International Journal of Data Warehousing and Mining
Volume 15 • Issue 3 • July-September 2019

A Method of Sanitizing Privacy-Sensitive Sequence Pattern Networks Mined From Trajectories Released

Haitao Zhang, Nanjing University of Posts and Telecommunications, Nanjing, China
Yunhong Zhu, Nanjing University of Posts and Telecommunications, Nanjing, China

ABSTRACT

Mobility patterns mined from released trajectories can help to allocate resources and provide personalized services, although these also pose a threat to personal location privacy. As the existing sanitization methods cannot deal with the problems of location privacy inference attacks based on privacy-sensitive sequence pattern networks, the authors proposed a method of sanitizing the privacy-sensitive sequence pattern networks mined from trajectories released by identifying and removing influential nodes from the networks. The authors conducted extensive experiments and the results were shown that by adjusting the parameter of the proportional factors, the proposed method can thoroughly sanitize privacy-sensitive sequence pattern networks and achieve the optimal values for security degree and connectivity degree measurements. In addition, the performance of the proposed method was shown to be stable for multiple networks with basically the same privacy-sensitive node ratio and be scalable for batches of networks with different sensitive nodes ratios.

KEYWORDS

Influential Nodes, Location Privacy Inference Attacks, Privacy-Sensitive Sequence Pattern Network, Sanitized, Trajectory

INTRODUCTION

With a widespread use of GPS (Global Positioning System) positioning devices in automotive and terminal equipment in addition to the fast development of social networks and location-based services, industry sectors can collect and store large amounts of trajectories in a variety of ways (Zhu, Zheng., & Wong, 2019), meaning that this type of data grows rapidly in daily life (Giannotti, 2011; Williams, Thomas, Dunbar, Eagle, & Dobra, 2015; Dobra, Williams, & Eagle, 2015). Analyzing trajectories using data mining tools can discover interesting patterns and regularities, which will help to provide auxiliary decisions for relevant industry applications (Gabrielli, Fadda, Rossetti, Nanni, Piccinini, Pedreschi et al., 2018; Blondel, Decuyper, & Krings, 2015), promote personalized medical care and precision marketing. In addition, trajectory data as a new type of data can also assist scientific workers to carry out intelligent transportation (Kujala, Aledavood, & Saramäki, 2016), urban planning (Louail, Lenormand, Ros, Picornell, Herranz, Friasmartinez et al., 2014; Li, Sun, Cao, He, & Zhu, 2016) and other research works (Ortale, Ritacco, Pelekis, Trasarti, Costa, Giannotti et al., 2008).

As technologies are intended to be neutral, they harbor neither benevolent nor malevolent intent with respect to the individuals using them. In particular, a curious or malicious user can also use the trajectory data mining tools to find non-interesting patterns. Specifically, this can include privacy-

DOI: 10.4018/IJDWM.2019070104

sensitive mobility patterns (i.e., mobility patterns involve privacy-sensitive spatial regions, such as military restricted areas, religious sites, private houses, private clubs, red-light-district, etc.), which will pose a threat to the location privacy of specific users (Giannotti & Pedreschi, 2008; de Montjoye, Hidalgo, Verleysen, & Blondel, 2013). Privacy-preserving data mining outsourcing (Liu, Wang, Shang, Li, & Zhang, 2017; Monreale, Rinzivillo, Pratesi, Giannotti, & Pedreschi, 2014) and privacy-preserving distributed data analytics (Monreale, Rinzivillo, Pratesi, Giannotti, & Pedreschi, 2014) are two methods to ensure that privacy-sensitive patterns are not detected by attackers in systems with trusted central servers and untrusted central servers, respectively. While, when a collector (i.e., location service provider) of trajectories wants to release (i.e., publish and share) the trajectories with a third party, the sanitization methods based on the strategy of knowledge hiding will be adopted, that is, (s)he must sanitize the trajectories to eliminate privacy-sensitive mobility patterns to prevent a threat to the privacy of the users whose trajectories were collected.

The existing sanitization methods for privacy-sensitive mobility patterns mainly aim to hide privacy-sensitive mobility patterns (Rajesh, Sujatha, & Lawrence, 2017; Bonchi & Ferrari, 2010; Aggarwal & Yu, 2008), while changing the original trajectories as little as possible. In addition, these methods are specified to certain types of mining techniques, which include association rule hiding (Tsai, Wang, Song, & Ting, 2016), sequence pattern hiding (Quang, Tai, Huynh, & Le, 2016), sequence rule hiding (Zhang, Wu, Chen, Liu, & Zhu, 2017) and so on.

However, these sanitization methods (Tsai, Wang, Song, & Ting, 2016; Quang, Tai, Huynh, & Le, 2016; Zhang, Wu, Chen, Liu, & Zhu, 2017) cannot effectively prevent location privacy inference attacks based on analyzing the relationship between privacy-sensitive mobility patterns. Specifically, an attacker can connect some single privacy-sensitive mobility patterns to construct a privacy-sensitive mobility pattern network and perform location privacy inference attacks based on the network connectivity analysis. In fact, there is a high probability of the occurrence of attacks on the privacy of locations based on privacy-sensitive mobility patterns, as researchers are more likely to study human mobility patterns from a network view. For example, previous studies (Cho, Myers, & Leskovec, 2011; Nguyen, & Szymanski, 2012) suggested highlighting to what extent human movements affect social dynamics and how social interactions influence the way people move, which was achieved by studying the interplay between human mobility networks and social networks. Furthermore, by exploring the interplay between human mobility networks and social networks, macroscopic characteristics of many complex geographic systems, such as traffic (Bajardi, Poletto, Ramasco, Tizzoni, Colizza, & Vespignani, 2011), energy (Louail, Lenormand, Picornell, Cantú, Herranz, Friasmartinez et al., 2015) and population (Balcan, Colizza, Gonçalves, Hu, Ramasco, Vespignani et al, 2009; Brockmann, & Helbing, 2013), can be also discovered as a mobility pattern network is an abstraction of spatial topological relations of a complex geographical system.

Therefore, it is necessary from a network view to analyze inference attacks based on the privacy-sensitive mobility patterns and design the corresponding countermeasures. In this paper, we aimed to study the location privacy inference attacks based on the privacy-sensitive sequence pattern network mined from trajectories and to design a method of sanitizing the privacy-sensitive sequence pattern network.

The remainder of this article is organized as follows. Section Preliminaries provides necessary preliminary information and the basic concepts utilized in our research. In Section Location Privacy Inference Attacks Based on Privacy-Sensitive Sequence Pattern Network, we define privacy attacks formally. In Section Proposed Sanitization Method, we present a method of sanitizing privacy-sensitive sequence pattern networks for defending against these attacks. In Section Experiments and discussion, we describe our comprehensive experiments and provide an analysis of the results. Section Conclusions and future work concludes the paper and discusses further work.

PRELIMINARY

Sequence Pattern of Trajectory

A trajectory is usually defined as a spatial-temporal evolution of a moving object (Brilhante, Macedo, Renso, & Casanova, 2011). A trajectory can be represented as: $SPD = \left\{SP_1, SP_2, ..., SP_n\right\}\left(n \geq 1\right)$, where $SPN = \left\{V, E\right\}$ is the number of sample points recorded during the movement of the object $V = \left\{dsr_1, dsr_2, ..., dsr_m\right\}\left(2 \leq m \leq 2n\right)$; $E = \left\{SP_1, SP_2, ..., SP_n\right\}$ represents a trajectory point; $SinglePattern$ represents the time stamp; and $\left(SP = \left\{dsr_1 \rightarrow dsr_2\right\}\right) \in SPD$ occurs before $dsr_j \in V\left(1 \leq j \leq 2\right)$.

Mobility patterns mined from trajectories are also known as "trajectory patterns," which consist of local patterns and global models. Generally, trajectory patterns refer to local patterns, which are also known as spatiotemporal sequence patterns and adopt two basic assumptions: 1) a pattern is frequent and therefore involves (or appears in) several trajectories; and 2) a pattern must describe the movement in space of the objects involved, instead of only showing some spatial or highly-abstracted spatial features. A sequence pattern of trajectory involves the following basic definitions:

Definition 1: A sequence pattern mined from trajectories (hereinafter referred to as a sequence pattern of trajectory) is defined as $T = \left\{tp_1, t_1, tp_2, t_2, ..., tp_n, t_n\right\}$, where n (o) represents a discretization spatial region and tp_i is the length of t_i.

Definition 2: For a trajectory tp_i and a sequence pattern of trajectory tp_{i+1}, $SP = \left\{dsr_1 \rightarrow dsr_2 \rightarrow, ..., \rightarrow dsr_m\right\}$ is said to support dsr_i, denoted as $1 \leq i \leq m$, if there exists integers m such that SP (i.e., the trajectory point $A = \left\{tp_1, t_1, tp_2, t_2, ..., tp_n, t_n\right\}, n \geq 1,$ is contained by the spatial region $B = \left\{dsr_1 \rightarrow dsr_2 \rightarrow, ..., \rightarrow dsr_m\right\}, m \geq 1$), A. The measurement of the support B to $A \supseteq B$ is formulated as $1 \leq i_1 < ...i_m \leq n$, otherwise $tp_k \in dsr_{i_k}$.

Definition 3: For a sequence database tp_k and a sequence pattern of trajectory dsr_{i_k}, the support $1 \leq k \leq m$ in A can be formulated as B. Furthermore, $Supp_A^B = 1$ is said to be frequent if $Supp_A^B = 0$ is not less than a user-specified minimum support threshold $SeD = \left\{T_1, T_2, ..., T_n\right\}$.

Definition 4: For a sequence pattern of trajectory $A = \left\{dsr_1 \rightarrow dsr_2 \rightarrow, ..., \rightarrow dsr_m\right\}$, A is called the length of SeD. Specifically, if $Supp_{SeD}^A = \dfrac{\sum_i^n Supp_{T_i}^A}{n} \times 100\%$, we call the pattern as a A; otherwise, if $Supp_{SeD}^A$, it is a $MinSup$. For the sake of simplicity, we only considered $A = \left\{dsr_1 \rightarrow dsr_2 \rightarrow, ..., \rightarrow dsr_m\right\}$ in this paper. In fact, a m is a combination of multiple A .For example, a $m = 2$ $SinglePattern$ can be obtained by combining the following multiple $m > 2$: $MultiPattern$, $SinglePatterns$. Following this, the sanitizing method proposed in this paper can also be extended to $MultiPattern$.

An example of $SinglePattern$ is given in Table 1.

