A trajectory data publishing algorithm satisfying local suppression

Xiaohui Li, Yuliang Bai, Yajun Wang and Bo Li

Abstract
Suppressing the trajectory data to be released can effectively reduce the risk of user privacy leakage. However, the global suppression of the data set to meet the traditional privacy model method reduces the availability of trajectory data. Therefore, we propose a trajectory data differential privacy protection algorithm based on local suppression (TPLS) to provide the user with the ability and flexibility of protecting data through local suppression. The main contributions of this article include as follows: (1) introducing privacy protection method in trajectory data release, (2) performing effective local suppression judgment on the points in the minimum violation sequence of the trajectory data set, and (3) proposing a differential privacy protection algorithm based on local suppression. In the algorithm, we achieve the purpose Maximal frequent sequence (MFS) sequence loss rate in the trajectory data set by effective local inhibition judgment and updating the minimum violation sequence set, and then establish a classification tree and add noise to the leaf nodes to improve the security of the data to be published. Simulation results show that the proposed algorithm is effective, which can reduce the data loss rate and improve data availability while reducing the risk of user privacy leakage.

Keywords
Classification tree, differential privacy, local suppression, minimum violation sequence

Introduction
With the development of positioning technology and the popularization of intelligent devices, a large number of trajectory data of moving objects are produced, and the information contained in these trajectory data makes the in-depth study and analysis of trajectory data becomes a research hot spot in the field of data mining.2 Through the analysis and exploration of trajectory data, researchers obtain a large amount of valuable information to study the privacy protection of user information. If the security of the data is not ensured during the analysis and mining of the data, the attacker with background knowledge can infer the user’s privacy information by analyzing the trajectory data,5 resulting in the disclosure of the user’s privacy information, and threatening the user’s property security. However, if the processing of the trajectory data set is too strict, the integrity of the data set will be lost, the availability of information will be greatly reduced, and the digging value of the data set will lose, resulting in a waste of information.8,9 So how to ensure that the published trajectory data does not reveal the privacy of users while having high data availability is an important research topic.
At present, some achievements have been made in the research of privacy protection methods in trajectory data release. For example, Mohammmed et al.10 propose an LKC-privacy model suitable for radio-frequency identification (RFID) data and used an anonymous algorithm to implement the LKC-privacy model; the algorithm first identifies the minimum violation sequence set in the trajectory data set, and then global suppression of the violation sequence is carried out by the Greedy method to minimize the loss of the maximum frequent sequence. However, the global suppression method needs to remove a large amount of data, which does not effectively improve the data availability. In Ghasemzadeh et al.,11 the authors realize the data, which does not effectively improve the data availability method needs to remove a large amount of data. However, there is no detailed suppression scheme for the location point in the process of local suppression, so it will produce a new minimum violation sequence and increase the suppression cost. According to the relationship between privacy correlation degree and data utility, Jing et al.13 proposed to improve the global suppression operation of trajectory data to only suppress part of trajectory data when the trajectory data are processed anonymously. The trajectory sequence with higher frequency is processed first, so as to reduce the number of positions to be suppressed. However, this algorithm will also produce new points to be suppressed in the process, so the time complexity is relatively high.

Through the above analysis, we find that local suppression can effectively solve the problem of reducing data availability caused by global suppression, but it is possible to produce a new minimum violation sequence in the process of suppression, which may lead to the risk of data privacy disclosure again. Therefore, we need effective local suppression so that it does not produce a new minimum violation sequence in the process of data processing, which helps to improve the security and availability of data. So, in this article, we focus on improving the security of user trajectory data release process and reducing the degree of data loss due to global suppression or other ineffective local suppression and proposing a trajectory data differential privacy protection algorithm based on local suppression which satisfies the LKC-privacy model. In the algorithm, we first find out the subsequences that satisfy the effective local suppression in the frequent sequences, and then judge the effective local suppression of the points in the subsequences, avoiding the occurrence of new minimum violation sequences due to invalid inhibition judgment. The analysis results of a large number of trajectory data sets show that compared with other algorithms, the algorithm can effectively reduce data loss while ensuring the security of trajectory data release. The rest of this article is organized as follows. In section “Relevant work,” we describe some related work. In section “The basic idea of the algorithm,” we explain the proposed algorithm in two parts: one is the relevant design ideas and the other is the algorithm description. We also perform some experiment to evaluate our proposed algorithm along with two scenarios for better illustration. Finally, we conclude this article in section “Conclusion.”

