the quasi-identifier values to provide k-anonymity. Generalization causes information loss.

In 2015, Liu et al. proposed the MNSA method for several numerically sensitive attributes. Unfortunately, this method can only be used for numerically sensitive attributes [22].

In 2016, Susan and Christopher introduced a new method called anatomization by slicing to maintain the confidentiality of high-dimensional data with multiple sensitive attributes [24]. This method uses a slicing technique to increase the association of attributes and uses the anatomization technique to reduce information loss, and because it uses Group-ID to reassociate the relationship between attributes, it reduces the utility of the data.

In the subtree generalization scheme [32, 33], if the value of a leaf in a taxonomic tree generalizes to the value of its ancestor node such as x, then all the leaves in the subtree of x should also generalize to the value of x. For example, if the leaf "F" in Figure 5 generalizes to the value of its ancestor like "RI," then only values of "P" and "B" should be generalized to "RI."

In 2018, the composition attacks were studied by Hasan et al. [34]. They believe that, in two hospitals, if a patient is examined for a similar disease and his information is independently distributed, it would be the probability of a composition attack. They have used the cell generalization approach [35, 36] to disable the composition attack and increase data utility.

In 2019, the new methods have been presented to publish [26, 27] data with multiple sensitive attributes. 1 : M means an individual can have multiple records with multiple sensitive attributes.

In the proposed technique (SNI), from the view point of the confidence bounding method, each attack threshold is considered to be a maximum confidence level. From the outlook of Anatomy, each partial sensitive label plays the role of a Group-ID. From the view point of the personalized privacy preservation method, each permitted generalization limit plays the role of guarding nodes on a taxonomic tree. SNI, contrary to the personalized privacy preservation method that allows any data owner to define a guarding node for himself, allows the data holder to define only one attack threshold for all values of a sensitive label. Because the distribution of sensitive values is available in the entire vertices on the original graph, the partial sensitive values (guarding nodes) are properly calculated. The new partial sensitive label, in addition to being used in reconstructing the correlation of labels, is used by many analyses, especially membership analyses. SNI uses the subtree generalization scheme which is usually more practical for categorization [37].

3. Formalization of SNI

Suppose, according to Figure 1, each vertex v in a simple graph G (V, E, VL, EL) has a quasi-identifier label and b sensitive labels and is expressed as

$$v = (v[Q_1], v[Q_2], \ldots, v[Q_a], v[S_1], v[S_2], \ldots, v[S_b]), \tag{7}$$

where the symbols of S_i and Q_i refer to the ith sensitive label and ith quasi-identifier label on the vertex $v \in G$, respectively; $v[Q]$ refers to the values of all quasi-identifier labels on v (i.e., values of $v[Q_1]$, $v[Q_2]$, ..., $v[Q_a]$); and $v[S_i]$ refers to the value of the ith sensitive label on v. Anyhow, Q_i can be numerical or categorical ($1 \le i \le a$), but S_i must be categorical ($1 \le i \le b$).

The proposed technique requires a taxonomic tree and an attack threshold be determined for each sensitive label by the expert. The symbols of TAX_i and h_i refer to a taxonomic tree and an attack threshold for S_i, respectively.

Suppose any leaf $l \in \mathrm{TAX}_i$ has three fields that is expressed as

$$l = (l[N], l[F], l[R]), \tag{8}$$

where $l[N]$, $l[F]$, and $l[R]$ refer to the value of the ith sensitive label in G, the frequency of l in G, and the relative frequency of l in a specified subtree of TAX_i, respectively. For example, in Figure 12, for leaf $l =$ "p" in the subtree of "RI," we have

$$\begin{aligned} l[N] &= \text{“}P\text{”}, \\ l[F] &= 3, \\ l[R] &= \frac{3}{3+2+1} = 0.5. \end{aligned} \tag{9}$$

Note that the relative frequency of "P" in the subtree of "RI" has been calculated.

Assume any internal nodes $n \in \mathrm{TAX}_i$ have three fields (N, F, R) that are expressed as

$$n = (n[N], n[F], n[R]), \tag{10}$$

where $n[N]$, $n[F]$, and $n[R]$ refer to the category name of all leaves, sum of the frequencies of all leaves, and maximum of the relative frequencies of all leaves, in the subtree of n. For exampluations 3, 4, and 5, for the internal node $y =$ "RI," we have

$$\begin{aligned} y[N] &= \text{“RI”}, \\ y[F] &= \sum_{\text{All } x \in \text{ subtree of } y} x[F] = 2 + 2 + 1 = 5, \\ y[R] &= \mathrm{Max}\,(\text{All } x[F] \text{ where } x \in \text{ subtree of } y) \\ &= \mathrm{Max}\,(0.5, 0.17, 0.33) = 0.5. \end{aligned} \tag{11}$$

Assuming that we can decrease the sensitivity of the sensitive values of an individual to the expert's desired attack thresholds, we present some new concepts for formalization of SNI.

3.1. Concepts. SNI requires specifying the immune nodes and permitted generalization limits for any taxonomic tree.

Definition 1 (immune node). An internal node $y \in \mathrm{TAX}_i$ in which $y[R] \le h_i$ is considered to be an immune node in TAX_i.

For example, in Figure 13, the nodes of "WC," "BC," and "TE" are immune nodes because for $h_2 = 0.5$ and all $y \in$ {"WC," "BC," "TE"}, we have

$$y[R] \le h_2. \tag{12}$$

Definition 2 (Cut_i). If a set of leaves in a subtree of the kth immune node is referred to as Set_k, then the immune node $m \in \mathrm{TAX}_i$ which satisfies two equations (6) and (7) is considered to be a Cut_i for TAX_i; for $1 \le k \ne j \le m$ and leaf $l \in \mathrm{TAX}_i$,

$$\cup_{k=1}^{m} \mathrm{Set}_k = \{\text{all } l \in \mathrm{TAX}_i\}, \tag{13}$$

$$\mathrm{Set}_k \cap \mathrm{Set}_j = \varnothing. \tag{14}$$

For example, in Figure 13, Cut_2 contains only two immune nodes "WC" and "BC" because

$$\begin{aligned} \mathrm{Set}_{\mathrm{WC}} &= \{\text{“}J\text{”}, \text{“}M\text{”}, \text{“C”}, \text{“}T\text{”}\}, \\ \mathrm{Set}_{\mathrm{BC}} &= \{\text{“MA”}, \text{“}A\text{”}, \text{“}L\text{”}\}, \\ \mathrm{Set}_{\mathrm{WC}} \cup \mathrm{Set}_{\mathrm{BC}} &= \{\text{all leaves of } \mathrm{TAX}_{\mathrm{Job}}\}, \\ \mathrm{Set}_{\mathrm{WC}} \cap \mathrm{Set}_{\mathrm{BC}} &= \varnothing. \end{aligned} \tag{15}$$

Definition 3 (partial sensitive label). For each S_i on vertices $v \in G$, a new label named "partial sensitive label" (P_i) is defined and its values are derived from the generalization of S_i to Cut_i contained in TAX_i. In fact, the domain of P_i is Cut_i.

For example, in Figure 11, the partial disease label is P_1 derived from the generalization of the disease label to Cut_1 in Figure 12. In fact, any P_i reduces sensitivity of S_i to Cut_i.

Definition 4 (SNI). SNI always obtains an original graph (G) and b attack thresholds ($h_1, \ldots, h_b$) and b taxonomic trees ($\text{TAX}_1, \ldots, \text{TAX}_b$) as the input and produces b permitted generalization limits ($\text{Cut}_1, \text{Cut}_2, \ldots, \text{Cut}_b$), b supplementary trees ($\text{ST}_1, \text{ST}_2, \ldots, \text{ST}_b$), and an immune graph (IG) as the output. Each Cut_i represents a permitted limit to generalizing S_i. Each ST_i is related to $S_i \in G$ and is expressed as

$$\text{ST}_i = \{S_i, P_i, F_i\}. \tag{16}$$

The symbol of F_i refers to the frequency of S_i in G. The domain of $S_i \in \text{ST}_i$ equals the domain of $S_i \in G$, and Cut_i is considered the domain of P_i. But IG is expressed as

$$\text{IG} = (\text{IV}, \text{IE}, \text{IVL}, \text{IEL}), \tag{17}$$

where IV is a set of vertices, $\text{IE} \subseteq \text{IV} \times \text{IV}$ is a set of edges, IVL is a set of labels on vertices, and IEL is a set of labels on edges. IV is equivalent to V, IE is equivalent to E plus the new edges, and IEL is equivalent to EL plus the labels on new edges. But IVL contains all quasi-identifier and nonsensitive labels in VL plus the new labels named "partial sensitive labels." Any vertex $v' \in \text{IV}$ has a quasi-identifier label and b partial sensitive labels and is expressed as

$$v' = (v'[Q_1], v'[Q_2], \ldots, v'[Q_a], v'[P_1], v'[P_2], \ldots, v'[P_b]), \tag{18}$$

where the symbols of P_i and Q_i refer to the ith partial sensitive label and ith quasi-identifier label in IVL, respectively. Each Q_i of IVL is equivalent to one Q_i of L. Each P_i of IVL is equivalent to one immune node in Cut_i whose publication and disclosure are allowed.