Sequence Pattern Network

By connecting the sequence patterns of trajectories with common items (i.e., for sequence pattern A ⇒ C and sequence pattern C ⇒ B, they have a common term *C*), a sequence pattern network can be constructed, which is defined as follows:

Table 1. An example of SinglePatterns

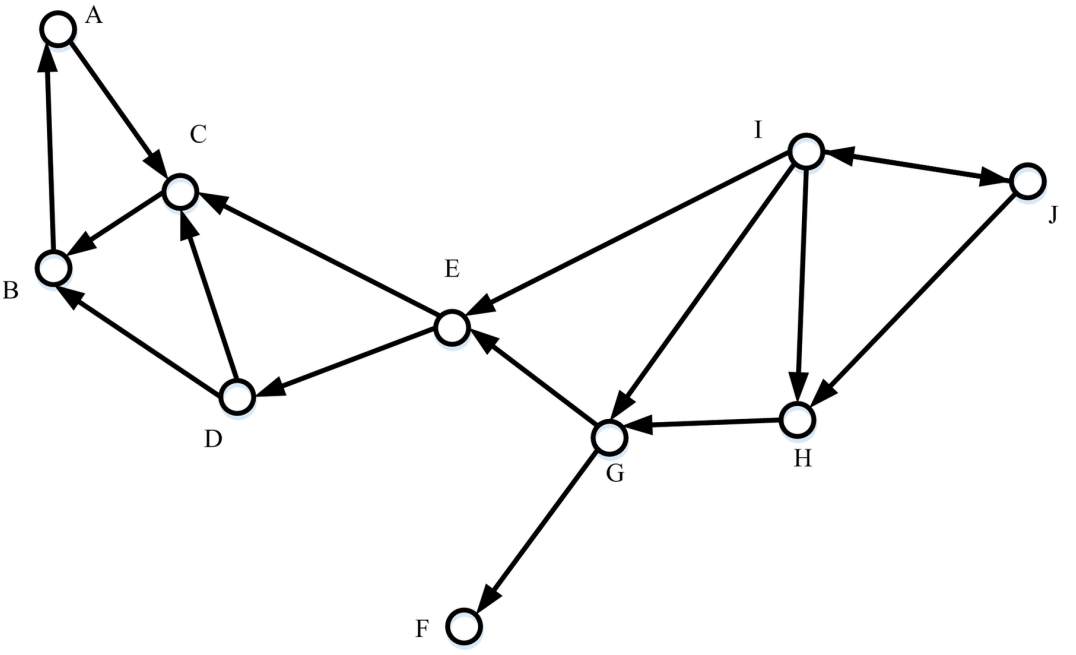
No.	Sequence pattern	No.	Sequence pattern	No.	Sequence pattern	No.	Sequence pattern
1	A ⇒ C	5	D ⇒ C	9	G ⇒ F	13	I ⇒ H
2	B ⇒ A	6	E ⇒ C	10	H ⇒ G	14	I ⇒ J
3	C ⇒ B	7	E ⇒ D	11	I ⇒ E	15	J ⇒ H
4	D ⇒ B	8	G ⇒ E	12	I ⇒ G	16	J ⇒ I

Definition 5: Given a sequence pattern database $MultiPattern$, $A = \left\{dsr_1 \rightarrow dsr_2 \rightarrow, ..., \rightarrow dsr_n\right\}$ is defined as a sequence pattern network, where $SinglePatterns$, $B_1 = \left\{dsr_1 \rightarrow dsr_2\right\}$, and the condition is satisfied that for any frequent $B_2 = \left\{dsr_1 \wedge dsr_2 \rightarrow dsr_3\right\}, \cdots, B_{n-1} = dsr_1 \wedge dsr_2 \wedge \ldots \wedge dsr_{n-1} \rightarrow dsr_n$ $MultiPatterns$, $SinglePatterns$.

For the sequence patterns in Table 1, the constructed sequence pattern network is shown as Figure 1.

We can see from Definition 5 that a sequence pattern network is actually a geographical network, as a node indicates a spatial region and an edge represents a frequent interaction between two different spatial regions corresponding two nodes.

Figure 1. Sequence pattern network constructed from sequence patterns in Table 1



LOCATION PRIVACY INFERENCE ATTACKS BASED ON PRIVACY-SENSITIVE SEQUENCE PATTERN NETWORK

A sequence pattern network can perform network connectivity analysis (i.e., a shortest path analysis), as it is a basic feature of a geographical network. Meanwhile, this analysis may pose a potential threat to the privacy of users' locations, once the sequence pattern network is privacy-sensitive. Specifically, as a spatial region corresponding to any node of a sequence pattern network is involved in a privacy-sensitive spatial region, the network will be defined as a privacy-sensitive sequence pattern network, which is formally defined as:

Definition 6: Given a sequence pattern network $SPN = \left\{V, E\right\}$ and a privacy-sensitive spatial region set $PSRs = \left\{psr_1, psr_2, ..., psr_n\right\}$, if $\exists dsr \in V$ and $\left(dsr \cap psr\right) \neq \varnothing$, where $psr \in PSRs$, then $SPN = \left\{V, E\right\}$ is called as a privacy-sensitive sequence pattern network, which is denoted as $PsSPN$, and dsr is called as a privacy-sensitive node.

An attack model generally consists of four components: (1) role classification; (2) private information; (3) background knowledge; and (4) inference attack process. In the context of location privacy inference attacks based on a privacy-sensitive sequence pattern network, the details are as follows:

1. **Role classification:** Attackers, who get large-scale trajectories released; and victims, whose historical trajectories are included in the released trajectories;
2. **Privacy information:** The shortest path along which a user moved from a spatial region, passed by the other spatial region(s) and arrived at another spatial region, and any one of the spatial regions may be privacy-sensitive;
3. **Background knowledge:** Privacy-sensitive spatial regions related to released trajectories, which are obtained by an attacker in a legitimate manner (e.g., getting data from a data exchange or sharing website) or by means of an illegal means (e.g., getting data by a network hacker attack) (Zhu Y, Zheng G., & Fitch M, 2018). In addition, the attacker knows that the historical trajectory of an identified user is actually present in the released trajectories;
4. **Inference attack process:** See below:
    a. An attacker obtains sequence patterns by mining released trajectories and constructs a sequence pattern network by connecting the sequence patterns with common items, for example, the sequence pattern network in Figure 1;
    b. The attacker achieves a privacy-sensitive sequence pattern network by matching the sequence pattern network's spatial regions (i.e., corresponding to all nodes of the network) with privacy-sensitive spatial regions (i.e., background knowledge).Continuing Figure 1 as an example, the attacker knows that node *C* is a privacy-sensitive node, and transforms the network in Figure 1 into a privacy-sensitive sequence pattern network shown as Figure 2;
    c. The attacker obtains the node corresponding to one trajectory point among a historical trajectory of a identified user (i.e., background knowledge), by spatially matching the trajectory point with the privacy-sensitive sequence pattern network's spatial regions. Assuming the attacker learns a user's historical trajectory point is located in the spatial region corresponding to the privacy-sensitive node *C* in Figure 2, then based on the shortest path between *C* and *A*, the attacker can infer that the user must arrive at *A* along the shortest path $C \rightarrow B \rightarrow A$ (Figure 3). as privacy-sensitive node *C*. Since the privacy-sensitive node *C* is the source of the shortest path $C \rightarrow B \rightarrow A$, we call this attack scenario a source attack.

Similarly, if the attacker learns the user's trajectory point falls into the spatial region corresponding to the node *J*, then based on the shortest path $J \rightarrow I \rightarrow E \rightarrow C$ (Figure 4), the attacker infers that the user went along the path to the spatial region corresponding to the node *C*. As the privacy-sensitive node *C* is the destination of the shortest path $J \rightarrow I \rightarrow E \rightarrow C$, we call this attack scenario a sink attack.

However, then based on the shortest path $J \rightarrow I \rightarrow E \rightarrow C \rightarrow B \rightarrow A$ (Figure 5), the attacker infers the user passed by the spatial region (e.g.,) corresponding to the node *C*. As the privacy-sensitive node *C* is a way point of the shortest path $J \rightarrow I \rightarrow E \rightarrow C \rightarrow B \rightarrow A$, we call this attack scenario an intermediate attack.

PROPOSED SANITIZATION METHOD

As trajectories collected from intelligent devices that are enhanced by GPS are usually distributed over large space-time domain, a privacy-sensitive sequence pattern network constructed from sequence patterns mined from the trajectories usually has an irregular structure and a non-trivial topology. Specifically, the privacy-sensitive sequence pattern network usually is a scale-free and small-world complex network, which is characterized by specific structural features: power-law degree distributions, short path lengths and high clustering (Brilhante, Macedo, Renso, & Casanova, 2011). The robustness of complex a network largely depends on its structure, specifically, the robustness is resilient to random failures (e.g., removing random nodes), but is quite vulnerable to targeted attacks (e.g., removing of hub nodes) (Gao, Chen, Nie, Ma, & Guan, 2017; Albert, Jeong, & Barabasi, 2000; Cohen, Erez, Ben-Avraham, & Havlin, 2001).

Considering the privacy-sensitive sequence pattern network in Figure 2, if we randomly delete the node *F,* the network connectivity does not cause much change (as shown in Figure 6). However, as we deliberately choose to delete the node *C*, the connectivity of the network will be severely damaged (as shown in Figure 7).

Figure 2. Privacy-sensitive sequence pattern network transformed from the network in Figure 1

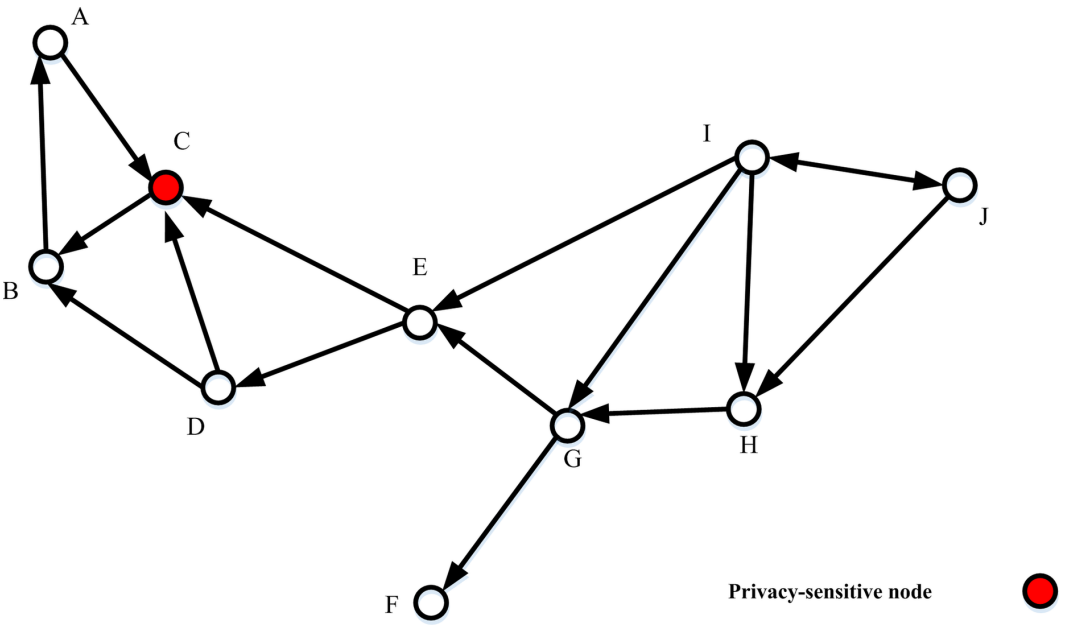


Figure 3. Scenarios of a source attack between privacy-sensitive node C to node A