### Relevant work

This section mainly describes some basic research works and related concepts, including trajectory, violation sequence, frequent sequence, classification tree, and so on.

#### Trajectory and sequence

**Definition 1 (trajectory data set).** $T$ is a collection of trace data, expressed as: $T = \{tr_1, tr_2, \ldots, tr_n\}$, where $n$ represents the number of trajectory bars contained in the $T$; it also can be expressed as $|T|$, approach $|T| = i$.

In Table 1, $d_2, b_3, c_7, e_8, \ldots$ and other symbols represent different position points in the trajectory sequence. When the same sensitive attribute and several location points frequently appear in the same trajectory in the

| Table 1. A trajectory data set $T$ containing sensitive attributes. |
|---------------------------------------------------------------|
| Trajectory | Sequence | Sensitive attributes |
| tr1        | $d_2 \rightarrow b_3 \rightarrow c_7 \rightarrow e_8$ | DIABETES         |
| tr2        | $b_3 \rightarrow c_4 \rightarrow f_5 \rightarrow e_8$ | DIABETES         |
| tr3        | $d_2 \rightarrow c_5 \rightarrow f_6 \rightarrow e_9$ | SARS             |
| tr4        | $d_2 \rightarrow f_6 \rightarrow c_7 \rightarrow e_9$ | SARS             |
| tr5        | $d_2 \rightarrow c_5 \rightarrow f_6 \rightarrow c_7 \rightarrow e_9$ | HIV              |
| tr6        | $a_1 \rightarrow a_2 \rightarrow b_3 \rightarrow c_4 \rightarrow f_6 \rightarrow e_8$ | HIV              |
| tr7        | $a_1 \rightarrow a_2 \rightarrow c_5 \rightarrow f_6 \rightarrow c_7 \rightarrow e_9$ | HIV              |

SARS: severe acute respiratory syndrome.
trajectory data set shown in Table 1, the malicious invader can infer the sensitive attribute of other users with the same location point according to this characteristic. For example, in the table, if the background knowledge obtained by $b_1, e_8$, malicious invader can infer the probability that the user may have DIABETES is 66.67%, since the $t_{r1}, t_{r2}, t_{r5}$ contains $b_1, e_8$ two location points, two of the three trajectories have sensitive attributes of DIABETES.

**LKC-privacy.** $L$ is the maximum trajectory length that the attacker holds, $K$ represents the number of anonymity, $C$ represents the confidence threshold of the anonymous set. $T$ is the trajectory data set for all users, $S$ are the sensitive attribute values in data set $T$. The condition that made $T$ satisfy the model of LKC-privacy is that when and only when, any subsequence $p$ in $T$ satisfies the following conditions when $|p| < L$:

1. $|T_p| \geq K$, $T_p$ is the user including $p$ in trajectory.
2. $\text{Conf}(s|T_p) \leq C$, $0 \leq C < 1$, $s \in S$, $\text{Conf}(s|T_p) = |T_{p(s)}|/|T_p|$. $\text{Conf}$ is the abbreviation of confidence. $C$ is the confidence threshold of the anonymous set, which can flexibly adjust the degree of anonymity according to the demand.

**Definition 2 (violation sequence).** Suppose the length of the sequence $p$ satisfy the prerequisite: $0 < |p| \leq L$, according to LKC-privacy, if the sequence $p$ does not satisfy any of the conditions in Definition 3, it is called a violation sequence.

**Definition 3 (minimum violation sequence).** If the sequence $p$ is a violation sequence and there is no violation sequence in any of its subsequences, the $p$ is called a minimum violation sequence, which is recorded as an MVS.

**Definition 4 (frequent sequences).** Given a threshold $E > 0$, if the times of sequence $p$ appear in a trajectory are greater than or equal to $E$, it is called $p$ a frequent sequence.

**Definition 5 (maximum frequent sequences).** If a sequence $p$ is a frequent sequence in the trajectory data set $T$, and any parent sequence of $p$ is not frequent sequences, it is called $p$ a maximum frequent sequence, recorded as MFS.

**Classification tree.**

**Classification trees.** For a given trajectory data set, sensitive information items in the data set are used as leaf nodes of the classification tree, and generalized leaf nodes become nodes of the classification tree, and the root nodes of the classification tree are the sets of all leaf nodes. Figure 1 shows a simple classification tree about garbage classification.