For example, SNI receives Figure 1 as G, taxonomic trees in Figures 5 and 8 as TAX_1 and TAX_2, and $h_1 = h_2 = 0.5$ and extracts Figure 11 as IG and Figures 12 and 13 as ST_1 and ST_2.

In fact, SNI uses partial sensitive labels to preserve the correlation among labels. IG actually contains sensitive values with a lower sensitivity degree that can be published and preserves correlation among labels.

3.2. Privacy Requirements. Obviously, since none of $b\,\text{ST}_i$ have quasi-identifier labels and IGs have no sensitive labels, each of those protects the privacy, alone. But the level of privacy protection in any dataset derived from $\text{IG} \infty \text{ST}_1 \ldots \infty \text{ST}_b$ (natural join) should be checked. The query E is a case of natural join.

E:

Select the vertex number of the IG, quasi-identifier, Partial sensitive labels, sensitive labels, frequency
From $ST_1, ST_2, \ldots, ST_n$, *IG*
Where P_1 *of* $ST_1 = P_1$ *of IG and*
P_2 *of* $ST_2 = P_2$ *of IG and*
.
.
P_b *of* $ST_b = P_b$ *of IG and*
Q *of IG* $= v[Q]$

The symbol of $v[Q]$ refers to the quasi-identifier of the target vertex $v \in G$. In fact, the query E states if there are IGs with several ST_i, an adversary can partially reconstruct the vertex $v \in G$ belonging to a target victim. In this section, we want to prove that SNI significantly supplies PPVL and PPIL.

3.2.1. PPVL. In PPVL, if n vertices $v_1, \ldots, v_n \in \text{IG}$ (i.e., the vertices 2, 6, and 7 in Figure 11) are found whose quasi-identifiers are equal to a target victim, each of them is considered separately, and its sensitive values are estimated. SNI will supply PPVL, if we can reconstruct a sensitive value of any vertex $v \in G$ when the maximum probability equals h_i or all sensitive values of any vertex $v \in G$ when the maximum probability equals $\prod_{i=1}^{b} h_i$. We will study two modes: 1, one ST_i with IG and 2, multiple ST_i with IG.

(1) First Mode (One ST_i *).* Obviously, to reconstruct a vertex $v \in G$ by one ST_i and an IG, firstly we must run the query F on the IG and ST_i and then estimate $v[S_i]$.

F:

Select vertex number of the IG, quasi-identifier, Partial sensitive labels, sensitive labels, frequency
From ST_i, *IG*
Where P_i*of IG* $= P_i$ *of* ST_i *and* Q *of IG* $= v[Q]$

Clearly, the result of running the query F is a table as T_q (i.e., Table 2) which has tuples such as

$$x = (x[\text{ID}], x[Q], x[P_i], x[S_i].x[F]), \tag{19}$$

where $x[\text{ID}]$, $x[Q]$, $x[P_i]$, $x[S_i]$, and $x[F]$ are the vertex number of the IG, quasi-identifier, ith partial sensitive label, ith sensitive label, and frequency of the tuples $x \in T_q$, respectively.

For example, Table 2 as T_q is the result of running the query F on Figures 11 and 12, where $v[Q] =$ "27, M, 13000." According to the clause "Q of IG $= v[Q]$" in the query F, the quasi-identifiers of all tuples $x \in T_q$ equal the quasi-identifier of the target vertex $v \in G$, and according to the clause "P_i of $\text{ST}_i = P_i$ of IG," the partial sensitive labels of all tuples in T_q are equal to the partial sensitive label of the target vertex $v \in G$. When there is one ST_i with an IG, the PPVL is described in Lemma 1 formally.

Lemma 1. *If* T_q *is the result of running the query F on IG and one* ST_i*, then an adversary can reconstruct the ith sensitive label of a target vertex* $v \in G$ *up to* h_i*. In other words, for* $x \in T_q$ *and* $v \in G$,

$$\Pr\{v[S_i] = x[S_i]\} \leq h_i. \tag{20}$$

Proof. Firstly, according to the clause "Q of IG $= v[Q]$" in the query F, since the quasi-identifier of all tuples in T_q is equal to

$v[Q]$, any tuple $x \in T_q$ can belong to the target vertex $v \in G$. Secondly, according to the clause "P_i of $\mathrm{ST}_i = P_i$ of IG," since the result of running the query F is a table as T_q (i.e., Table 2) in which all tuples are exactly equal to all leaves in the subtree of $P_i \in \mathrm{Cut}_i$, and on the contrary, in any ST_i, P_i is an immune node. Then, according to Definitions 1 and 2, we have

$$\Pr\{v[S_i] = x[S_i]\} = x[R] \leq h_i, \tag{21}$$

where $x[R]$ is the relative frequency of the tuple $x \in T_q$ or the relative frequency of the leaf x in the subtree of the immune node $P_i \in \mathrm{Cut}_i$. So, according to equation (9), Lemma 1 is correct.

Using Table 2, we will describe Lemma 1. As in Table 2, for $v[Q] =$ "27, M, 13000," we have

$$\begin{aligned}
\Pr\{v[S_1] = \text{“}P\text{”}\} &= \frac{3}{3+2+1} = 0.5,\\
\Pr\{v[S_1] = \text{“}B\text{”}\} &= \frac{2}{3+2+1} = 0.33,\\
\Pr\{v[S_1] = \text{“}F\text{”}\} &= \frac{1}{3+2+1} = 0.17.
\end{aligned} \tag{22}$$

On the contrary, since $h_1 = 0.5$, we can see that equation (8) is correct.

(2) Second Mode (Multiple ST_i). To reconstruct a target vertex $v \in G$ through b ST_i and an IG, firstly we must run the query E on b ST_i and an IG and then estimate $v[S_1] \cdots v[S_b]$, simultaneously. Clearly, as in the previous mode, the result of running the query E is a table as T_q (i.e., Table 4) which has tuples such as

$$x = (x[\mathrm{ID}], x[Q], x[P_i], x[S_1], \ldots, x[P_b], x[S_b], x[F]). \tag{23}$$

For example, Table 4 as T_q is the result of running the query E on Figures 11–13, where $V[Q] =$ "27, M, 13000." But when there are multiple ST_i with an IG, PPVL is described in Lemma 2 formally. □

Lemma 2. *If T_q is the result of running the query E on IG and b ST_i, then an adversary can simultaneously deduce the values of b sensitive labels of any target vertex $v \in G$ up to $\prod_{i=1}^{b} h_i$. In other words, for $x \in T_q$ and $v \in G$,*

$$\Pr\{v[S_1] = x[S_1], \ldots, v[S_b] = x[S_b]\} \leq \prod_{i=1}^{b} h_i. \tag{24}$$

Proof. Firstly, according to the clause "Q of IG $= v[Q]$" in the query E, since the quasi-identifier of all tuples in T_q is equal to $v[Q]$, any tuple $x \in T_q$ can belong to the target vertex $v \in G$. Secondly, according to any clause "P_i of $\mathrm{ST}_i = P_i$ of IG," and Lemma 1, we have

$$\begin{aligned}
&\Pr\{v[S_1] = x[S_1]\} \leq h_1\\
&\cdot\\
&\cdot\\
&\Pr\{v[S_b] = x[S_b]\} \leq h_b.
\end{aligned} \tag{25}$$

Table 4: Result of the query E on Figures 11–13.

v	Q	P_1	P_2	S_1	S_2	F
8	27, M, 13000	RI	BC	B	C	2
8	27, M, 13000	RI	BC	B	J	4
8	27, M, 13000	RI	BC	B	M	2
8	27, M, 13000	RI	BC	B	T	2
8	27, M, 13000	RI	BC	F	C	1
8	27, M, 13000	RI	BC	F	J	2
8	27, M, 13000	RI	BC	F	M	1
8	27, M, 13000	RI	BC	F	T	1
8	27, M, 13000	RI	BC	P	C	3
8	27, M, 13000	RI	BC	P	J	6
8	27, M, 13000	RI	BC	P	M	3
8	27, M, 13000	RI	BC	P	T	3

On the contrary, since maximum values for each of these probabilities can be considered equal to h_i, and these probabilities are independent, we can say that

$$\begin{aligned}
&\Pr\{v[S_1] = x[S_1]. \ldots . v[S_b] = x[S_b]\}\\
&\qquad = \prod_{i=1}^{b} \Pr\{v[S_i] = x[S_i]\} \leq \prod_{i=1}^{b} h_i
\end{aligned} \tag{26}$$

and that Lemma 2 is correct. Note that the probability of estimating all the sensitive values of $v \in G$ is extremely less.