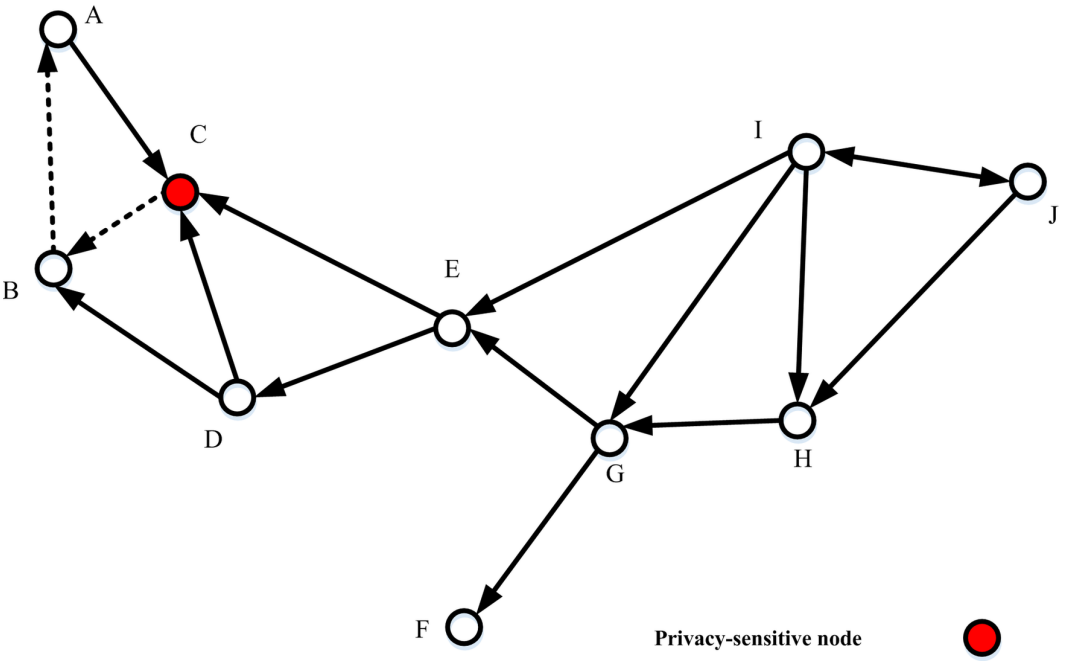


Figure 4. Scenarios of a sink attack between node J and privacy-sensitive node C

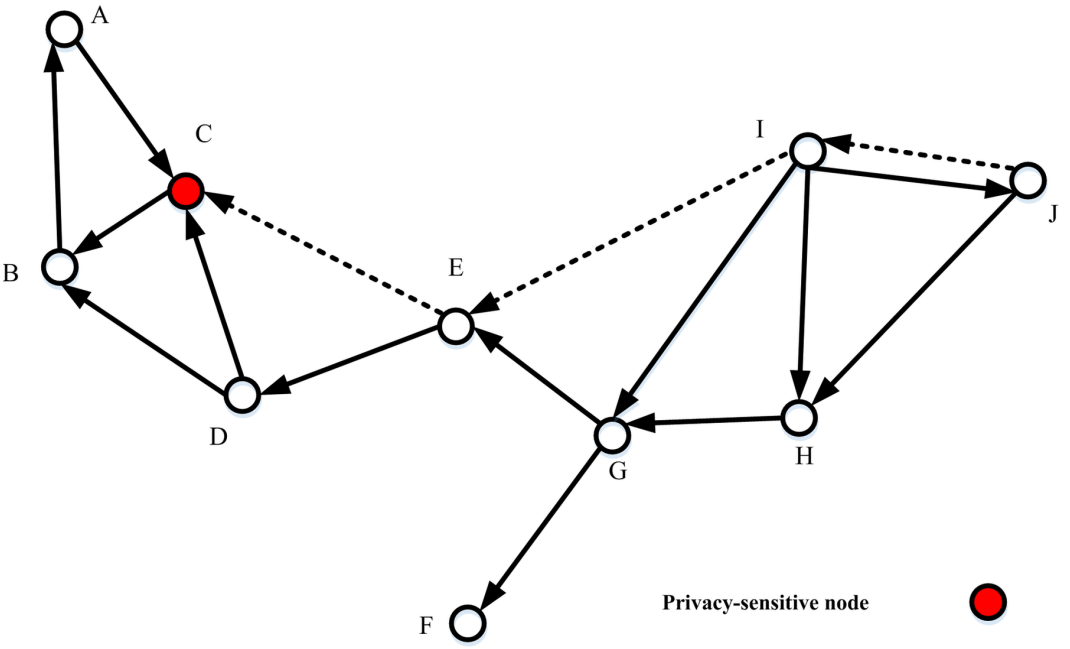


Figure 5. Scenarios of an intermediate attack between node J to node A through privacy-sensitive node C

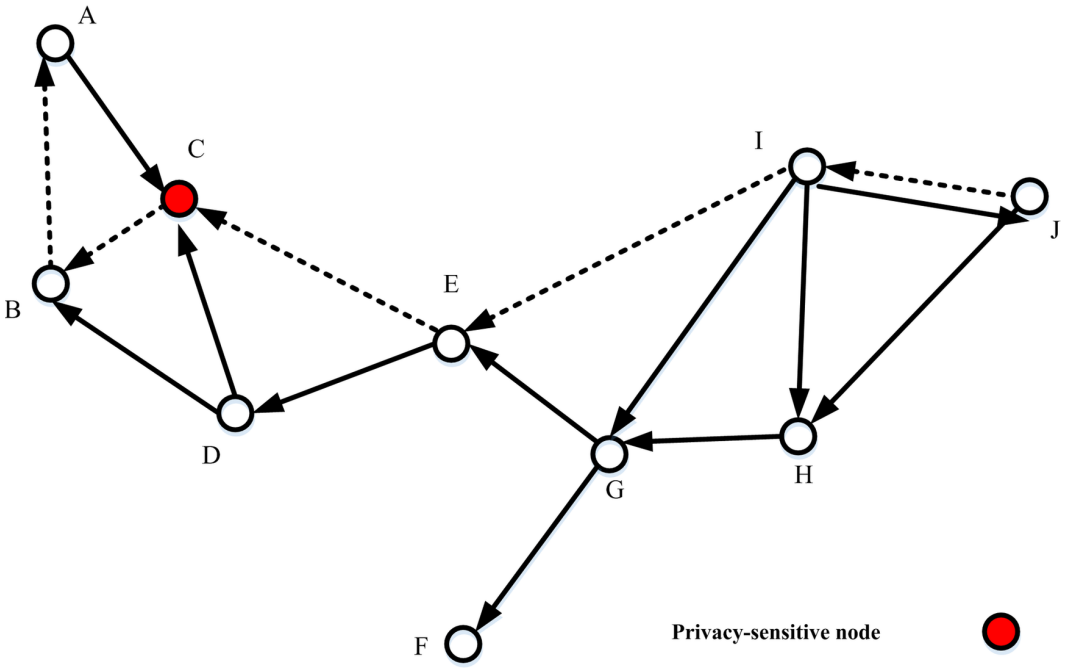
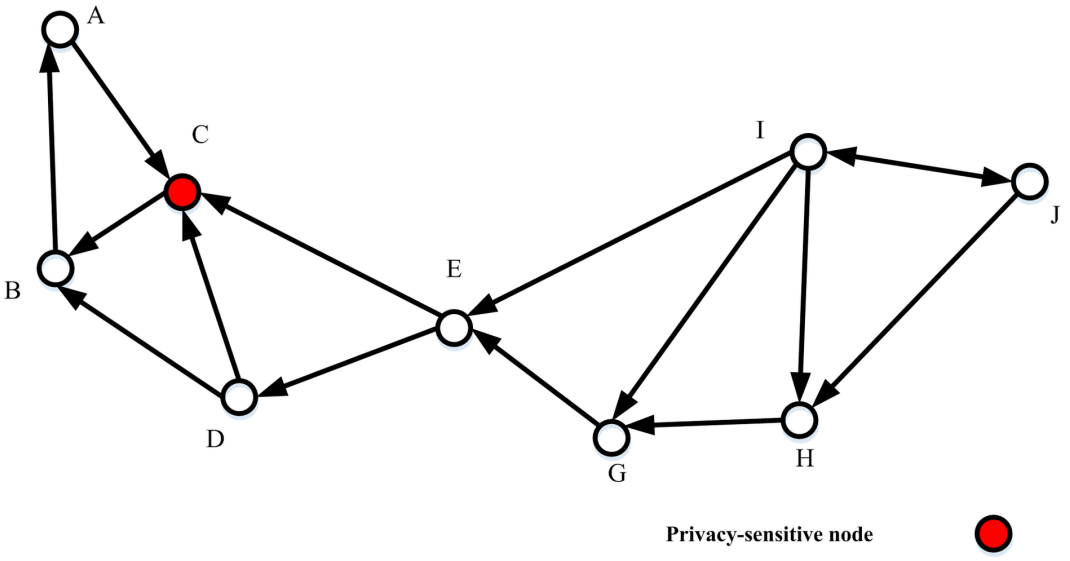


Figure 6. Sanitizing a privacy-sensitive sequence pattern network by randomly deleting its nodes



However, deliberately deleting nodes does not mean that all privacy-sensitive nodes in the privacy-sensitive sequence pattern network are directly deleted. For example, a privacy-sensitive sequence pattern network is shown in Figure 8. Figure 9 illustrates a sanitized network after all privacy-sensitive nodes are directly removed. However, an attacker can rediscover all the removed privacy-sensitive nodes and reconstruct the privacy-sensitive sequence pattern network. The reason

Figure 7. Sanitizing a privacy-sensitive sequence pattern network by deliberately deleting its nodes