**Differential privacy**

Differential privacy protection is a privacy technology based on data distortion, using noise technology to make sensitive data distortion, but at the same time keep some data or data properties unchanged. Differential privacy protection can guarantee that adding or deleting a piece of data in a data set will not affect the query output result, so even in the worst case, all sensitive data known to the attacker except a record can still guarantee that the sensitive information of this record will not be leaked.

**Differential privacy.** There is at most one different record between the given two data sets: $T_1$ and $T_2$. Given a privacy algorithm $A$, $\text{Rang}(A)$ represents the range of $A$ values, if algorithm A satisfy the $\varepsilon$-differential privacy, it represents that any output result $(O(O \in \text{Rang}(A)))$ on data set between $T_1$ and $T_2$ satisfies the inequality

$$\Pr[A(T_1) = O] \leq \exp(\varepsilon) \cdot \Pr[A(T_2) = O] \quad (1)$$

In the inequality, $\varepsilon$ represents the privacy budget, and $\Pr[A(T_1) = O]$ represents the rate of algorithm A—it is controlled by the randomness of A (The differential privacy protection noise mechanism used in this article is the Laplace noise mechanism.).

**Definition 6 (global sensitivity).** For any function $f : T \rightarrow R^d$, the global sensitivity of function $f$ is $\Delta f$

$$\Delta f = \max_{T_1, T_2} \|f(T_1) - f(T_2)\|_p \quad (2)$$
where $R$ represents the real space of mapping, $d$ represents the query dimensions of functions, and $p$ represents the distance used for measure $\Delta f$.

**Theorem 1 (Laplace mechanism).** For any function $f : T \rightarrow R^d$, if algorithm $A$ satisfies $\varepsilon$-differential privacy, the output result of $A$ satisfies the inequality

$$A(T) = f(T) + \left( \text{Lap}_1\left(\frac{\Delta f}{\varepsilon}\right), \text{Lap}_2\left(\frac{\Delta f}{\varepsilon}\right), \ldots, \text{Lap}_i\left(\frac{\Delta f}{\varepsilon}\right) \right)$$

(3)

In the inequality, $\text{Lap}_i(\Delta f/\varepsilon)(1 \leq i \leq d)$ are the independent Laplace variables; the amount of noise is proportional to $\Delta f$, and inversely proportional to $\varepsilon$. The greater the global sensitivity of the algorithm, the greater the noise is required.

**Definition 7 (count queries).** For the given trajectory data set $T$ and sequence $q$, the count of count query $R$ to data set $T$ is $R(T) = |\{r \in T, q \in r\}|$.

**Definition 8 (average relative error).** Average relative error is used to measure the utilization of trajectory data set $\tilde{T}$ after data processing, using count query $R$

$$\text{MRE} = \frac{|R(\tilde{T}) - R(T)|}{\max\{R(T), b\}}$$

(4)

where $R(T)$ represents the count query of raw data set, $R(\tilde{T})$ represents the count query of processed data set, and $b$ is a rational constraint set to prevent the denominator from being too small.

**The basic idea of the algorithm**

In order to ensure the security of user privacy information in trajectory data set while ensuring the availability of data to be published, this article proposes a trajectory data differential privacy protection algorithm based on local suppression. In the algorithm, we use the method of effective local suppression judgment to avoid the trajectory sequence to produce a new minimum violation sequence, at the same time, we constantly update the original minimum violation sequence set in the trajectory sequence so that all trajectory sequences satisfy the LKC-privacy model, this processing effectively avoids the over-repression of information brought by global suppression and improves data availability. Second, the classification tree is based on trajectory data information and noise is added to leaf nodes to improve the security of user privacy information.

The algorithm flowchart is shown in Figure 2.

The algorithm first calculates the updated minimum violation sequence (NewMVS), and specific processes are described below. First, find out the MVS set in the trajectory data set, then find the maximum frequent sequence according to the given frequent threshold $E$, and then the MFS tree is constructed to determine the order of suppression according to the suppression priority score $\text{Score}(p)$ of point $p$

$$\text{Score}(p) = \frac{\text{Eliminate}(p)}{\text{Loss}(p)}$$

(5)

where $\text{Eliminate}(p)$ represents the number of MVSs that the inhibition point $p$ can eliminate, and $\text{Loss}(p)$ represents the loss of usefulness caused by the inhibition point. The algorithm pseudocode is as follows.