Using Table 4, we will describe Lemma 2. In accordance with Table 4, since $v[Q] =$ "27, M, 13000," we have

$$\begin{aligned}
&\Pr\{v[S_1] = \text{“}B\text{”}, v[S_2] = \text{“}C\text{”}\} = \frac{2}{30} = 0.07\\
&\leq h_1 \times h_2 = 0.5 \times 0.5 = 0.25\\
&\cdot\\
&\cdot\\
&\Pr\{v[S_1] = \text{“}P\text{”}, v[S_2] = \text{“}T\text{”}\} = \frac{3}{30} = 0.1\\
&\leq h_1 \times h_2 = 0.5 \times 0.5 = 0.25,
\end{aligned} \tag{27}$$

and then we can see that equation (10) is correct. In accordance with Lemmas 1 and 2, SNI always supplies PPVL. □

3.2.2. PPIL. In PPIL, if n vertices $v_1, \ldots, v_n \in$ IG (i.e., the vertices 4 and 5 in Figure 11) are with the same quasi-identifier, we do not consider each of them separately, since each of them can belong to the target individual $X \in G$ with a probability of $1/n$. Obviously, in this case, for all vertices $v_1, \ldots, v_n \in$ IG, one S_i has to be estimated, and for the target individual, $X \in G$ is considered.

SNI will supply PPIL, if we can reconstruct the sensitive values of any individual $X \in G$ when the maximum probability equals h_i. Here too, we consider the same two previous modes.

(1) First Mode (One ST_i). Obviously, to reconstruct an individual $X \in G$, firstly we must run the query F on IG and ST_i and then estimate $X[S_i]$. For example, Table 5 as T_q is the result of running the query F on Figures 11 and 12, where $v[Q] =$ "65, F, 25000." When there is one ST_i with an IG, the PPIL is described in Corollary 1 formally.

$$\Pr\{X[S_i] = x[S_i]\} = \sum_{k=1}^{n} \frac{1}{n} \Pr\{X[S_i] = x_k[S_i]\}, \tag{29}$$

Corollary 1. *If T_q is the result of running the query F on IG and one ST_i, then an adversary can deduce the sensitive label of a target individual $X \in G$ up to h_i. In other words, for $x \in T_q$ and $X \in G$,*

$$\Pr\{X[S_i] = x[S_i]\} \le h_i. \tag{28}$$

Proof. We consider two cases. In the first case, suppose n tuples $x_1, \ldots, x_n \in T_q$ (i.e., the tuples 1 and 4 in Table 5) are found whose partial sensitive labels and sensitive labels are the same. Obviously, in this case, for all tuples $x_1, \ldots, x_n \in T_q$, one $S_i$has to be estimated by $x[S_i]$ which is referred. However, we deduce the sensitive label of a target individual $X \in G$ in two steps. In the first step, since each of the tuples $x_1, \ldots, x_n \in T_q$can belong to the target individual $X \in G$ with a probability of $1/n$, we obtain the total probability for a target individual $X \in G$ as where $x[S_i] = x_1[S_i] = x_2[S_i] = \cdots = x_n[S_i]$. In the second step, using Lemma 1, for each tuple $x_k\,(1 \le k \le n)$, we obtain that $\Pr\{X[S_i] = x_k[S_i]\} \le h_i$, and since maximum values for each of these probabilities can be considered equal to h_i, we have

$$\sum_{k=1}^{n} \frac{1}{n} \Pr\{X[S_i] = x_k[S_i]\} \le \sum_{k=1}^{n} \frac{1}{n} h_i \le h_i, \tag{30}$$

and according to equation (13), Corollary 1 is correct.

Using Table 5, we will describe Corollary 1. In accordance with Table 5, for $h_1 = 0.5$, it can be said that

$$\begin{aligned}
&\Pr\{X[S_1] = \text{“}D\text{”}\} = \frac{2}{6} = 0.33 \le 0.5,\\
&\Pr\{X[S_1] = \text{“GU”}\} = \frac{2}{6} = 0.33 \le 0.5,\\
&\Pr\{X[S_1] = \text{“}G\text{”}\} = \frac{2}{6} = 0.33 \le 0.5.
\end{aligned} \tag{31}$$

In the second case, suppose n tuples $x_1, \ldots, x_n$ are found in T_q whose partial sensitive labels are the same (i.e., the tuples 1-4 or 5-7 in Table 6) but sensitive labels are not the same. Obviously, in this case, using Lemma 1, we obtain $\Pr\{X[S_i] = x_k[S_i]\} \le h_i$, and Corollary 1 is correct.

Table 5: Result of the query F on Figures 11 and 12.

v	Q	P_1	S_1	F
4	65, F, 25000	SD	D	1
4	65, F, 25000	SD	GU	1
4	65, F, 25000	SD	G	1
5	65, F, 25000	SD	D	1
5	65, F, 25000	SD	GU	1
5	65, F, 25000	SD	G	1

Using Table 6, which is the result of running the query F on Figures 11 and 13 with $v[Q] =$ "65, F, 25000," we will describe Corollary 1. In accordance with Table 6, for $h_2 = 0.5$, it can be said thatand then equation (11) is correct. But when there are multiple ST_i with an IG, PPIL can be described in Corollary 2 formally. □

$$\begin{aligned}
&\Pr\{X[S_1] = \text{“}C\text{”}\} = \frac{1}{5} = 0.2 \le 0.5,\\
&\Pr\{X[S_1] = \text{“}J\text{”}\} = \frac{2}{5} = 0.4 \le 0.5,\\
&\Pr\{X[S_2] = \text{“}M\text{”}\} = \frac{1}{5} = 0.2 \le 0.5,\\
&\Pr\{X[S_2] = \text{“}T\text{”}\} = \frac{1}{5} = 0.2 \le 0.5,\\
&\Pr\{X[S_2] = \text{“}A\text{”}\} = \frac{1}{4} = 0.25 \le 0.5,\\
&\Pr\{X[S_2] = \text{“}L\text{”}\} = \frac{2}{4} = 0.5 \le 0.5,\\
&\Pr\{X[S_2] = \text{“MA”}\} = \frac{1}{4} = 0.25 \le 0.5,
\end{aligned} \tag{32}$$

Corollary 2. *If T_q is the result of running the query E on IG and b ST_i, then an adversary can simultaneously deduce the values of b sensitive labels of any target individual $X \in G$ up to $\prod_{i=1}^{b} h_i$. In other words, for $x \in T_q$ and $X \in G$,*

$$\Pr\{X[S_1] = x[S_1]. \ldots . X[S_b] = x[S_b]\} \le \prod_{i=1}^{b} h_i. \tag{33}$$

Proof. We consider two samples. In the first sample, assume n tuples $x_1, \ldots, x_n \in T_q$ are discovered whose partial sensitive and sensitive labels are the same. Clearly, in this sample, for all tuples $x_1, \ldots, x_n \in T_q$, one $S_i$is cited to be computed with $x[S_i]$. However, we deduce b sensitive values of a target individual $X \in G$ in two phases. In the first phase, since each of the tuples $x_1, \ldots, x_n \in T_q$ can belong to the target individual $X \in G$ with a probability of $1/n$, we obtain the total probability for a target individual $X \in G$ as

Table 6: Result of the query F on Figures 11 and 13.

v	Q	P_2	S_2	F
4	65, F, 25000	BC	C	1
4	65, F, 25000	BC	J	2
4	65, F, 25000	BC	M	1
4	65, F, 25000	BC	T	1
5	65, F, 25000	WC	A	1
5	65, F, 25000	WC	L	2
5	65, F, 25000	WC	MA	1

$$\Pr\{X[S_1] = x[S_1]. \ldots . X[S_b] = x[S_b]\} = \sum_{k=1}^{n} \frac{1}{n} \Pr\{X[S_1] = x_k[S_1]. \ldots . X[S_b] = x_k[S_b]\}, \quad (34)$$

where $X[S_i] = x_1[S_i] = x_2[S_i] = \ldots = x_n[S_i]$. In the second phase, using Lemma 2, for each tuple $x_k (1 \le k \le n)$, we obtain that

$$\Pr\{X[S_1] = x_k[S_1]. \ldots . X[S_b] = x_k[S_b]\} \le \prod_{i=1}^{b} h_i, \quad (35)$$

and since the maximum value for these probabilities can be considered equal to $\prod_{i=1}^{b} h_i$, we have

$$\sum_{k=1}^{n} \frac{1}{n} \Pr\{X[S_1] = x_k[S_1]. \ldots . X[S_b] = x_k[S_b]\} \le \sum_{k=1}^{n} \frac{1}{n} \prod_{i=1}^{b} h_i = \prod_{i=1}^{b} h_i, \quad (36)$$

and Corollary 2 is correct.