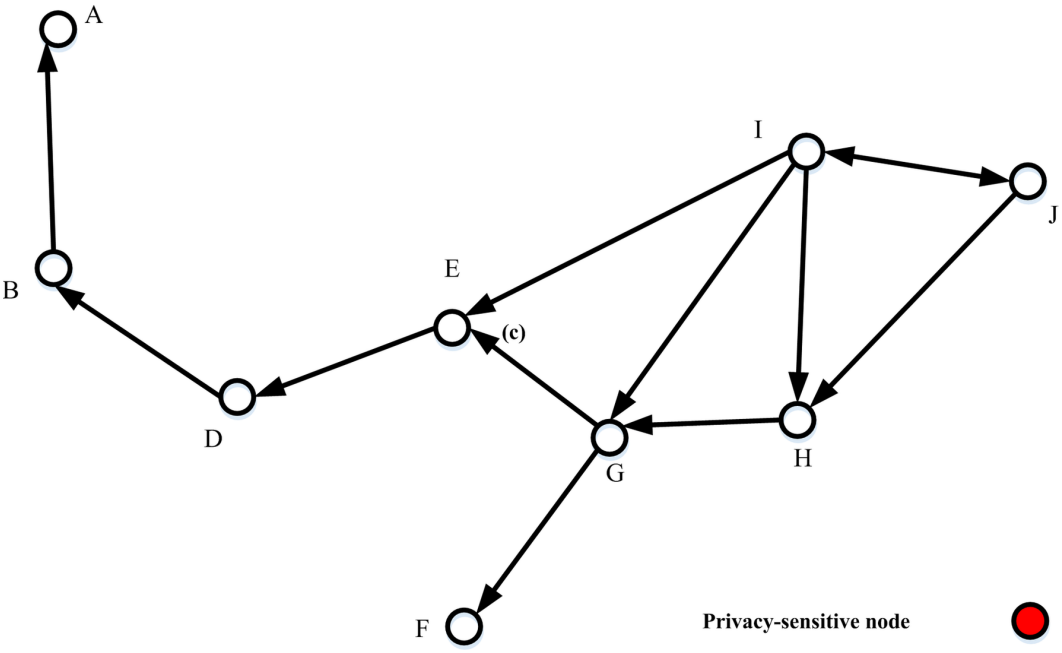
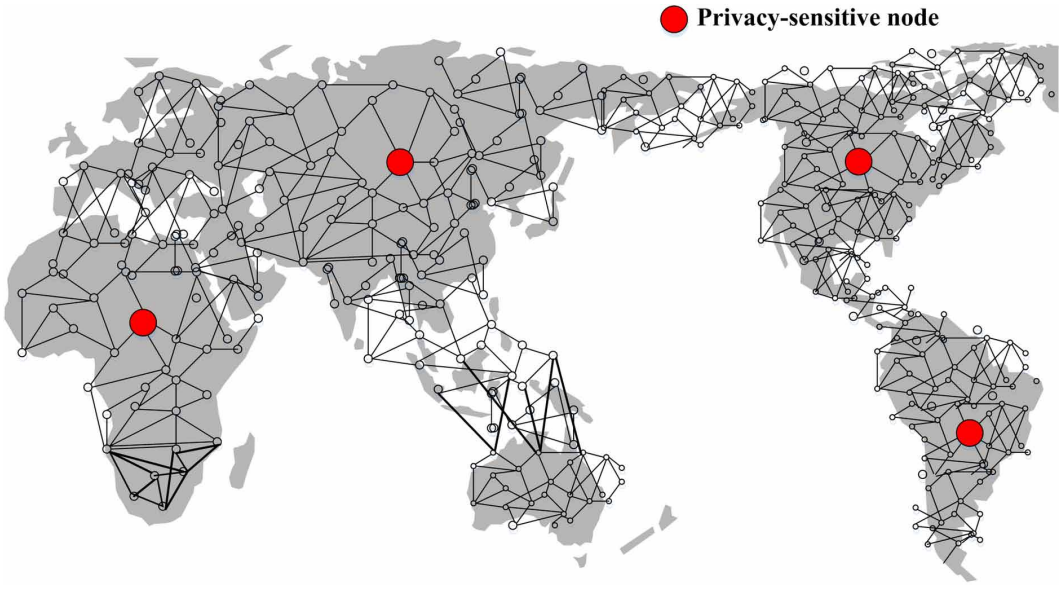
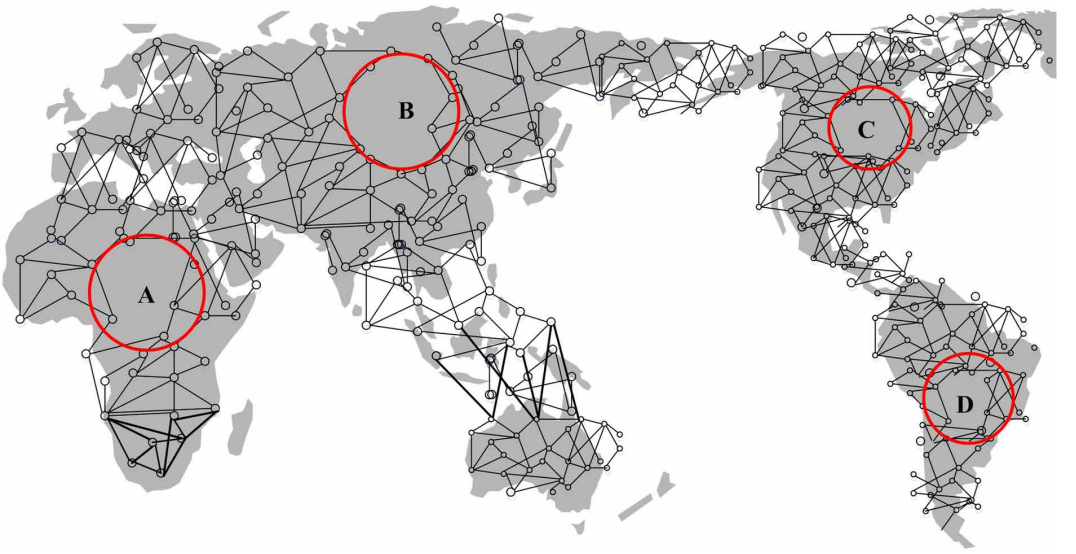


Figure 8. Directly deleted privacy-sensitive nodes mobility patterns before elimination



is that based on the joint analysis of the topology of the sanitized network and related geographical background knowledge, the attack can infer *A*, *B*, *C* and *D* should not be presented as "holes" in the sequence pattern network, furthermore, the spatial regions responding to *A*, *B*, *C* and *D* should be privacy-sensitive.

Figure 9. Directly deleted privacy-sensitive nodes mobility patterns after elimination



Therefore, the principle of selecting a node to delete should be the node's importance, rather than whether the node is privacy-sensitive. That is, it is essential to identify and remove influential nodes in a privacy-sensitive network.

Next, we first give some basic definitions, and then introduce our proposed method.

Basic Definitions

Definition 7: Given a privacy-sensitive sequence pattern network $PsSPN = \{V, E\}$, a adjacency matrix of $PsSPN$ is defined as a square matrix:

$$AM = \begin{pmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{12} & a_{22} & \cdots & a_{n2} \\ \cdots & \cdots & \cdots & \cdots \\ a_{1n} & a_{2n} & \cdots & a_{nn} \end{pmatrix}$$

where n is the number of the vertex set V; the element $a_{i,j}\left(1 \le i \le n, 1 \le j \le n\right)$ is one when there is an edge from vertex i to vertex j or zero when there is no edge.

Considering the privacy-sensitive sequence pattern network shown in Figure 2, after encoding the nodes *A-J* into *1-10*, its adjacency matrix AM is:

$$AM = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Definition 8: Given a privacy-sensitive sequence pattern network $PsSPN = \{V, E\}$ and its adjacency matrix:

$$AM = \begin{pmatrix} a_{11} & a_{21} & \ldots & a_{n1} \\ a_{12} & a_{22} & \ldots & a_{n2} \\ \ldots & \ldots & \ldots & \ldots \\ a_{1n} & a_{2n} & \ldots & a_{nn} \end{pmatrix}$$

the shortest path between node v_i and v_j is defined as d_{ij}, with all the shortest paths $d_{ij}\left(1 \le i \le n, 1 \le j \le n\right)$ constituting a shortest path matrix:

$$SPM = \begin{pmatrix} d_{11} & d_{21} & \ldots & d_{n1} \\ d_{12} & d_{22} & \ldots & d_{n2} \\ \ldots & \ldots & \ldots & \ldots \\ d_{1n} & d_{2n} & \ldots & d_{nn} \end{pmatrix}$$

Considering the privacy-sensitive sequence pattern network shown in Figure 2, it's the shortest path matrix is:

$$SPM = \begin{bmatrix} 0 & 2 & 1 & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ 1 & 0 & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ 2 & 1 & 0 & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ 2 & 1 & 1 & 0 & \infty & \infty & \infty & \infty & \infty & \infty \\ 3 & 2 & 1 & 1 & 0 & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & 0 & \infty & \infty & \infty & \infty \\ 4 & 3 & 2 & 2 & 1 & 1 & 0 & \infty & \infty & \infty \\ 5 & 4 & 3 & 3 & 2 & 2 & 1 & 0 & \infty & \infty \\ 4 & 3 & 2 & 2 & 1 & 2 & 1 & 1 & 0 & 1 \\ 5 & 4 & 3 & 3 & 2 & 3 & 2 & 1 & 1 & 0 \end{bmatrix}$$

Definition 9: Given a privacy-sensitive sequence pattern network and its corresponding shortest path matrix:

$$SPM = \begin{pmatrix} d_{11} & d_{21} & \ldots & d_{n1} \\ d_{12} & d_{22} & \ldots & d_{n2} \\ \ldots & \ldots & \ldots & \ldots \\ d_{1n} & d_{2n} & \ldots & d_{nn} \end{pmatrix}$$

where n is the total number of nodes in V', the center degree of the node $v_i \in V'$ is defined as $Cde_i = \frac{1}{n}\sum_{j=1,i\neq j}^{n}\frac{1}{d_{ij}}$. Essentially, Cde_i. reflects the convenience of the node v_i to other nodes $v_j\left(1 \le j \le n\right)$.

Considering the privacy-sensitive sequence pattern network shown in Figure 2, the center degree of node *A* is calculated as:

$$Cde_1 = \frac{1}{n}\left(\sum_{j=2}^{n}\frac{1}{d_{1j}}\right) = \frac{1}{10}\times\left(\frac{1}{2}+1+0+0+0+0+0+0+0\right) = \frac{1}{10}\times 1.5 = 0.15$$

Similarly, we obtain center degrees 0.10, 0.15, 0.25, 0.28, 0, 0.36, 0.31, 0.61, and 0.45 for the nodes *B*-*J*, respectively.

Definition 10: Given a privacy-sensitive sequence pattern network $PsSPN = \left\{V, E\right\}$ and its corresponding adjacency matrix:

$$AM = \begin{pmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{12} & a_{22} & \cdots & a_{n2} \\ \cdots & \cdots & \cdots & \cdots \\ a_{1n} & a_{2n} & \cdots & a_{nn} \end{pmatrix}$$

shortest path matrix:

$$SPM = \begin{pmatrix} d_{11} & d_{21} & \cdots & d_{n1} \\ d_{12} & d_{22} & \cdots & d_{n2} \\ \cdots & \cdots & \cdots & \cdots \\ d_{1n} & d_{2n} & \cdots & d_{nn} \end{pmatrix}$$

Therefore, the importance degree of the node $v_i \in V$ is defined as:

$$Ide_i = Cde_i \times \sum_{j=1, j\neq i}^{n} \frac{a_{ij} de_j Cde_j}{k^2}$$

where n is the total number of nodes in V; a_{ij} is a element of AM, that is $a_{ij} \in AM$:

$$Cde_i = \frac{1}{n} \sum_{j=1, i\neq j}^{n} \frac{1}{d_{ij}} \text{ and } Cde_j = \frac{1}{n} \sum_{i=1, i\neq j}^{n} \frac{1}{d_{ji}}$$

are the center degrees of node v_i and v_j $\left(1 \le i \le n, 1 \le j \le n\right)$; de_j is the degree of node *j*; and k is the average value of degree for all nodes in $PsSPN$.

Using Figure 2 as an example, where *n=10*; Cde_1 - Cde_{10} are 0.15, 0.10, 0.15, 0.25, 0.28, 0, 0.36, 0.31, 0.61 and 0.45; $a_{1\ 2}$ ~ $a_{1\ 10}$ are 0, 1, 0, 0, 0, 0, 0, 0 and 0;the values of de_1 ~ de_{10} are 2, 3, 4, 3, 4, 1, 4, 3, 5 and 3, respectively:

$$k = \frac{2+3+4+3+4+1+4+3+5+3}{10} = 3.2$$

The importance of node *A* is calculated as:

$$\begin{aligned} Ide_1 &= Cde_1 \times \frac{1}{\left\langle k \right\rangle^2} \times \left(a_{1,2} de_2 Cde_2 + a_{1,3} de_3 Cde_3 + a_{1,4} de_4 Cde_4 + \cdots + a_{1,10} de_{10} Cde_{10}\right) \\ &= \frac{0.15}{\left\langle 3.2 \right\rangle^2} \times \left(0 + 1 \times 4 \times 0.15 + 0 + 0 + 0 + 0 + 0 + 0 + 0\right) = \frac{0.15}{10.24} \times 0.6 = \frac{0.09}{10.24} = 0.00879 \end{aligned}$$

Similarly, we obtain the importance degrees of 0.00293, 0.00439, 0.02197, 0.03691, 0, 0.03937, 0.04359, 0.28488, and 0.17490 for the nodes *B-J*, respectively.