In the second part, the classification tree is constructed using the data iteratively processed by the LKC-privacy model of the trajectory data set, and the sensitive information is protected by the Laplace noise mechanism. First, initialize the data set $T$, select two groups of frequent sequences in the trajectory data set
to construct a classification tree. The idea is that according to the number of occurrences of any two points in each trajectory record, select the number of the most corresponding trajectory sequences as the first group, and then pick out the most frequent points on the trajectory of the sequence as the second group after picking out the least number of sequences from all the sequences that contain this location. According to the above method, the other trajectories are selected iteratively and put into the two groups until all the trajectories are put into the classification tree, and a classification tree is constructed, regarded as $T - \text{tree}_{[0]}$. Since the trajectory data set is constantly updated, new records are constantly added to the classification tree, and we need a given privacy budget $\varepsilon$, when the new data set $\Delta T_i$ arrives, add all the records in $\Delta T_i$ to the root node of $T - \text{tree}_{[i-1]}$ and send recursively to the disjoint subset. If some records are added to a leaf node of $T - \text{tree}_{[i-1]}$, redistribute the node’s privacy budget and continue to split the node; if some records are added to non-leaf nodes of $T - \text{tree}_{[i-1]}$, the records are processed according to the classification method of $T - \text{tree}_{[i-1]}$; if some records are empty, do the above steps for the next record until all nodes are assigned to complete, forming a new classification tree, regarded as $T - \text{tree}_{[i-1]}$. And then add Laplace noise to the leaf node. The algorithm pseudocode is as follows.

---

**Input:** raw trajectory data set $T$; the maximum length of the trajectory an attacker can master $L$; number of anonymous $K$; confidence threshold $C$; collection of sensitive values $S$; minimum violation sequence MVS

**Output:** NewMVS

1. select MVS set from $T$; 2. select MFS set from $T$ and structure an MFS tree; 3. for each $q$ in MVS $\rightarrow$ effective local suppression judgment; 4. $m = MVS; \forall p \in m; P \rightarrow$ be affected by suppressing $p$; 5. $Q =$ the violation sequence set include single point violation sequence and the point of $P$; 6. delete the points belong to $Q$ from $P$ except $p$; 7. return a new sequence; 8. then loop steps 3 to 6; 9. return score = local suppression & global suppression; 10. sheet1 = structure the score sheet by score; 11. select the point $h$ of highest score from sheet1; 12. $h =$ local suppression; 13. restrain $h$ & update MFS 14. Else: 15. restrain all instance & delete the MFSA includes $h$ 16. loop steps 11 to 15; 17. update sheet1 and return new MVS.

---

**Algorithm analysis**

The first part of the algorithm Trajectory local suppression (TLS, Steps 1–18 in the above algorithm) first identifies the minimum violation sequence (MVS) in the original trajectory data set $T$, then constructs the MFS tree according to the maximum frequent sequence set, and then determines the suppression by the suppression priority score $\text{Score}(p)$ of the point $p$. Finally, the MVS set is updated, and the time complexity is $O(|x|^2|T|)$, where $|x|$ is the dimension of the trajectory data and $|T|$ is the number of trajectories in the trajectory data set. This algorithm optimizes local suppression by updating the MVS, effectively solves the problem that global suppression leads to reduced data availability, and improves trajectory data security through local suppression.

The second part of the algorithm Trajectory classification tree node (TCN, Steps 19–43 in the above algorithm) first initializes the original trajectory data set $T$ and establishes iteratively the classification tree $T - \text{tree}_{[0]}$ by the segmentation of top-down based on frequent items, then the privacy budget should be allocated according to the number of sensitive information layers, and for the incremental data sets, it is subjected to the same iterative processing, and the privacy budget allocation structure should be adjusted according to the specific situation. The time complexity is $O(|D| \cdot |S|)$, where $D$ is the total length of the original data and the updated data, $S$ is the whole set of sensitive information. For the privacy budget during the iterative segmentation of the classification tree should be detailed segmentation in the form of Laplace mechanism. First, allocate $\varepsilon$ equally to each incremental update data set $\varepsilon_m$, then divide $\varepsilon_m$ into two parts on average, the formula is $z \rightarrow \varepsilon_m = \varepsilon_m / 2$, for the Laplace mechanisms during data iteration and adding Laplace noise to leaf nodes. According to the nature of differential privacy, the privacy budget in all data processing processes is not larger than $\varepsilon$, the algorithm satisfies differential privacy, and further ensures the security of data.