In the second sample, assume n tuples $x_1, \ldots, x_n$ are discovered in T_q whose partial sensitive labels are the same (i.e., the tuples 1-12 or 13-21 in Table 7) but some of their sensitive labels are not the same. Obviously, in this case, using Lemma 2, we obtain

$$\Pr\{X[S_1] = x_k[S_1]. \ldots . X[S_b] = x_k[S_b]\} \le \prod_{i=1}^{b} h_i, \quad (37)$$

and Corollary 2 is correct. Using Table 7, we will explain Corollary 2. In accordance with Table 7, it can be said that

$$\begin{aligned} &\Pr\{X[S_1] = \text{“}D\text{”}, X[S_2] = \text{“}C\text{”}\} = \frac{1}{15} = 0.067 \\ &\le h_1 \times h_2 = 0.5 \times 0.5 = 0.25 \\ &. \\ &. \\ &\Pr\{X[S_1] = \text{“}G\text{”}, X[S_2] = \text{“MA”}\} = \frac{1}{12} = 0.083 \\ &\le h_1 \times h_2 = 0.5 \times 0.5 = 0.25. \end{aligned} \quad (38)$$

Table 7: Result of the query E on IG, ST_1, and ST_2.

v	Q	P_1	P_2	S_1	S_2	F
4	65, F, 25000	SD	BC	D	C	1
4	65, F, 25000	SD	BC	D	J	2
4	65, F, 25000	SD	BC	D	M	1
4	65, F, 25000	SD	BC	D	T	1
4	65, F, 25000	SD	BC	GU	C	1
4	65, F, 25000	SD	BC	GU	J	2
4	65, F, 25000	SD	BC	GU	M	1
4	65, F, 25000	SD	BC	GU	T	1
4	65, F, 25000	SD	BC	G	C	1
4	65, F, 25000	SD	BC	G	J	2
4	65, F, 25000	SD	BC	G	M	1
4	65, F, 25000	SD	BC	G	T	1
5	65, F, 25000	SD	WC	D	A	1
5	65, F, 25000	SD	WC	D	L	2
5	65, F, 25000	SD	WC	D	MA	1
5	65, F, 25000	SD	WC	GU	A	1
5	65, F, 25000	SD	WC	GU	L	2
5	65, F, 25000	SD	WC	GU	MA	1
5	65, F, 25000	SD	WC	G	A	1
5	65, F, 25000	SD	WC	G	L	2
5	65, F, 25000	SD	WC	G	MA	1

However, in accordance with Corollaries 1 and 2, SNI always supplies PPIL because an adversary can reconstruct a sensitive value of an individual up to h_i and all sensitive values of him up to $\prod_{i=1}^{b} h_i$. □

3.3. Utility Requirements. The correlation among the published graph labels is always significant for analysts. In this section, we want to show that SNI almost preserves the correlation among sensitive labels, by preserving the correlation among partial sensitive labels. In SNI, each of STs and IG can separately be used by analysts, since each partial sensitive label results from the generalization of a sensitive label which can solve the problem of scattered data for analysts. We study the membership analysis for two models: 1, one sensitive label and 2, multiple sensitive labels.

3.3.1. One Sensitive Label. Assume we want to study categorization (classification) of the job titles. In accordance with Figure 13, the taxonomic tree of the job label has 7 leaves and 5 internal nodes. Table 8 has 12 queries related to the nodes of that taxonomic tree. In fact, each of these queries extracts the vertices having special conditions.

Suppose we want to compute each of the queries Q_1–Q_{12} in Table 8 by objects which are derived from Anatomy or SNI. In accordance with Figure 13, the clauses in queries Q_{11}-Q_{12}, like “WC,” are on Cut_2 and those in queries Q_1–Q_{10}, like “PR,” are below Cut_2. Hence, we will study two modes for the clauses in these queries: 1, below cut and 2, above cut.

(1) First Mode (Below Cut). Assume, by query Q_{10}, we would like to find out the individuals who are in the category of “PR.” If G (Figure 1) is available, according to Figure 13, the result of running the query Q_{10} is three valid vertices 2, 5,

and 6 in G. If Figures 11 and 13 are available, then the easiest way is to run the query H instead of the query Q_{10}.

H:

Select distinct vertex
From Figures 11 and 13
Where partial job of Figure 11 = partial job of Figure 13
And job in ("L," "A")

The result of running the query H is shown in Figure 14 that has 3 valid vertices and 1 invalid vertex. In accordance with equations (1) and (2), the MA and ME for the query H are 0.75 and 0.25 (MA = 3/4 and ME = 1/4). But if Figure 4 and Table 1 are available, then the easiest way is to run the query I.

I:

Select distinct vertex
From Table 1 and Figure 4
Where Group-ID2 of Table 1 = Group-ID2 of Figure 4
And job in ("L," "A")

Figure 15 shows the result of the query I that has 3 valid vertices and 6 invalid vertices (MA = 3/9 = 0.33 and ME = 6/9 = 0.67).

(2) Second Mode (Above Cut). Assume, by query Q_{12}, we would like to extract those who are in the category of "BC." The result of running Q_{12} by Figure 1 is shown in Figure 16 which has 5 valid vertices 1, 3, 4, 8, and 9. If Figure 11 is available, then we can run the query J instead of the query Q_{12}.

J:

Select distinct vertex
From Figure 11
Where partial job in ("BC")

Obviously, the result of running the query J is shown in Figure 16 which has just 5 valid vertices 1, 3, 4, 8, and 9 because Figure 11 contains the complete information required by the query J. Interestingly, the MA and ME for the query J are 1 and 0 (MA = 5/5 and ME = 0/5). Note that, by query J, the vertices can be reconstructed only with partial sensitive values, not with original sensitive values.

But if Figure 4 and Table 1 are available, then we can run the query K instead of the query Q_{12}.

K:

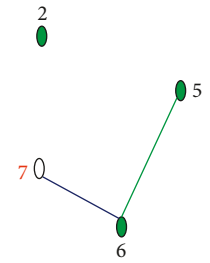
Select distinct vertex
From Figure 4 and Table 1
Where Group-ID2 of Figure 4 = Group-ID2 of Table 1
and
Job in ("J," "M," "T," "C")

Figure 17 shows the result of the query K (MA = 5/9 = 0.55 and ME = 4/9 = 0.44). Note that the accuracy of membership analyses is extremely decreased, by Anatomy.

Figure 18 shows the MA and ME of the queries Q_1–Q_{12} by Anatomy and SNI. In accordance with Figure 18, it is observed that SNI shows higher performance than Anatomy,

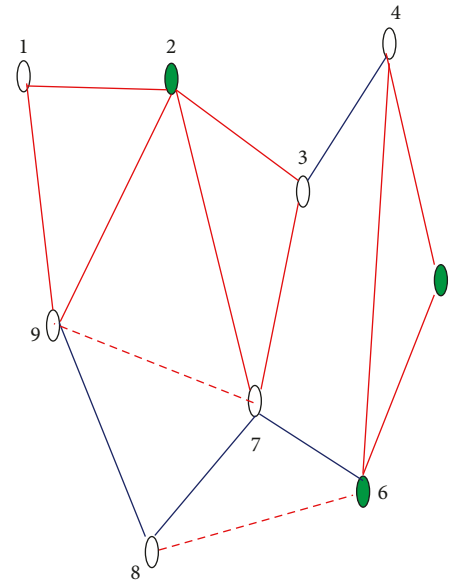
Table 8: Queries Q_1–Q_{12} related to the taxonomic tree of job.

Q	Select	From	Where
Q1	Distinct vertex	G	Job in {"C"}
Q2	Distinct vertex	G	Job in {"L"}
Q3	Distinct vertex	G	Job in {"M"}
Q4	Distinct vertex	G	Job in {"J"}
Q5	Distinct vertex	G	Job in {"MA"}
Q6	Distinct vertex	G	Job in {"T"}
Q7	Distinct vertex	G	Job in {"A"}
Q8	Distinct vertex	G	Job in {"TE"}
Q9	Distinct vertex	G	Job in {"NT"}
Q10	Distinct vertex	G	Job in {"PR"}
Q11	Distinct vertex	G	Job in {"WC"}
Q12	Distinct vertex	G	Job in {"BC"}



Valid vertices = {2, 5, 6}
Invalid vertex = {7}

Figure 14: Result of the query H.



Valid vertices = {2, 5, 6}
Invalid vertices = {1, 3, 4, 7, 8, 9}

Figure 15: Result of the query I.

and from the view point of categorization, it can even produce the complete instances in some cases (i.e., Q11-Q12).