Definition 11: Given a privacy-sensitive sequence pattern network $PsSPN = \{V, E\}$ and its corresponding adjacency matrix:

$$AM = \begin{pmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{12} & a_{22} & \cdots & a_{n2} \\ \cdots & \cdots & \cdots & \cdots \\ a_{1n} & a_{2n} & \cdots & a_{nn} \end{pmatrix}$$

removing a node $v_i \in V\left(1 \le i \le n\right)$ from the network $PsSPN$ is equivalent to performing the following operations on its adjacency matrix $AM: \left(a_{ij} = 0\right) \wedge \left(a_{ji} = 0\right), \left(1 \le j \le n\right)$. The sanitized privacy-sensitive sequence pattern network is denoted as $sPsSPN$ and its adjacency matrix is denoted as sAM.

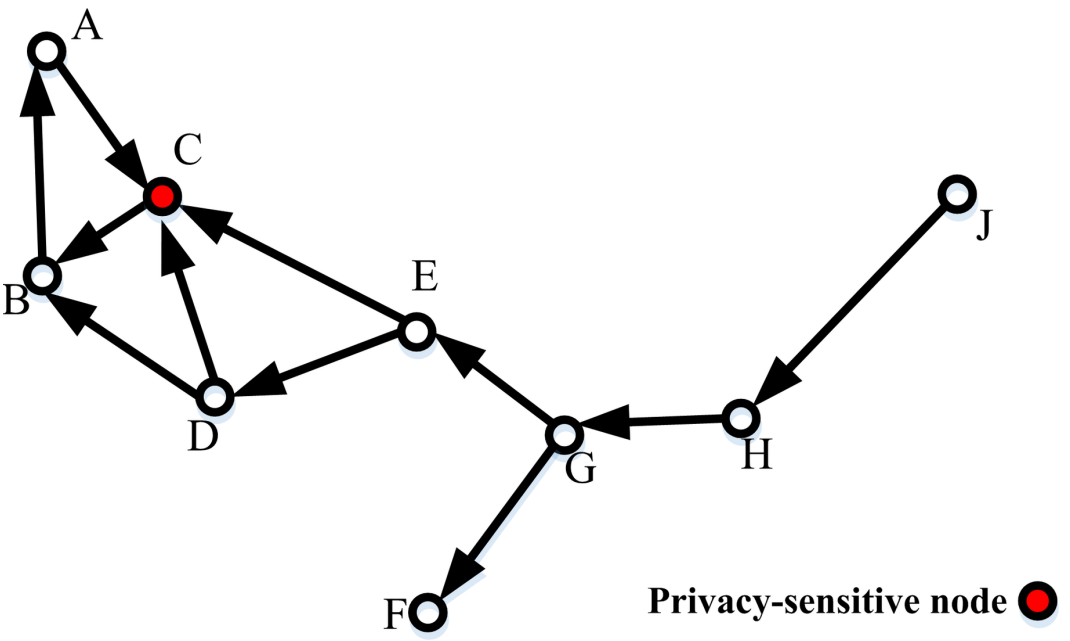
Continuing Figure 2 as an example, the privacy-sensitive sequence pattern network after the node E being sanitized is shown as Figure 10, while the corresponding adjacency matrix is:

$$sAM = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Definition 12: Given a privacy-sensitive sequence pattern network $PsSPN = \{V, E\}$ and its corresponding sanitized network $PsSPN_{sani} = \{V', E'\}$, the obtained security degree after the network being sanitized is defined as:

$$Sd = 1 - \frac{\sum_{i=1}^{m} \left(P'_{sink_i} + P'_{source_i} + P'_{inter_i}\right)}{\sum_{i=1}^{n} \left(P_{sink_i} + P_{source_i} + P_{inter_i}\right)}$$

Figure 10. Privacy-sensitive sequence pattern network after node I is sanitized



where n is the number of the vertex set as V; m is the number of the vertex set V'; P_{sink_i}, P_{source_i} and P_{inter_i} are the numbers of the shortest paths in $PsSPN$ for performing source attacks, sink attacks and intermediate attacks, respectively; and P'_{sink_i}, P'_{source_i} and P'_{inter_i} are the numbers of the shortest paths in $PsSPN_{sani}$.

In Figure 2, the shortest paths for sink attack contain $A \to C$, $B \to A \to C$, $D \to C$, $E \to C$, $G \to E \to C$, $H \to G \to E \to C$, $I \to E \to C$ and $J \to I \to E \to C$. Essentially, $P_{sink}=8$. If the node I is removed from the privacy-sensitive sequence pattern network, the shortest paths for the sink attack consist of $A \to C$, $B \to A \to C$, $D \to C$, $E \to C$, $G \to E \to C$, $H \to G \to E \to C$ and $J \to H \to G \to E \to C$. Therefore, $P'_{sink}=7$. Similarly, we obtain $P_{source}=2$, $P_{inter}=11$ as well as $P'_{source}=2$ and $P'_{inter}=9$. Following this:

$$Sd = 1 - \frac{7+2+9}{8+2+11} = 0.143$$

Definition 13: Given a privacy-sensitive sequence pattern network $PsSPN = \{V, E\}$ and its corresponding sanitized network $PsSPN_{sani} = \{V', E'\}$, the maximum connected sub-graph of $PsSPN_{sani}$ is $McSubG = \{V'', E''\}$, and the utility degree measurement is defined as:

$$Ud = \frac{m}{n}$$

where n and m are the numbers of the vertex set V and V'', respectively.

In Figure 2, after node *I* is removed, the maximum connected sub-graph is $McSubG = \{V'', E''\}$, where, $V'' = \{A, B, C, D, E, F, G, H, J\}$. Therefore:

$$Ud = \frac{m}{n} = \frac{9}{10} = 0.9$$

Algorithms

Following this, we described the detailed implementation process of destroying the connectivity of a privacy-sensitive sequence pattern network, which consists of three key steps: 1) calculating the nodes' importance degrees and sorting them; 2) selecting a proportion of the nodes to remove from the network; and 3) calculating the obtained security degree and the change of connectivity degree after the network being sanitized. The pseudo codes of the implementation process are shown in Algorithm 1-4, where Algorithm 1 is the main program and Algorithm 2-4 are the subroutines. We used Java to implement the algorithms and the adopted software is IntelliJ IDEA 15.0.4.

Algorithm 1 sanitizes a privacy-sensitive sequence pattern network in addition to calculating the obtained security degree and the change of connectivity degree after the network being sanitized. The pseudo codes of Algorithm 1 are shown as follow:

Algorithm 1: Sanitization ($PsSPN$, $SensitiveVertexlist$, δ , ref sAM , ref Sd and ref Ud)

Input: $PsSPN$ represents a privacy-sensitive sequence pattern network; δ represents the proportion factor, that is, the ratio of the number of nodes that need to be removed to the number of nodes in $PsSPN$; and $SensitiveVertexlist$ represents a set of privacy-sensitive nodes.
Output: sAM represents an adjacency matrix of the network $PsSPN$ after being sanitized; Sd represents the obtained security degree after the network $PsSPN$ being sanitized; and Ud represents the utility degree after the network $PsSPN$ being sanitized.

$AM = GetAM(PsSPN);$
$SPM = GetSPM(AM');$
$Importancelist = GetSortImportance(SPM, AM);$
1. for i = 1 to $(Importancelist \cdot count) \times \delta$
2. $KeyVertexlist \cdot add(Importancelist[i]);$.
3. end for
4. for i = 1 to $KeyVertexlist \cdot count$
5. $sAM = Sanitizing(AM, i);$
6. end for

$sSPM = GetSPM\left(sAM\right);$

$$Sd = 1 - \frac{GetSensitivePath\left(SPM, SensitiveVertexlist\right)}{GetSensitivePath\left(sSPM, SensitiveVertexlist\right)}$$

Line 1 obtains the adjacency matrix of the privacy-sensitive sequence pattern network $PsSPN$ (Definition 7); Line 2 obtains the shortest path matrix SPM from the adjacency matrix AM' (Definition 8). Line 3 calls a function $GetSortImportance$ with parameters SPM and AM to obtain a node set sorted by their importance degrees. Lines 4-6 select the top $\left(Importancelist \cdot count\right) \times \delta$ nodes in $Importancelist$ to add to the node set $KeyVertexlist$, which will be removed. Lines 7-9 removed the nodes in $KeyVertexlist$ from the network $PsSPN$ by performing operations on its adjacency matrix AM and obtaining the sanitized adjacency matrix sAM (Definition 11). Line 10 gets the shortest path matrix $sSPM$ from the sanitized adjacency matrix sAM (Definition 8). Line 11 calls a function $GetSensitivePath$ with parameters $SensitiveVertexlist$, SPM and $sSPM$ to respectively obtain the privacy-sensitive paths from AM and sAM, before finally getting the obtained security degree Sd after the network $PsSPN$ being sanitized. Line 12 calls a subroutine $GetMcSubG$ with the parameter sAM to obtain the maximum connected sub-graph in the sanitized network $PsSPN$, before obtaining the utility degree of the network, which is the ratio of the node count of the maximum connected sub-graph to the node count (i.e., the row count (or column count) of the matrix AM of the network $PsSPN$ (Definition 13).

Algorithm 2 is a node importance sorting algorithm for a privacy-sensitive sequence pattern network. The pseudo codes of Algorithm 2 are shown as follow:

Algorithm 2: $GetSortImportance\left(SPM, AM\right)$

Input: SPM and AM, which respectively represent a shortest path matrix and an adjacency matrix of a privacy-sensitive sequence pattern network.
Output: $Importancelist$, which represents a node set sorted by importance degrees in an ascending order.

$n = SPM \cdot rowCount;$
1. *for i = 1 to n*
2. *for j = 1 to n*
3. *if (j! = i)*
4. $d_{ij} = SPM \cdot GetEntry\left(i, j\right);$
5. $tmpSum += 1 / d_{ij}$;
6. *end if*
7. *end for*
8. $Cde_i = tmpSum / n$;
9. *end for*
10. *for i = 1 to n*
11. *for j = 1 to n*
12. $a_{ij} = AM.GetEntry\left(i, j\right);$
13. $de_i += a_{ij}$;
14. *end for*
$k += de_i$
15. *end for*

$k = square\left(\frac{k}{n}\right);$
$tmpSum = 0;$
16. for i = 1 to n
17. for j = 1 to n
18. $tmpSum+ = \frac{\left(a_{ij} \times de_j \times cde_j\right)}{k}$;
19. end for
20. $Ide_i = Cde_i \times tmpSum$;
21. end for
22. $Importancelist = SortbyAsce\left(Ide_i\right)$;

Line 1 obtains $rowCount$ of a shortest path matrix SPM. Lines 2–10 calculate the center degrees of nodes in SPM (Definition 9). Lines 11–17 calculate the degrees of nodes in SPM. Line 18 obtains the value of k^2. Lines 19–25 calculate the importance degrees of nodes in SPM (Definition 10). Line 26 obtains the set of nodes sorted by importance degrees in an ascending order.