---

**Results of the experiment**

**Experiment environment and data set**

This experiment runs in Python environment and is implemented by MyEclipse integrated development software. The experiment hardware environment: processor is Intel® Core™ i7-5500U CPU 2.40 GHz, RAM is 8.0 GB, and Linux operating system is adopted in this article. The open-source data set provided by the GeoLife project of the Institute of Continental Studies, which contains 18,670 real user traces, is widely used in trace data–related research experiments.

**Measurement criteria**

Data loss is an important reference to measure the influence of an algorithm on the availability of trajectory data. This article measures from three aspects:
frequent sequence (MFS), trajectory sequence, and average relative error:22

1. MFS data loss (MFS Loss) depends on the number of MFSs in the original trajectory data set and the number of MFSs remaining in the data set after local suppression processing

\[ \text{MFS Loss} = \frac{M(T) - M(T')} {M(T)} \]  

where \( M(T) \) is the number of MFSs in the original trajectory data set, and \( M(T') \) is the number of MFSs in the data set after the local suppression processing.

2. The trajectory sequence loss (TLoss) depends on the number of sequences in the original trajectory data set and the number of sequences processed

\[ \text{TLoss} = \frac{L(T) - L(T')} {L(T)} \]  

where \( L(T) \) is the number of trajectories in the original trajectory data set, and \( L(T') \) is the number of trajectories in the data set after the local suppression processing.

3. The algorithm calculates the average relative error of data by counting query; the results will be regarded as the standards of measuring data loss.17 The error formula is shown in Definition 12.

**Results of the experiment**

In order to verify the validity of this algorithm, the TLS algorithm in this article is compared with the Trajectory Privacy-preserving based on Non-Sensitive information Analysis (TPNSA) algorithm in Jinsong et al.23 The anonymous number \( K \) and the confidence threshold \( C \) in the algorithm have different effects on the experimental results, so the experimental results of the two algorithms are compared under different values of \( K \) and \( C \):

1. Effect of different \( K \) values on data loss.

From the experimental results in Figures 3 and 4, as the \( K \) value increases, the MFS loss and the sequence loss also increase, since an increase in the \( K \) value leads...
to an increase in the MVS, that will lead to an increase in the sequence which needs to be suppressed, so the loss of data increases. Although TP-NSA algorithm has certain utility to reduce data loss, it does not optimize the MVS in the trajectory data set and does not effectively utilize the advantage of local suppression, so the TLS algorithm in this article causes less loss to the data.

2. Effect of different $C$ values on data loss.

It can be seen from the experimental results in Figures 5 and 6 that the MFS loss and the sequence loss decrease with the increase in the $C$ value, and the MFS loss and the sequence loss are both due to the decrease in the number of MVSs to be suppressed due to the increase in the $C$ value step-by-step reduction. The experimental results can be seen that whether the increasing value is $K$ or $C$, the data processing results of this TLS algorithm are always superior to those of the TP-NSA algorithm in the literature, because the TLS algorithm is based on the suppression point score priority to determine the suppression order, and reasonably update the MVS, which can better reduce the loss caused by data suppression.

For average relative error, the experimental results of Hierarchical data fusion publishing mechanism (HDFPM) in Li et al. are compared with those of TCN algorithm in this article. The experimental results show that the average relative error of the data increases with the increase in the length of the trajectory data set, but the average relative error of the data decreases with the increase in the privacy budget. The experimental results show that compared with HDFPM algorithm, the TLS algorithm used in this article can reduce the average relative error and effectively protect the user’s trajectory privacy while improving the availability of data, because the trajectory data classification tree of this algorithm is established on the basis of the processed data through local suppression, which avoids the data distortion caused by adding too much Laplace noise. The results are shown in Figure 7 and Figure 8.

Conclusion

A trajectory data differential privacy protection algorithm based on local suppression is proposed by this article. In the process of data release processing, local suppression is used instead of global suppression, which effectively solves a large amount of data waste caused by global suppression and improves the availability of published data. According to the Laplace mechanism, differential privacy protection processing of classification trees based on trajectory data information ensures the security of user privacy information. Compared with other algorithms, the proposed algorithm reduces the data loss rate to a certain extent, and in the data security aspect, the average relative error is smaller under the same noise addition state, and the data privacy is not increased at the same time with significant data distortion. Further research and analysis on data release will continue in the future.
Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported, in part, by the National Natural Science Foundation of China (no. 61802161), the Department of Education of Liaoning Province (no. JZL202015402), and the Natural Science Foundation of Liaoning Province (nos 2020-MS-291 and 2020-MS-292).