In accordance with Figure 18, for one sensitive label, it is observed that the ME criterion in SNI is less. In the next

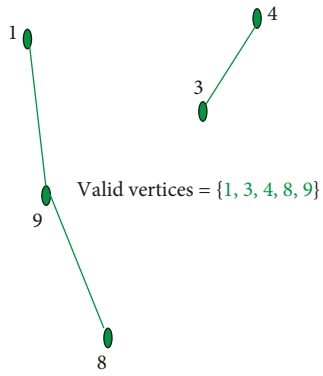


Figure 16: Result of the query *J*.

section, the MA and ME for graphs having several sensitive labels with correlation are calculated, which is interesting.

3.3.2. Multiple Sensitive Labels. Suppose that we want to apply each of the queries Q_{13}–Q_{22} in Table 9 by Anatomy and SNI. Note that each of these queries applies two sensitive labels (disease and job). In accordance with Figures 12 and 13, the clauses in the queries Q_{17}–Q_{19} and Q_{22}, like "WC" and "RI," are on Cut_1 and Cut_2, and the clauses in the queries Q_{13}–Q_{16} and Q_{20}–Q_{21}, like "PR," are below Cut_2. We will study two modes for the clauses in these queries: 1, below cut and 2, above cut.

(1) First Mode (Below Cut). Assume, by query Q_{15}, we would like to realize the individuals who are in the categories of the "PR" and "RI." If Figure 1 is available, according to Figures 12 and 13, the result of running Q_{15} is two vertices 2 and 6 in Figure 1 (original graph). If Figures 11 and 13 are available, then the easiest way is to run the query *L*.

L:

Select distinct vertex
From Figures 11 and 13
Where partial job of Figure 11 = partial job of Figure 13 and
Partial disease in ("RI") and
Job in ("L," "A")

The result of running the query *L* is shown in Figure 19 which has 2 valid vertices and 1 invalid vertex (MA = 2/3 = 0.67 and ME = 1/3 = 0.33). But, if Figure 4 and Tables 1 and 3 are available, then we must run the query *M*.

M:

Select distinct vertex
From Figure 4 and Tables 1 and 3
Where Group-ID2 of Figure 4 = Group-ID2 of Table 1 and
Group-ID of Figure 4 = Group-ID of Table 3 and
Job in ("L," "A") and disease in ("F," "P," "B")

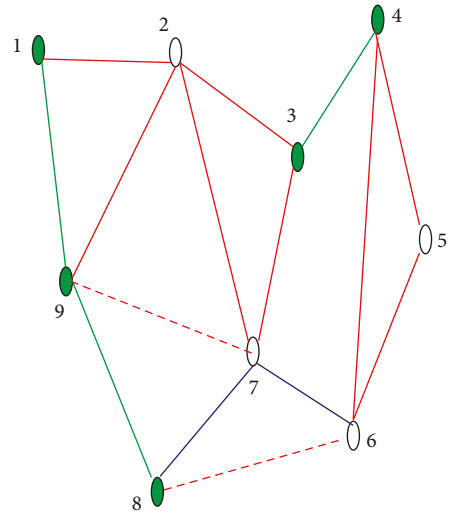


Figure 17: Result of the query *K*.

The result of running the query *M* is shown in Figure 20 which has 2 valid vertices and 7 invalid vertices (MA = 2/9 = 0.22 and ME = 7/9 = 0.78).

(2) Second Mode (Above Cut). Assume, by query Q_{18}, we would like to discover those who are in the categories of the "BC" and "RI." According to Figures 1, 12, and 13, the result of running the query Q_{18} is three valid vertices 1, 8, and 9 in Figure 21. If Figure 11 is available, then the easiest way is to run the query *N*.

N:

Select distinct vertex
From Figure 11
Partial job in ("BC") and
Partial disease in ("RI")

Figure 21 presents the output of the query *N* which has three valid vertices 1, 8, and 9 because Figure 11 contains the complete information required by the query *N*. Note that the MA and ME for the query *N* are 1 and 0 (MA = 3/3 and ME = 0/3). If Figure 4 and Tables 1 and 3 are available, the easiest way is to run the query *O*.

O:

Select distinct vertex
From Figure 4 and Tables 1 and 3
Where Group-ID2 of Figure 4 = Group-ID2 of Table 1 and
Group-ID of Figure 4 = Group-ID of Table 3 and
Job in ("J," "M," "T," "C") and
Disease in ("F," "P," "B")

Figure 22 states the output of the query *O* which has 6 invalid vertices and 3 valid vertices (MA = 3/9 = 0.33 and ME = 6/9 = 0.67).

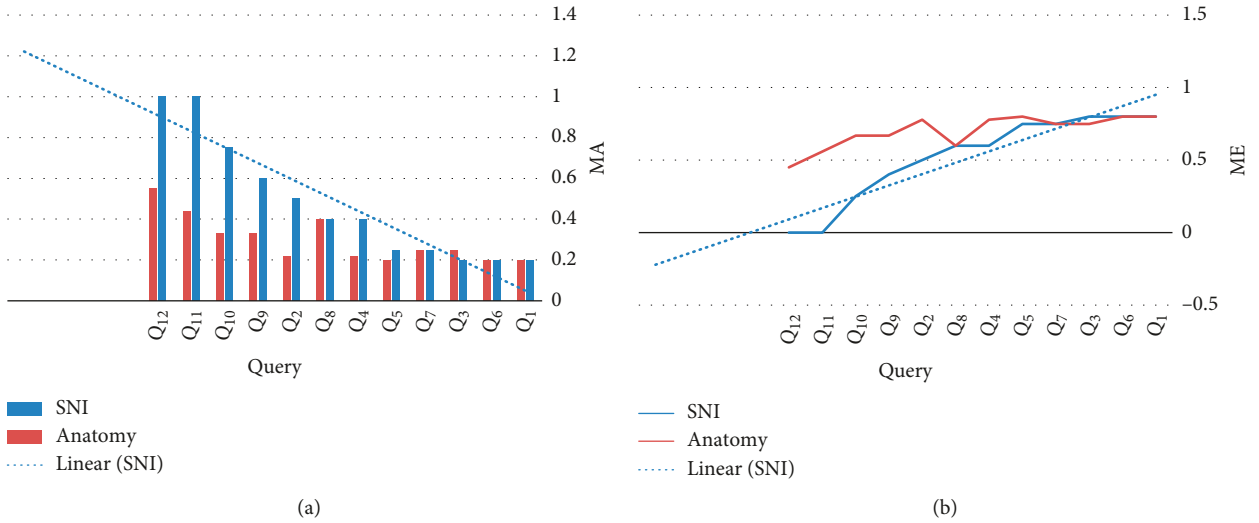


Figure 18: MA (a) and ME (b) for queries Q_1–Q_{12}.

Table 9: Queries Q_{13}–Q_{22} related to taxonomic trees of disease and job.

Query	Select	From	Where
Q_{13}	Distinct vertex	G	Job in {nontechnical} and disease in {stomach disease}
Q_{14}	Distinct vertex	G	Job in {technical} and disease in {respiratory infection}
Q_{15}	Distinct vertex	G	Job in {professional} and disease in {respiratory infection}
Q_{16}	Distinct vertex	G	Job in {professional} and disease in {stomach disease}
Q_{17}	Distinct vertex	G	Job in {white-collar} and disease in {stomach disease}
Q_{18}	Distinct vertex	G	Job in {blue-collar} and disease in {respiratory infection}
Q_{19}	Distinct vertex	G	Job in {blue-collar} and disease in {stomach disease}
Q_{20}	Distinct vertex	G	Job in {technical} and disease in {stomach disease}
Q_{21}	Distinct vertex	G	Job in {nontechnical} and disease in {respiratory infection}
Q_{22}	Distinct vertex	G	Job in {white-collar} and disease in {respiratory infection}

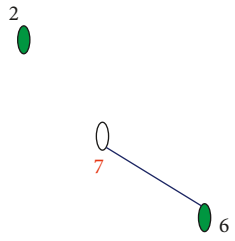
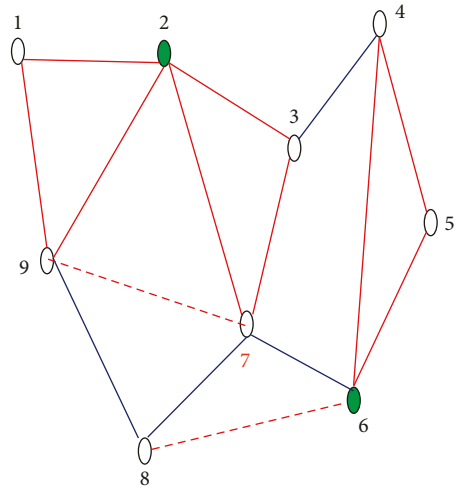


Figure 19: Result of the query L.