Algorithm 3 calculates the sum of shortest paths for source attacks, sink attacks and intermediate attacks in a privacy-sensitive sequence pattern network. The pseudo codes of Algorithm ***3*** are shown as follow:

Algorithm 3: $GetSensitivePath\left(SPM, SensitiveVertexlist\right)$

Input: SPM, which represents a shortest path matrix of a privacy-sensitive sequence pattern network; and $SensitiveVertexlist$, which represents a set of sensitive nodes.
Output: $totalSensitivePath$, which represents the sum of three shortest paths $P_{source}P_{sink}$ and P_{inter}.
$n = SPM \cdot rowCount;$
1. for i = 1 to $SensitiveVertexlist \cdot count$
2. for j = 1 to n
3. for k = 1 to n
4. if $\left(SensitiveVertexlist\left[i\right] \cdot NodeNum == j\right) \&\& \left(SPM \cdot GetEntry\left(i,j\right)! = 0\right)$
5. $P_{source}++;$
6. end if
7. if $\left(SensitiveVertexlist\left[i\right] \cdot NodeNum == k\right) \&\& \left(SPM \cdot GetEntry\left(i,j\right)! = 0\right)$
8. $P_{sink}++;$
9. end if
10. if $\left(SPM \cdot GetEntry\left(i,j\right)! = 0\right)$
11. For each node in $SPM \cdot Fullpath\left(i,j\right)$
12. if $node \cdot Num == SensitiveVertexlist\left[i\right] \cdot NodeNum$
13. $P_{inter}++;$
14. end if
15. end for each
16. end if
17. end for

18. *end for*
19. $totalSensitivePath = P_{source} + P_{sink} + P_{inter}$;

Line 1 obtains the node count of a privacy-sensitive sequence pattern network by the $rowCount$ of the network's shortest path matrix SPM . Lines 2–19 obtain the privacy-sensitive shortest paths number $P_{source} P_{sink}$ and P_{inter} by comparing each node in the $SensitiveVertexlist$ with the elements in the shortest path matrix SPM . Lines 5–7 obtain the number of the shortest paths P_{source} by counting the number of the elements in SPM , which have nonzero values and have the row numbers of elements equal to the number of nodes in the $SensitiveVertexlist$. Lines 8–10 obtain the number of the shortest paths P_{sink} by counting the number of the elements in SPM , which have nonzero values and have the column numbers of the elements equal to the number of the nodes in the $SensitiveVertexlist$. Lines 11–17 obtain the number of the shortest paths P_{inter} by counting the number of the elements in SPM , which have nonzero values and have their full paths containing the number of the nodes in $SensitiveVertexlist$. Finally, Line 20 obtains the sum of the three shortest paths, which is $totalSensitivePath$.

Algorithm 4 obtains the maximum connected sub-graph of a privacy-sensitive sequence pattern network sanitized. The pseudo codes of Algorithm 4 are shown as follow:

Algorithm 4: $GetmCSubG\left(sAM\right)$

Input: sAM, which is an adjacency matrix of a sanitization privacy-sensitive sequence pattern network.
Output: ArrayList <Integer> *maxGraph*, which is the maximum connected sub-graph in the sanitization privacy-sensitive sequence pattern network.

1. *boolean [] visited=new boolean[* $sAM \cdot rowcount$ *];*
2. *int depth = 0;*
3. *for i = 1 to* $sAM \cdot rowcount$
4. $visited\left[i\right] = false$;
5. *end for*
6. *for i = 1 to* $sAM \cdot rowcount$
7. *if (!visited[i])*
8. *ArrayList<Integer>list = new ArrayList<>();*
9. $DFS\left(sAM, i, list\right)$;
10. *if(list.size()>depth)*
11. *depth = list.size();*
12. *maxGraph = list;*
13. *end if*
14. *end if*
15. *end for*

Lines 1-5 initialize the state variable *visited*. Lines 6-15 perform a depth-first search DFS for the matrix sAM to get a series of connected sub-graphs *list*, before obtaining the maximum connected sub-graph $maxGraph$ by comparing the nodes number of all the connected sub-graphs.

EXPERIMENTS AND DISCUSSION

Data Preparation

First, in order to test the performance stability of our proposed method and to explore the influence from parameters of the implementation algorithms, we adopted nine batches of experimental data sets, which are simulated by mining sequence patterns from the pre-processed GPS trajectories (Zhang, Wu, Chen, Liu, & Zhu, 2017) and specifying privacy-sensitive sequence patterns. The basic information of the nine batches of sequences patterns is shown in Table 2.

Second, in order to test the performance scalability of our proposed method, we simulated 81 batches of privacy-sensitive sequences patterns and obtained 81 privacy-sensitive sequence pattern networks by connecting the sequence patterns with common items. The basic information of the 81 privacy-sensitive networks is shown in Table 3, where these 81 networks are divided into 7 levels based on the ratios of the number of privacy-sensitive nodes to the total number of nodes of the networks.

Experimental Result and Analysis

Experiment 1: Changes of Security Degrees and Utility Degrees With Proportion Factor

We adopted our proposed method to sanitize the simulated nine privacy-sensitive sequence pattern networks, which are constructed from batches of privacy-sensitive sequences patterns in Table 2. By gradually increasing the value of proportion factor $\left(\delta\right)$ from 0.05 to 0.45 at intervals of 0.05, we collected the changes in security degrees and utility degrees after the nine networks being sanitized, which are shown in Figure 11 and Figure 12, respectively.

As we can see from Figure 11, for all nine privacy-sensitive sequence pattern networks, the obtained security degrees quickly formed a trend toward a higher range of values with an increase in proportion factor (i.e., more and more influential nodes were removed from the networks). Specifically, after the proportion factor reached 0.3, the obtained security degree was at a level of 1, meaning that the source attacks, sink attacks and intermediate node attacks can be completely avoided by removing all privacy-sensitive shortest paths and P_{inter}.

Meanwhile, the opposite trends are observed for the change in utility degrees, which is shown in Figure 12. Essentially, the utility degrees of the sanitized privacy-sensitive sequence pattern networks gradually decline with an increase in the proportion factor, which means that removing the influential nodes from the privacy-sensitive sequence pattern networks can also reduce the utility of the networks. This means that some non-privacy-sensitive shortest paths may also be eliminated

Table 2. Basic information of nine batches of sequences patterns

No.	Number of Sequence Patterns	Number of Nodes	Number of Privacy-Sensitive Nodes
1	39	78	8
2	30	60	6
3	36	72	8
4	28	52	5
5	24	48	4
6	37	74	7
7	25	50	6
8	39	78	7
9	27	54	6

Table 3. Basic information of 81 privacy-sensitive sequence pattern networks

Level	Ratio of Privacy-Sensitive Nodes	Number of PsSPN	Level	Ratio of Privacy-Sensitive Nodes	Number of PsSPN
1	0.1875	1	4	0.4	4
2	0.2143	1		0.4118	2
	0.2308	2		0.4167	7
	0.25	4		0.4211	2
	0.2667	3		0.4286	2
	0.2857	2		0.4375	3
	0.2941	1		0.4444	1
3	0.3077	1		0.4545	1
	0.3125	1		0.4615	1
	0.3158	1		0.4667	5
	0.3333	2	5	0.5	10
	0.3529	2		0.5385	3
	0.3571	5		0.5455	1
	0.3636	2		0.5833	4
	0.375	1	6	0.6	1
	0.3846	1		0.6667	1
	0.3889	2	7	0.7	1

from the networks, which can be seen as a side effect of our proposed sanitization method and should be effectively minimized.

Therefore, we hypothesized whether the optimal values of security degree and utility degree can be obtained by adjusting the proportion factor. We used the average of the security degree and utility degree as an indicator to find out the optimal value, with the experimental results shown in Figure 13.

We can see from Figure 13 that for all privacy-sensitive sequence pattern networks, the change trends of the average values of security degrees and utility degrees with the proportion factor are very obvious: the average values increase first, reach a peak and finally decrease gradually. Essentially, for each privacy-sensitive sequence pattern network, there is a peak of the average value, which means that the optimal values of security degrees and connectivity degrees for each network can be achieved by adjusting their proportion factors. For example, for the No 6 privacy-sensitive sequence pattern network, the average value of 0.81275 is the maximum when the proportion factor is 0.1.

Experiment 2: Changes of Security Degree and Utility Degree With Ratios of Privacy-Sensitive Nodes

We adopted our proposed method to sanitize the generated 81 privacy-sensitive sequence pattern networks in Table 3, with the experimental results shown in Figure 14 and Figure 15.

As we can see from Figure 14, for a specific proportion factor, the changing trends of the obtained security degrees are overall gradually reduced with increasing levels of ratios of privacy-sensitive nodes. This result means that it is more difficult to sanitize a privacy-sensitive network with a higher ratio of privacy-sensitive nodes. Specifically, more influential nodes may need to be removed from the privacy-sensitive network in order to eliminate the privacy-sensitive shortest paths among the

Figure 11. Changes in security degrees with changes in proportion factors

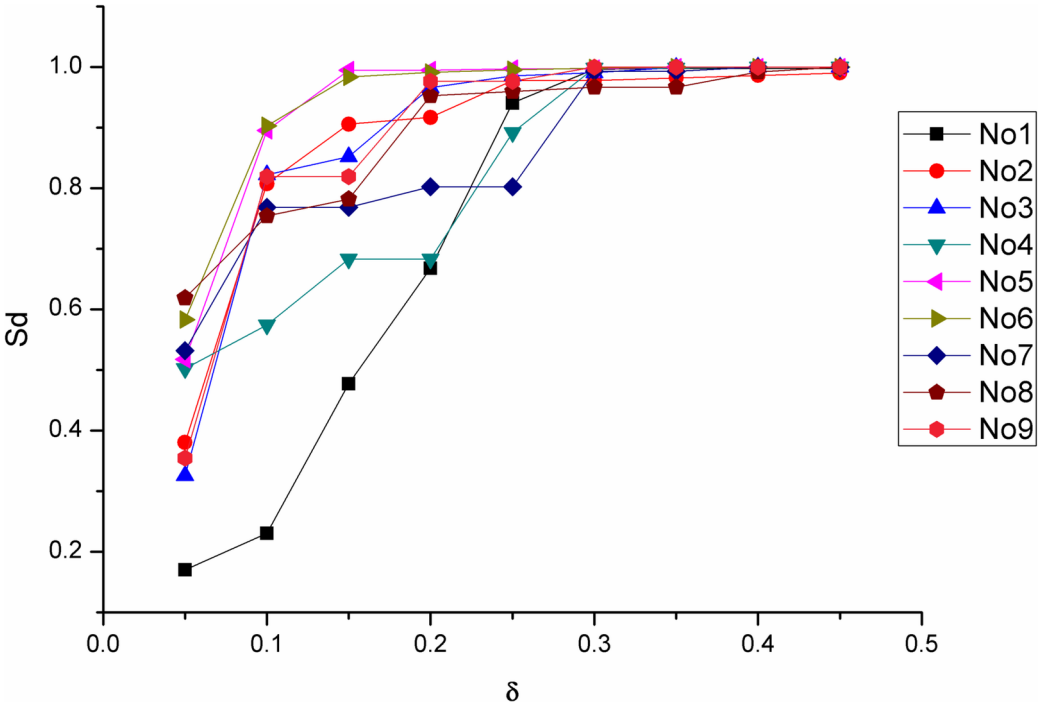


Figure 12. Changes in utility degrees with changes in proportion factors

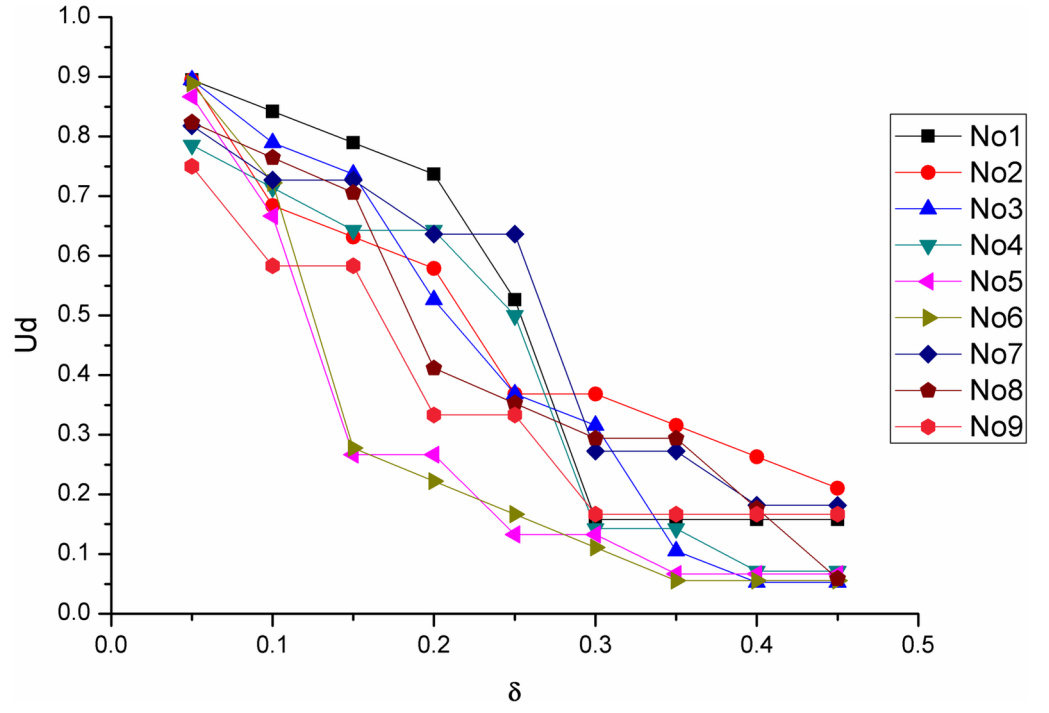


Figure 13. Changes in the average of security degree and utility degree with change in proportion factor for nine privacy-sensitive sequence pattern networks