ORCID iD
Xiaohui Li https://orcid.org/0000-0003-2357-1642

References
1. Li C, Lin D, Feng B, et al. Cryptanalysis of a chaotic image encryption algorithm based on information entropy. *IEEE Access* 2018; 6: 75834–75842.
2. Jinbo X, Jun R, Lei C, et al. Enhancing privacy and availability for data clustering in intelligent electrical service of IoT. *IEEE Internet Things J* 2018; 6: 1-1.
3. Wu J, Dong M, Ota K, et al. Big data analysis-based secure cluster management for optimized control plane in software-defined networks. *IEEE Trans Netw Serv Manag* 2018; 15: 27-38.
4. Li W, Ding S and Meng J. Privacy protection scheme based on space-time relation in location service. *Commun J* 2018; 39(5): 134-142.
5. Wang X, Jiang X, Wu Y, et al. A second-order HMM trajectory prediction method based on the spark platform. *J Inform Hiding Multim Sig Process* 2019; 10(2): 346–358.
6. Waddell JM, Remias SM, Kirsch JN, et al. Utilizing low-ping frequency vehicle trajectory data to characterize delay at traffic signals. *J Trans Eng Part A Syst* 2020; 146(8): 2473–2907.
7. Zhang S, Liu Q and Wang G. Location-based approach to trajectory privacy protection. *Commun J* 2018; 38(7): 85–95.
8. Xiong J, Zhang Y, Lin L, et al. Ms-PoSW: A multi-server aided proof of shared ownership scheme for secure deduplication in cloud. *Concurr Comput Pract Exp* 2020; 32(5):4252.
9. Yafeng H, Longgang X and Meng G. Non-relational management and similarity analysis of trajectory data. *Surv Mapp Bullet* 2020; 6: 81–86.
10. Mohammmed N, Fung BCM and Debbabi M. Preserving privacy and utility in RFID data publishing. Technical Report. 6850, Concordia University, Montreal, QC, Canada, 2010.
11. Ghasemzadeh M, Fung BCM, Chen R, et al. Anonymizing trajectory data for passenger flow analysis. *Trans Res Part C Emerg Technol* 2014; 39: 63–79.
12. Chen R, Fung BCM, Mohammed N, et al. Privacy-preserving trajectory data publishing by local suppression. *Inform Sci* 2011; 231(1): 83–97.
13. Jing Z, Yuan Z, Xinghua L, et al. A method of trajectory privacy protection based on trajectory frequency suppression. *J Comput Sci* 2014; 37(10): 2096–2106.
14. Dongfan X, Tailang Z and Qian L. Capturing driving behavior Heterogeneity based on trajectory data. *Int J Model Simul Sci Comput* 2020; 11(03): 2050023.
15. Komishani EG, Abadi M and Deldar F. PPTD: preserving personalized privacy in trajectory data publishing by sensitive attribute generalization and trajectory local suppression. *Knowl-based Syst* 2016; 94(15): 43–59.
16. Shan W. A study on a random forest classification algorithm based on differential privacy. Xiangtan, China: Xiangtan University, 2019.
17. Shi X and Hu Y. Privacy protection for dynamic set value-based data publishing based on classification tree. *Comput Sci* 2017; 44(5): 120–124.
18. Bistaffa F, Farinelli A, Chalkiadakis G, et al. A cooperative game-theoretic approach to the social ridesharing problem. *Artif Intell* 2017; 246: 86–117.
19. Wangjie L, Xing Z, Guanghui C, et al. Data classification fusion publishing mechanism based on differential privacy protection. *Microcomput Syst* 2019; 40(10): 2252–2256.
20. Lan W, Lin Y and Bao Y. Differential privacy trajectory data protection method incorporating interest region. *Comput Sci Explor* 2020; 14(1): 59–72.
21. Wang E, Yang YJ, Wu J, et al. An efficient prediction-based user recruitment for mobile crowdsensing. *IEEE Trans Mob Comput* 2018; 17(1): 16–28.
22. Qingying Y, Yanfei W, Zitong Y, et al. A privacy protection algorithm for trajectory data publishing based on optimized local inhibition. *Comput Eng* 2020; 46(8): 112–118.
23. Jinsong D, Yonglong L, Qingying Y, et al. Trajectory data privacy protection release based on non-sensitive information analysis. *Comput Appl* 2017; 37(318): 190–195.