Figure 23 shows the MA and ME for all queries Q_{13}–Q_{22} by Anatomy and SNI. Note that, according to Figure 23, when the values of the clause in a query are above Cut_1 and Cut_2, MA = 1 and ME = 0. In other words, the resultant graph only contains valid vertices.

By comparing Figures 18 and 23, it can be seen that when the clause in a query contains multiple sensitive labels, the MA in immunization is extremely more than that in Anatomy.



Valid vertices = {2, 6}
Invalid vertices = {1, 3, 4, 5, 7, 8, 9}

Figure 20: Result of the query M.

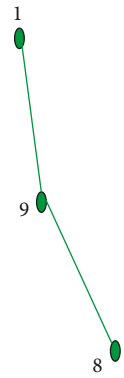


Figure 21: Output of the query N.

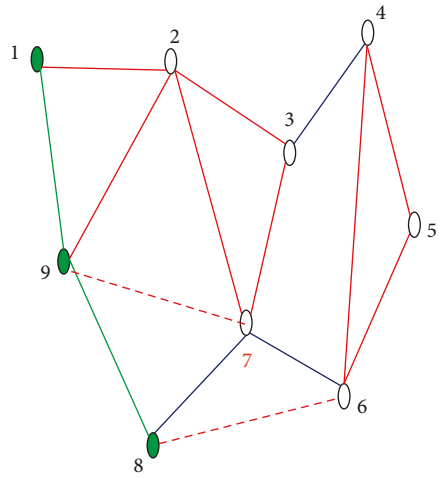
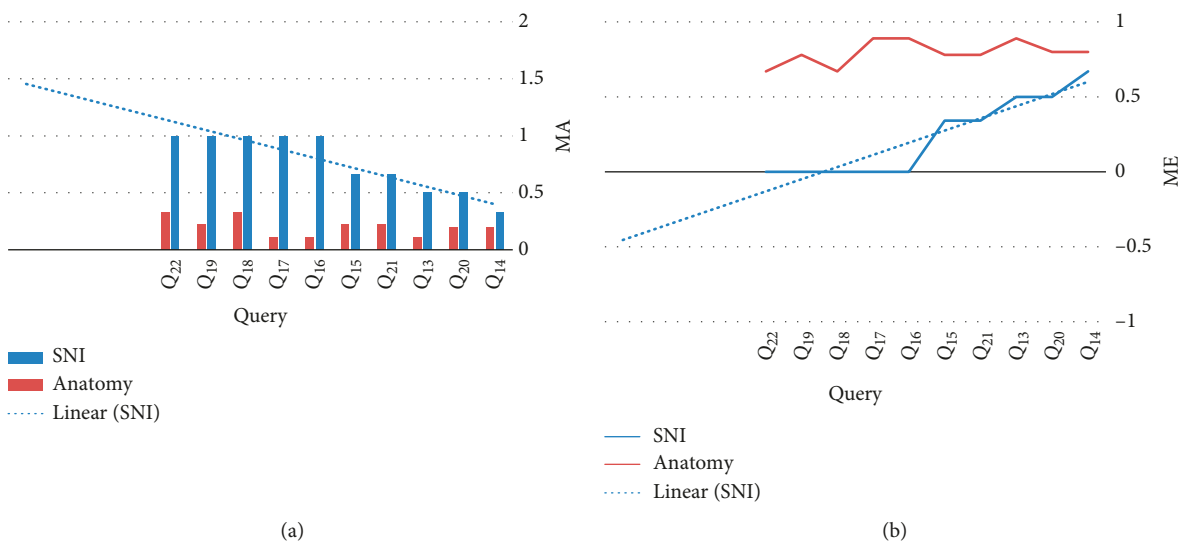


Figure 22: Result of the query O.



(a) (b)

Figure 23: MA (a) and ME (b) for queries Q_{12}–Q_{22}.

Input: *an original graph (G) and n taxonomic trees (TAX_1,…, TAX_n) and attack thresholds (h_1,…, h_n)*

Output: *an immune graph (IG) and n supplementary trees ST_1…..ST_n)*

Module:

Procedure 1: *to solve VLC and LLC for labels on edges, extract an anonymized graph, by conventional algorithms which do not edit sensitive labels on vertices*

Procedure 2: *to solve LLC for labels on vertices and weaken LCC, extract an IG and n ST_i by lines 1–16*

*/*The lines 1-3 initialize the frequency field of all leaves in taxonomic trees*/*

1. *For each leaf $x \in TAX_i$ do*
2. $x[F_i] \longleftarrow$ *frequency of $x[S_i]$ in all vertices of G;*
3. *End for*

*/*Extraction activity (lines 4-12) extracts n ST_i*/*

*/*the lines 4-7 calculate the internal nodes*/*

4. *For each internal node $y \in TAX_i$ do*
5. $y[F_i] \longleftarrow \Sigma_{All\ Leaves\ x\ in\ subtree\ y}\ x[F_i]$;
6. $y[R_i] \longleftarrow \dfrac{\mathrm{Max}(all\ x[Fi]\ in\ subtree\ of\ y)}{y[F_i]}$
7. *End for*

*/*the lines 8-11 extract the Cut_i*/*

8. *For each TAX_i, consider m internal nodes $y_k \in TAX_i$ as a Cut_i, If them satisfy following two conditions. (for $1 \le k, j \le m$)*
9. $\cup_{k=1}^{m}$ *{Leaves in subtree of y_k} = {all leaves of TAX_i}*
10. *{Leaves in subtree of y_k} ∩ {Leaves in subtree of y_j} = ø*
11. *End for*

*/*the line 12 extract n ST_i*/*

12. *Rename n taxonomy trees (TAX_1,…,TAX_n) to Trees (ST_1…..ST_n)*

*/*Reduction activity (lines 13-16) reduces sensitivity of the sensitive labels on vertices of G and extracts an Immune Graph (IG)*/*

13. *For each $v[S_i]$ do // vertex $v \in G$*
14. *find a leaf $x \in ST_i$ where $v[S_i] = x[S_i]$ and $v[S_i] \longleftarrow S_i$ of ancestor of $x \in Cut_i$*
15. *End for*
16. *Rename G to IG*
17. ***Return IG*** *and all* $\boldsymbol{ST_i}$.

Figure 24: SNI algorithm.

Also, the ME in immunization is extremely less than that in Anatomy. In fact, immunization increases the accuracy of membership analysis. The reason is that there is a correlation among the quasi-identifier and partial sensitive labels. As a result, multiple sensitive labels with correlation are a simple scenario in which the data publisher can publish multiple trees (one for each sensitive label) by using the SNI model.

4. SNI Algorithm

Figure 24 illustrates the SNI algorithm. The algorithm operates according to the framework in Figure 10. Using an original graph and n taxonomic trees (TAX_1, . . ., TAX_n) and attack thresholds (h_1, . . ., h_n) as the input, it produces an immune graph (IG) and n supplementary trees (ST_1, . . ., ST_n) as the output. In accordance with Figure 24, the algorithm contains 2 procedures. The second procedure which is the main focus of this article has three activities: 1, initialization (lines 1–3); 2, extraction (lines 4–12); and 3, reduction (lines 13–16).

The first procedure applies a conventional algorithm which does not edit the sensitive labels on vertices of G and forms an anonymized graph. For example, if Figure 1 is the input of this procedure, then its output will be Figure 3 which solves VLC and LLC for labels on edges.

In initialization activity, in accordance with the sensitive labels on any vertex $v \in G$, $x[F_i]$ of any leaf $x \in TAX_i$ is updated. For example, according to Figure 1, the amount of F_i for the leaf "L" in Figure 8 is set to 2 since the job title of two vertices in Figure 1 equals "L."

In extraction activity, first, $y[F_i]$ of any internal node $y \in TAX_i$ is set by the sum of $x[F_i]$ of all leaves in the subtree of y (line 5). Second, $y[R_i]$ is set by the maximum of $x[R_i]$ of all leaves x in the subtree of y (line 6), or by the maximum of the relative frequencies of all leaves in the subtree of y. Finally for each TAX_i, a set of nodes named "Cut_i" is extracted (lines 8–11). Finally, TAX_1, . . ., TAX_n are renamed to ST_1, . . ., ST_n, respectively.

In reduction activity, the sensitivity of the sensitive labels on any vertices $v \in G$ is decreased and a graph named "IG" is produced for publishing (lines 13–16). First, for any vertex $v \in G$, a leaf $x \in ST_i$ is found where $x[S_i] = v[S_i]$, and then the field P_i of the ancestor of the leaf x in Cut_i is copied to the field S_i of the vertex v (Line 14). Finally, G is renamed to IG (line 16).