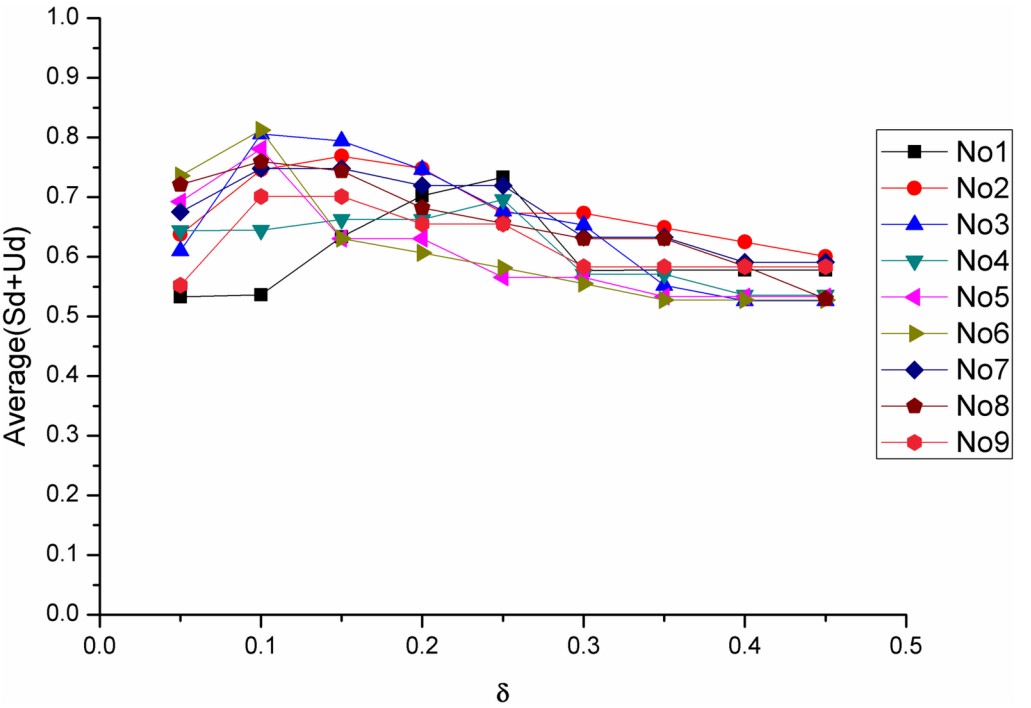


Figure 14. Changes in security degree with levels of ratios of privacy-sensitive nodes

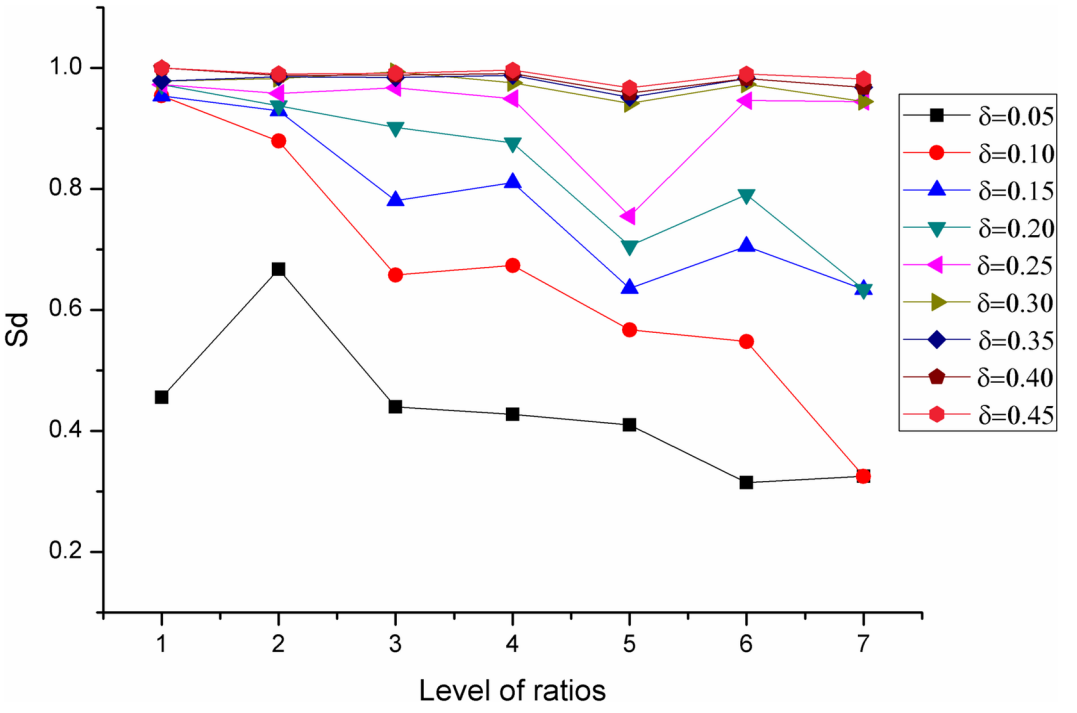
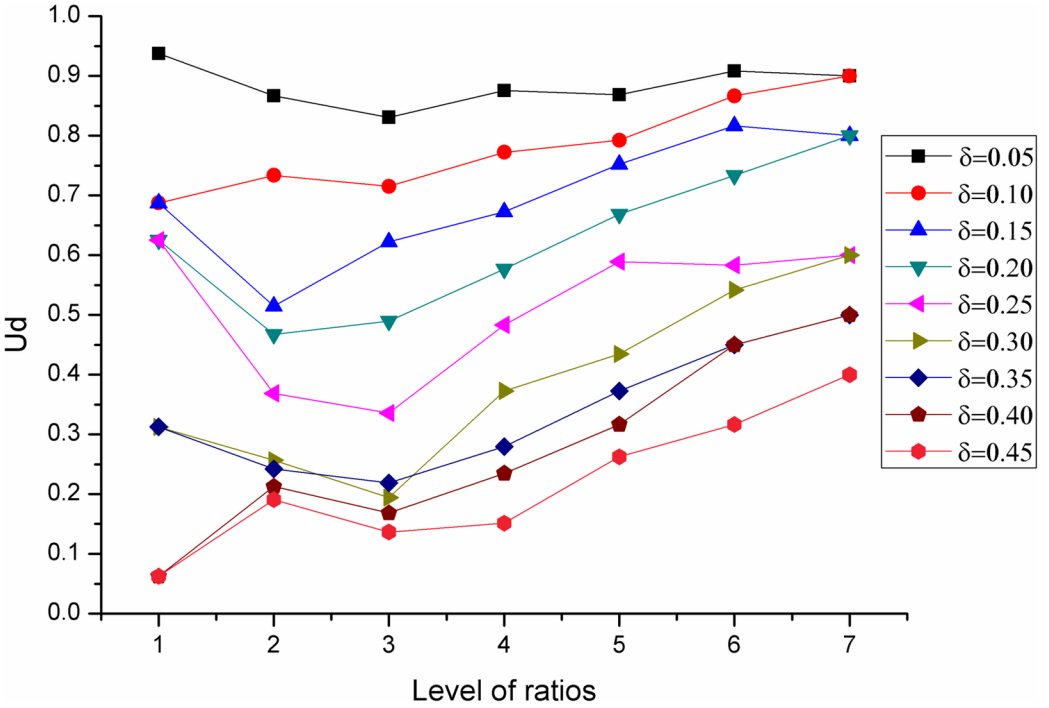


Figure 15. Changes in utility degree with levels of ratios of privacy-sensitive nodes



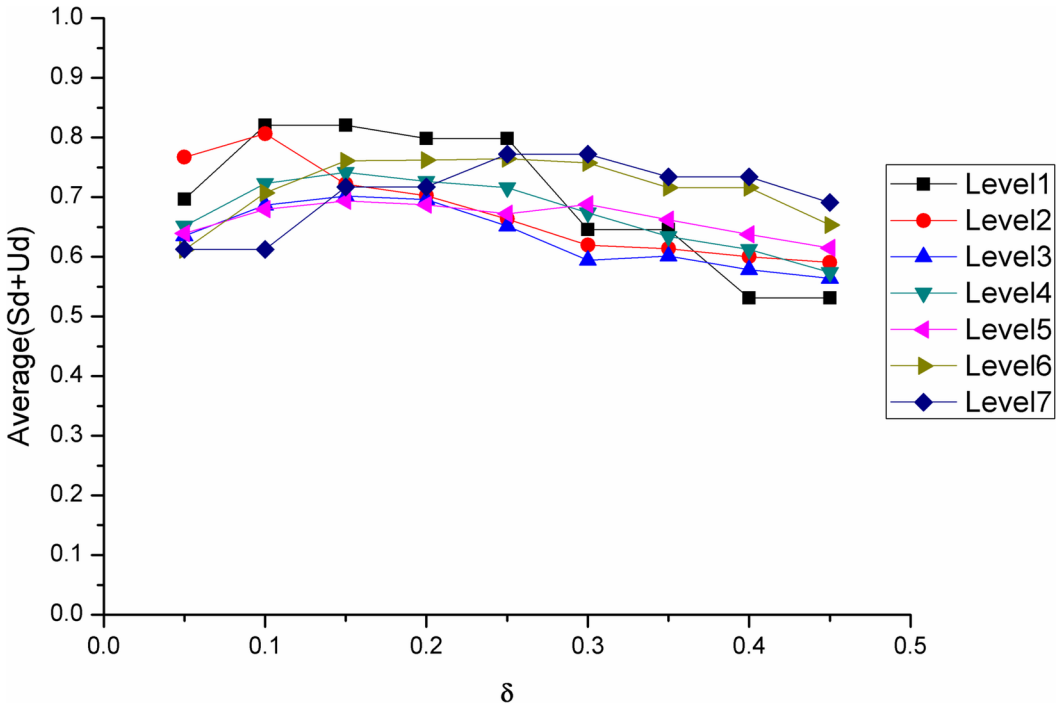
network. For example, in order to achieve a security degree of 0.8, for the networks with 1 or 2 levels of ratios of privacy-sensitive nodes, the proportion factor parameters only need to be set to greater than or equal to 0.10. However, for the networks with 3 or 4 levels of ratio of sensitive nodes, the proportion factor parameters need to be set to greater than or equal to 0.20. Furthermore, for the networks with 6 or 7 levels of ratio of privacy-sensitive nodes, the proportion factor parameters need to be set to greater than or equal to 0.25.