5. Experiments

In this section, an empirical experiment examines the degree of utility of data derived from SNI and Anatomy. For necessary experiments, a real dataset of cancer patients in an Iranian hospital that contains the cancer detection code of patients was used. The sensitive label in this dataset is the cancer detection code. Here, only results related to the cancer code label were presented. The ICD-O (International Classification of Diseases for Oncology) [38] was used to construct the taxonomic tree of cancer code. Figure 25 shows the part of this taxonomic tree. This taxonomic tree has 848 nodes, 98 of which are internal nodes. For example, according to Figure 25, the internal node C00 is a parent node for leaves C00.0–C00.9. C00.0 is the cancer code for the external upper lip (NOS, lipstick area, and vermilion border).

Table 10 shows the main internal nodes that were used in queries on the dataset. Table 11 contains the main queries that were used in tests. The degree of utility of data derived from SNI and Anatomy was calculated by the MA and ME (equations (1) and (2)). Figure 26 illustrates the results of all queries A′–O′ in Table 11 with $h_1 = 0.34$ for SNI and $l = 127$ for SNI and Anatomy.

In accordance with Figure 26, in SNI, when the query condition, like queries M′–O′, is close to Cut_{Cancer} (permitted generalization limit), the membership accuracy (MA)

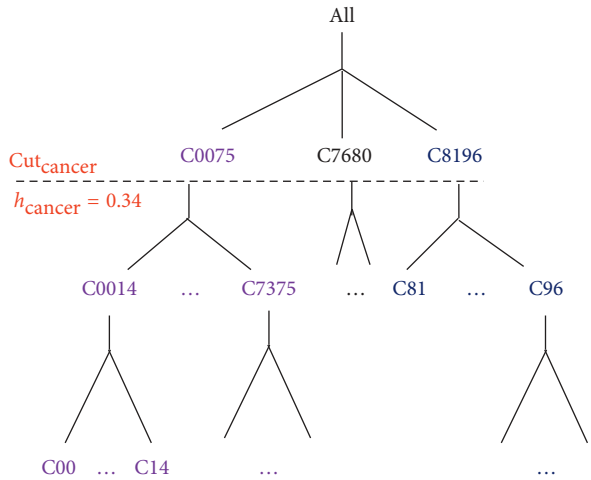


Figure 25: Taxonomic tree of cancer code.

Table 10: Main internal nodes in the taxonomic tree of cancer code.

Category	Node
Lip, oral cavity, and pharynx	C0014
Digestive organs	C1526
Respiratory and intrathoracic organs	C3039
Bone and articular cartilage	C4041
Skin	C4344
Mesothelial and soft tissue	C4549
Breast	C5050
Female genital organs	C5158
Male genital organs	C6063
Urinary tract	C6468
Eye, brain, and other parts of the central nervous system	C6972
Thyroid and other endocrine glands	C7375
Malignant neoplasms, stated or presumed to be primary, of specified sites, except for lymphoid, hematopoietic, and related tissue	C0075
Malignant neoplasms of ill-defined, secondary, and unspecified sites	C7680
Malignant neoplasms, stated or presumed to be primary, of lymphoid, hematopoietic, and related tissue	C8196

Table 11: Queries related to the main nodes in the taxonomic tree of cancer code.

Query	Select	From	Where
A′	Distinct vertex	Dataset	Cancer in {**C0014**}
B′	Distinct vertex	Dataset	Cancer in {**C1526**}
C′	Distinct vertex	Dataset	Cancer in {**C3039**}
D′	Distinct vertex	Dataset	Cancer in {**C4041**}
E′	Distinct vertex	Dataset	Cancer in {**C4344**}
F′	Distinct vertex	Dataset	Cancer in {**C4549**}
G′	Distinct vertex	Dataset	Cancer in {**C5050**}
H′	Distinct vertex	Dataset	Cancer in {**C5158**}
I′	Distinct vertex	Dataset	Cancer in {**C6063**}
J′	Distinct vertex	Dataset	Cancer in {**C6468**}
K′	Distinct vertex	Dataset	Cancer in {**C6972**}
L′	Distinct vertex	Dataset	Cancer in {**C7375**}
M′	Distinct vertex	Dataset	Cancer in {**C0075**}
N′	Distinct vertex	Dataset	Cancer in {**C7680**}
O′	Distinct vertex	Dataset	Cancer in {**C8196**}

increases up to 1. In other words, the resultant graph only contains valid vertices, and in fact, it does not contain any invalid vertices. This suggests a correlation of quasi-identifier and sensitive labels since there is a correlation among the quasi-identifier and partial sensitive labels in the immunized graph and supplementary trees.

Obviously, SNI depends highly on the structure of taxonomic trees and the frequency of data. Generally, deeper or more balanced taxonomic trees cause better results. The interesting point to be found in the experiments is that, for the data that need to be published using the SNI technique, a characteristic called the maturity time should be defined. In other words, the data are not allowed to be published until maturity time is reached. In fact, if the data holder realizes that the data frequency is not suitable for the publication of the data, he/she should wait until more data are collected or can use and add up previous years' data to supplement maturity time needs.

5.1. Details of Computational Effort. In accordance with the SNI algorithm in Figure 24, the frequency field of leaves of all taxonomic trees is initialized through sensitive label values in the graph, by once traversing the graph (lines 1-3 of the algorithm); all necessary calculations to extracting immunized supplementary trees are only done on taxonomic trees (lines 4-11 of the algorithm); and any taxonomic tree has finite and tolerable nodes. Obviously, the SNI algorithm is feasible and efficient for very large graphs having millions of nodes because the time of traversing a taxonomic tree that has the frequencies of the nodes of a very large dataset is exactly equal to the time of traversing the same taxonomic tree that has the frequencies of the nodes of a small dataset. In fact, the difference between two datasets—very large dataset and small dataset—is the mapping time of their nodes on a taxonomic tree that we

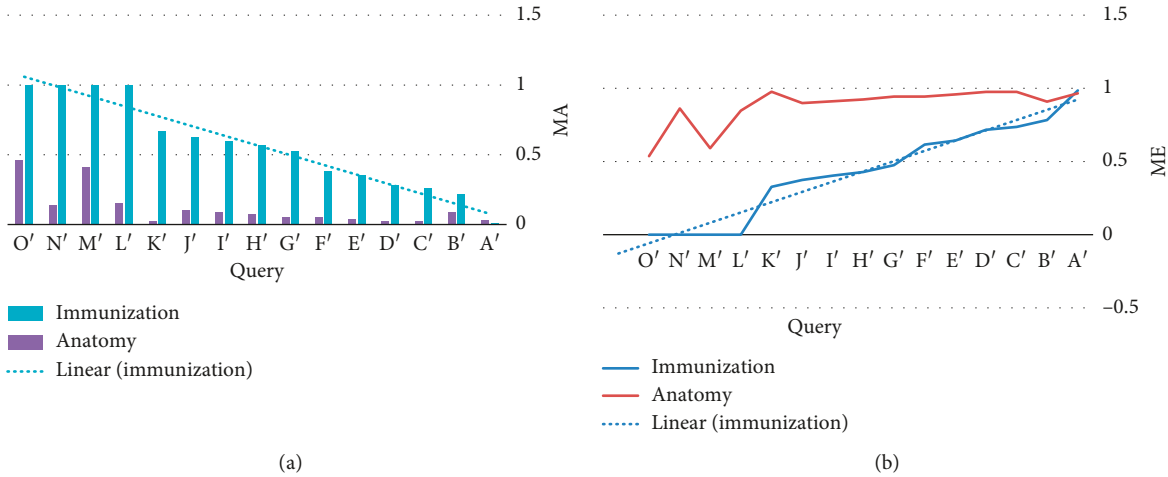


Figure 26: MA (a) and ME (b) of queries A′–O′ of cancer.

Table 12: Implementation of the taxonomy tree (cancer code).

ID	Type	Detection code	Detection code of the parent	Parent ID	Frequency	Attack risk
1	Leaf	C00.0	C00	10	0	0
2	Leaf	C00.1	C00	10	0	0
.	.	.	.	.	.	.
9	Leaf	C00.9	C00	10	3	1
10	Internal	C00	C0014	56	3	1
11	Internal	C01	C0014	56	0	0
12	Leaf	C02.0	C02	20	0	0
13	Leaf	C02.1	C02	20	0	0
14	Leaf	C02.2	C02	20	1	1
.	.	.	.	.	.	.

can use the best existing mapping algorithms. That is, the algorithm receives an original graph and multiple taxonomic trees as the input (i.e., Figures 3, 5, and 8) and just updates the frequency field of leaves in taxonomic trees (i.e., Figures 5 and 8). This algorithm computes immunized supplementary trees in $O(n/b)$ I/Os, where n is the cardinality of the dataset and b the page size. Note that if n is very large in practice (e.g., at the order of a million), then our algorithm is nearly optimal.