It is also worth noting that for all the networks with 1–7 levels ratio of sensitive nodes, the obtained security degrees basically tend to be 1 after the proportion factor was up to 0.3, which means that these networks can be also completely sanitized.

From Figure 15, we see that the overall changing trends of the obtained utility degrees are gradually increased with an increase in the levels of ratio of privacy-sensitive nodes, which means that the number of non-privacy-sensitive paths removed from privacy-sensitive networks is gradually reduced with an increase in the privacy-sensitive nodes ratios. This trend is what we expect.

Similarly, we also hypothesized whether the optimal values of security degree and utility degree of privacy-sensitive networks with different privacy-sensitive nodes ratios can be also obtained by adjusting proportion factors. We also use the average value of the security degree and utility degree as the indicator to find out the optimal value, with the experimental results shown in Figure 16. We can see that the changing trends are same with the ones in Figure 13. Essentially, for each network with different privacy-sensitive nodes ratios, an optimal value of security degree and utility degree can be achieved by setting an appropriate proportion factor parameter, which means that our proposed method has scalability for privacy-sensitive networks with different privacy-sensitive node ratios.

Figure 16. Changes in average of security degree and utility degree with changes in proportion factors for 81 privacy-sensitive sequence pattern networks with 7 levels of privacy-sensitive nodes ratios



CONCLUSION AND FUTURE WORK

To overcome the fact that most existing sanitization methods cannot effectively prevent location privacy inference attacks based on a privacy-sensitive sequence pattern network constructed from sequence patterns mined from trajectories, we proposed a method for sanitizing the privacy-sensitive sequence pattern network by identifying and removing influential nodes from the networks. We conducted extensive experiments and the results show that our proposed method can thoroughly sanitize privacy-sensitive sequence pattern networks from location privacy inference attacks, such as source attacks, sink attacks, intermediate attacks and so on, by completely eliminating the privacy-sensitive shortest paths in the network. In addition, our proposed method can achieve the optimal values for security degree and utility degree by adjusting the proportion factor parameter. Finally, the performance of our method was shown to be stable for multiple networks with basically the same privacy-sensitive node ratio and be scalable for batches of networks with different sensitive nodes ratios.

With the development of big data technology (such as Hadoop, Spark and so on), attackers may use the technology to explore the released trajectories and perform location privacy inference attacks. Therefore, it is a direction for us to study how to revise our proposed method to deal with such attacks in the future.

ACKNOWLEDGMENT

This research is supported by the Jiangsu Government Scholarship for Overseas Studies, grants from the Natural Science Foundation of China under grant number 41201465 and grants from the Natural Science Foundation of Jiangsu province under grant number BK2012439, BE2016774. The authors wish to thank the anonymous reviewers for their valuable comments.

REFERENCES

Aggarwal, C. C., & Yu, P. S. (2008). Privacy-preserving data mining: Models and algorithms. *Advances in Database Systems*, *35*, 11–52. doi:10.1007/978-0-387-70992-5_2

Albert, R., Jeong, H., & Barabasi, A. L. (2000). Error and attack tolerance of complex networks. nature. *Nature*, *406*(6794), 378–382. doi:10.1038/35019019 PMID:10935628

Bajardi, P., Poletto, C., Ramasco, J. J., Tizzoni, M., Colizza, V., & Vespignani, A. (2011). Human mobility networks, travel restrictions, and the global spread of 2009 h1n1 pandemic. *PLoS One*, *6*(1), e16591. doi:10.1371/journal.pone.0016591 PMID:21304943

Balcan, D., Colizza, V., Gonçalves, B., Hu, H., Ramasco, J. J., & Vespignani, A. (2009). Multiscale mobility networks and the spatial spreading of infectious diseases. *Proceedings of the National Academy of Sciences of the United States of America*, *106*(51), 21484–21489. doi:10.1073/pnas.0906910106 PMID:20018697

Blondel, V. D., Decuyper, A., & Krings, G. (2015). A survey of results on mobile phone datasets analysis. *EPJ Data Science*, *4*(1), 1–55. doi:10.1140/epjds/s13688-015-0046-0

Bonchi, F., & Ferrari, E. (2010). *Privacy-Aware Knowledge Discovery: Novel Applications and New Techniques*. CRC Press, Inc. doi:10.1201/b10373

Brilhante, I. R., Macedo, J. A. F. D., Renso, C., & Casanova, M. A. (2011). Trajectory data analysis using complex networks. In *Symposium on International Database Engineering & Applications* (pp. 17-25). ACM.

Brockmann, D., & Helbing, D. (2013). The hidden geometry of complex, network-driven contagion phenomena. *Science*, *342*(6164), 1337–1342. doi:10.1126/science.1245200 PMID:24337289

Cho, E., Myers, S. A., & Leskovec, J. (2011). Friendship and mobility:user movement in location-based social networks. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, CA (pp.1082-1090). DBLP.

Cohen, R., Erez, K., Ben-Avraham, D., & Havlin, S. (2001). Breakdown of the internet under intentional attack. *Physical Review Letters*, *86*(16), 3682–3685. doi:10.1103/PhysRevLett.86.3682 PMID:11328053

de Montjoye, Y. A., Hidalgo, C. A., Verleysen, M., & Blondel, V. D. (2013). Unique in the crowd: The privacy bounds of human mobility. *Scientific Reports*, *3*(6), 1376. doi:10.1038/srep01376 PMID:23524645

Dobra, A., Williams, N. E., & Eagle, N. (2015). Spatiotemporal detection of unusual human population behavior using mobile phone data. *PLoS One*, *10*(3), e0120449. doi:10.1371/journal.pone.0120449 PMID:25806954

Gabrielli, L., Fadda, D., Rossetti, G., Nanni, M., Piccinini, L., & Pedreschi, D. et al.. (2018). Discovering Mobility Functional Areas: A Mobility Data Analysis Approach. In *International Workshop on Complex Networks*. Cham: Springer. doi:10.1007/978-3-319-73198-8_27

Gao, Y. L., Chen, S. M., Nie, S., Ma, F., & Guan, J. J. (2017). Robustness analysis of interdependent networks under multiple-attacking strategies. *Physica A*, 496.

Giannotti, F. (2011). Mobility, Data Mining and Privacy Understanding Human Movement Patterns from Trajectory Data. In *IEEE International Conference on Mobile Data Management* (Vol. 1, pp. 4-5). IEEE. doi:10.1109/MDM.2011.103

Giannotti, F., & Pedreschi, D. (2008). *Mobility, Data Mining and Privacy: Geographic Knowledge Discovery*. Springer. doi:10.1007/978-3-540-75177-9

Kujala, R., Aledavood, T., & Saramäki, J. (2016). Estimation and monitoring of city-to-city travel times using call detail records. *EPJ Data Science*, *5*(1), 6. doi:10.1140/epjds/s13688-016-0067-3

Li, X., Sun, Z., Cao, D., He, Z., & Zhu, Q. (2016). Real-time trajectory planning for autonomous urban driving: Framework, algorithms, and verifications. *IEEE/ASME Transactions on Mechatronics*, *21*(2), 740–753. doi:10.1109/TMECH.2015.2493980

Liu, A., Wang, W., Shang, S., Li, Q., & Zhang, X. (2017). Efficient task assignment in spatial crowdsourcing with worker and task privacy protection. *GeoInformatica*, (3): 1–28.

Louail, T., Lenormand, M., Picornell, M., Cantú, O. G., Herranz, R., & Friasmartinez, E. et al.. (2015). Uncovering the spatial structure of mobility networks. *Nature Communications*, *6*(1), 6007. doi:10.1038/ncomms7007 PMID:25607690

Louail, T., Lenormand, M., Ros, O. G. C., Picornell, M., Herranz, R., & Friasmartinez, E. et al.. (2014). From mobile phone data to the spatial structure of cities. *Scientific Reports*, *4*(2973), 5276. PMID:24923248

Monreale, A., Rinzivillo, S., Pratesi, F., Giannotti, F., & Pedreschi, D. (2014). Privacy-by-design in big data analytics and social mining. *EPJ Data Science*, *3*(1), 10. doi:10.1140/epjds/s13688-014-0010-4

Nguyen, T., & Szymanski, B. K. (2012). Using Location-Based Social Networks to Validate Human Mobility and Relationships Models. In *International Conference on Advances in Social Networks Analysis and Mining* (pp.1215-1221). IEEE Computer Society. doi:10.1109/ASONAM.2012.210

Ortale, R., Ritacco, E., Pelekis, N., Trasarti, R., Costa, G., & Giannotti, F. et al.. (2008). The DAEDALUS framework:progressive querying and mining of movement data. In *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems* (pp. 1-4). ACM.

Quang, M. N., Tai, D., Huynh, U., & Le, B. (2016). MHHUSP: An integrated algorithm for mining and Hiding High Utility Sequential Patterns. In *Eighth International Conference on Knowledge and Systems Engineering* (pp. 13-18). IEEE.

Rajesh, N., Sujatha, K., & Lawrence, A. A. (2016). Survey on Privacy Preserving Data Mining Techniques using Recent Algorithms. *International Journal of Computers and Applications*, *133*(7), 30–33.

Tsai, Y. C., Wang, S. L., Song, C. Y., & Ting, I. H. (2016). Privacy and Utility Effects of k-anonymity on Association Rule Hiding. In *Multidisciplinary International Social Networks Conference on Social informatics 2016, Data Science* (pp.1-6). ACM. doi:10.1145/2955129.2955169

Williams, N. E., Thomas, T. A., Dunbar, M., Eagle, N., & Dobra, A. (2015). Measures of human mobility using mobile phone records enhanced with gis data. *PLoS One*, *10*(7), e0133630. doi:10.1371/journal.pone.0133630 PMID:26192322

Zhang, H., Wu, C., Chen, Z., Liu, Z., & Zhu, Y. (2017). A novel on-line spatial-temporal k-anonymity method for location privacy protection from sequence rules-based inference attacks. *PLoS One*, *12*(8), e0182232. doi:10.1371/journal.pone.0182232 PMID:28767687

Zhu, Y., Zheng, G., & Fitch, M. (2018). Secrecy rate analysis of UAV-enabled mmWave networks using Matérn hardcore point processes. *IEEE Journal on Selected Areas in Communications*, *36*(7), 1397–1409. doi:10.1109/JSAC.2018.2825158

Y. Zhu, G. Zheng., & K. K. Wong. (2019). Blockchain Empowered Decentralized Storage in Air-to-Ground Industrial Networks. *IEEE Transactions on Industrial Informatics*.

Haitao Zhang is an Associate professor at the College of Geography and Biological Information in Nanjing University of Posts and Telecommunications. His current research interests are data mining, spatio-temporal reasoning and LBS privacy protection.

Yunhong Zhu is a postgraduate student in College of Telecommunications and Information Engineering in Nanjing University of Posts and Telecommunications. Her research interests are in mining the temporal characteristics of trajectory data and the provision of location-based services