5.2. Implementation of Taxonomy Tree. For the experiments, the values of the sensitive labels on vertices are transferred to a table. Any taxonomic tree is transferred to a table that has 7 fields. For example, Table 12 illustrates how to implement the taxonomy tree shown in Figure 25. The ID attribute is used to refer to a record or to link between records. The type attribute is used to distinguish between the internal node and leaf node. The parent ID attribute is actually used to simulate the edge between the parent node and the child node. The reason for using only the parent ID attribute is that movement on the tree is only from bottom to top.

6. Conclusion

The SNI technique preserves correlation among multiple sensitive labels and almost solves LCC for membership analysis. It is appropriate for a graph that has multiple sensitive labels, and each sensitive label has a taxonomic tree and an attack threshold. It produces one IG and multiple STs to publish the original data. These IG and STs considerably supply privacy-preserving service and utility for membership analyses. Unlike Anatomy, in SNI, the sensitive values are not randomly grouped but are grouped based on the attack threshold and taxonomic tree. This technique uses the partial sensitive label to maintain the correlation among labels. This new technique can also help the researchers to get exact compact data about unknown original data as well as more detailed data such as partial sensitive values and real quasi-identifiers. It definitely increases the utility of the published data for membership analysis. It also makes way for future researchers. The output of the SNI method not only depends on the frequency of sensitive values but also depends on the structure of taxonomic trees.

Data Availability

The data analyzed during this study are described in the following metadata record: https://drive.google.com/file/d/1sovHfcitNX95CdER3JJCnv_6MeZJEfxZ/view?usp=sharing.

Disclosure

This work was performed at Najafabad Branch, Islamic Azad University, Najafabad, Iran.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] T. Dalenius, "Finding a needle in a haystack or identifying anonymous census records," *Journal of Official Statistics*, vol. 2, no. 3, pp. 329–336, 1986.

[2] P. Samarati and L. Sweeney, *Generalizing Data to Provide Anonymity when Disclosing Information*, PODS, Clearwater, FL, USA, 1998.

[3] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan, "Group formation in large social networks: membership, growth, and evolution," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06)*, Philadelphia, PA, USA, August 2006.

[4] B. Zhou and J. Pei, "The *k*-anonymity and *l*-diversity approaches for privacy preservation in social networks against neighborhood attacks," *Knowledge and Information Systems*, vol. 28, no. 1, pp. 47–77, 2011.

[5] S. H. Erfani and R. Mortazavi, "A novel graph-modification technique for user privacy-preserving on social networks," *Journal of Telecommunications and Information Technology*, vol. 3, pp. 27–38, 2019.

[6] X. Li, J. Yang, Z. Sun, and J. Zhang, "Differential privacy for edge weights in social networks," *Security and Communication Networks*, vol. 2017, Article ID 4267921, 10 pages, 2017.

[7] X. Xiao and Y. Tao, "Anatomy: simple and effective privacy preservation," in *Proceedings of the 32nd International Conference on Very Large Databases (VLDB)*, Seoul, Korea, September 2006.

[8] K. Baktha and B. K. Tripathy, "Alpha anonymization in social networks using the Lossy-join approach," *Transactions on Data Privacy*, vol. 11, no. 1, pp. 1–22, 2018.

[9] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "L-diversity," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, 2007.

[10] M. Yuan, L. Chen, P. S. Yu, and T. Yu, "Protecting sensitive labels in social network data anonymization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 3, pp. 633–647, 2013.

[11] S. Chakraborty and B. K. Tripathy, "Alpha-anonymization techniques for privacy preservation in social networks," *Social Network Analysis and Mining*, vol. 6, no. 1, 2016.

[12] S. Chakraborty and B. K. Tripathy, "Privacy preservation in social networks through alpha," in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015 (ASONAM '15)*, Paris, France, April 2015.

[13] M. Yuan, L. Chen, and P. S. Yu, "Personalized privacy protection in social networks," *Proceedings of the VLDB Endowment*, vol. 4, no. 2, pp. 141–150, 2010.

[14] Y. Song, P. Karras, Q. Xiao, and S. Bressan, "Sensitive label privacy protection on social network data," in *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 562–571, Springer Science+Business Media, Berlin, Germany, 2012.

[15] L. Liu, J. Wang, J. Liu, and J. Zhang, "Privacy preservation in social networks with sensitive edge weights," in *Proceedings of the 2009 SIAM International Conference on Data Mining*, Philadelphia, PA, USA, April 2013.

[16] P. Liu, Y. Bai, L. Wang, and X. Li, "Partial *k*-anonymity for privacy-preserving social network data publishing," *International Journal of Software Engineering and Knowledge Engineering*, vol. 18, 2016.

[17] M. Li, Z. Liu, and K. Dong, "Privacy preservation in social network against public neighborhood attacks," in *Proceedings of the 15th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, 10th IEEE International Conference on Big Data Science and Engineering and 14th IEEE International Symposium on Parallel and Distributed System*, Helsinki, Finland, August 2016.

[18] P. Samarati and L. Sweeney, "Generalizing data to provide anonymity when disclosing information (abstract)," in *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS '98)*, Seattle, Washington, USA, June 1998.

[19] L. Sweeney, "*k*-anonymity: a model for protecting privacy," *International Journal of Uncertainty, Fuzziness Knowledge-Based System*, vol. 10, no. 5, pp. 557–570, 2002.

[20] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "ℓ-Diversity: privacy beyond *k*-anonymity," in *Proceedings of the International Conference on Data Engineering*, Atlanta, GA, USA, April 2006.

[21] X. Xiao and Y. Tao, "Personalized privacy preservation," in *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data (SIGMOD '06)*, Chicago, IL, USA, June 2006.

[22] Q. Liu, H. Shen, and Y. Sang, "Privacy-preserving data publishing for multiple numerical sensitive attributes," *Tsinghua Science and Technology*, vol. 20, no. 3, pp. 246–254, 2015.

[23] J. Han, F. Luo, J. Lu, and H. Peng, "SLOMS: a privacy preserving data publishing method for multiple sensitive attributes microdata," *Journal of Software*, vol. 8, no. 12, 2013.

[24] V. S. Susan and T. Christopher, "Anatomisation with slicing: a new privacy preservation approach for multiple sensitive attributes," *Springerplus*, vol. 5, no. 1, 2016.

[25] S. Ram Prasad Reddy, K. VSVN Raju, and V. Valli Kumari, "A novel approach for personalized privacy preserving data publishing with multiple sensitive attributes," *International Journal of Engineering & Technology*, vol. 7, p. 197, 2018.

[26] A. Anjum, N. Farooq, S. U. R. Malik, A. Khan, M. Ahmed, and M. Gohar, "An effective privacy preserving mechanism for 1: M microdata with high utility," *Sustainable Cities and Society*, vol. 45, p. 213, 2019.

[27] T. Kanwal, S. A. A. Shaukat, A. Anjum et al., "Privacy-preserving model and generalization correlation attacks for 1:M data with multiple sensitive attributes," *Information Sciences*, vol. 488, pp. 238–256, 2019.

[28] C. Sun, P. S. Yu, X. Kong, and Y. Fu, "Privacy preserving social network publication against mutual friend attacks," *Transactions on Data Privacy*, vol. 7, pp. 71–97, 2014.

[29] K. Wang and B. C. M. Fung, "Anonymizing sequential releases," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '06)*, Philadelphia, PA, USA, August 2006.

[30] R. C. Wong, J. Li, A. W. Fu, and K. Wang, "(α, K)-anonymity: an enhanced *K*-anonymity model for privacy preserving data publishing," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Philadelphia, PA, USA, August 2006.

[31] K. Wang, B. C. M. Fung, and P. S. Yu, "Handicapping attacker's confidence: an alternative to κ-Anonymization," *Knowledge and Information Systems*, vol. 11, no. 3, pp. 345–368, 2007.

[32] R. J. Bayardo and R. Agrawal, "Data privacy through optimal *k*-anonymization," in *Proceedings of the 21st International Conference on Data Engineering*, Tokyo, Japan, April 2005.

[33] B. C. M. Fung, K. Wang, and P. S. Yu, "Top-down specialization for information and privacy preservation," in *Proceedings of the International Conference on Data Engineering*, Tokyo, Japan, April 2005.

[34] A. Hasan, Q. Jiang, H. Chen, and S. Wang, "A new approach to privacy-preserving multiple independent data publishing," *Applied Sciences*, vol. 8, no. 5, p. 783, 2018.

[35] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Incognito: efficient full-domain k-anonymity," in *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, Baltimore, MD, USA, June 2005.

[36] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. W.-C. Fu, "Utility-based anonymization using local recoding," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '06)*, Philadelphia, PA, USA, August 2006.

[37] A. Inan, M. Kantarcioglu, and E. Bertino, "Using anonymized data for classification," in *Proceedings of the International Conference on Data Engineering*, Shanghai, China, March 2009.

[38] J. C. Booth, "International classification of diseases for oncology (ICD-0)," *Pathology*, vol. 10, no. 2, pp. 202-203, 2016.