sensors

MDPI

Article

# Node Location Privacy Protection Based on Differentially Private Grids in Industrial Wireless Sensor Networks

**Jun Wang [1], Rongbo Zhu [1,*], Shubo Liu [2] and Zhaohui Cai [2]**

[1] College of Computer Science, South-Central University for Nationalities, Wuhan 430074, China; jameswang@whu.edu.cn

[2] School of Computer, Wuhan University, Wuhan 430074, China; lsb_whu@126.com (S.L.); zhcai@whu.edu.cn (Z.C.)

* Correspondence: rbzhu@mail.scuec.edu.cn

Received: 31 December 2017; Accepted: 25 January 2018; Published: 31 January 2018

**Abstract:** Wireless sensor networks (WSNs) are widely applied in industrial application with the rapid development of Industry 4.0. Combining with centralized cloud platform, the enormous computational power is provided for data analysis, such as strategy control and policy making. However, the data analysis and mining will bring the issue of privacy leakage since sensors will collect varieties of data including sensitive location information of monitored objects. Differential privacy is a novel technique that can prevent compromising single record benefits. Geospatial data can be indexed by a tree structure; however, existing differentially private release methods pay no attention to the concrete analysis about the partition granularity of data domains. Based on the overall analysis of noise error and non-uniformity error, this paper proposes a data domain partitioning model, which is more accurate to choose the grid size. A uniform grid release method is put forward based on this model. In order to further reduce the errors, similar cells are merged, and then noise is added into the merged cells. Results show that our method significantly improves the query accuracy compared with other existing methods.

**Keywords:** location; privacy guarantee; differential privacy; industrial wireless sensor networks

## 1. Introduction

Recent years have witnessed the rapid development of the industrial wireless sensor networks (IWSNs), which have been introduced into the industry area to meet requirements of higher flexibility and market share, and IWSNs are becoming the key and fundamental technology of Industry 4.0 [1]. In the industrial domain, mobile nodes are used in industrial systems incrementally [2]. Radio modules or wireless nodes have been installed on mobile devices to raise mobility and flexibility which are ignored in traditional WSNs [3]. IWSNs generally contain more moving nodes, such as mobile products, workers and other mobile devices [4]. The centralized cloud platform collects sensor data to provide strategy control and policy making. However, the data analysis and mining will bring the issue of privacy leakage since a semi-credible cloud server is curious about sensitive location information of monitored objects [5]. To address this problem, a novel privacy protection technique called differential privacy [6] has been introduced to location privacy preservation.

The Location information is called geospatial data [7]. For example, as shown in Figure 1, the location information of nodes will be collected and uploaded to the cloud. The release of static geospatial data brings great convenience to scientific research. The work in [8] indicates that it is possible to use geospatial information to forecast the spread of an infectious disease. However, data analysis also has the risk of privacy leaks. For instance, De Montjoye demonstrated that only simple date and location information of four shopping records can recognize more than 90% individuals in dataset [9]. The raw data must be sanitized before release for data analysis and mining [10].

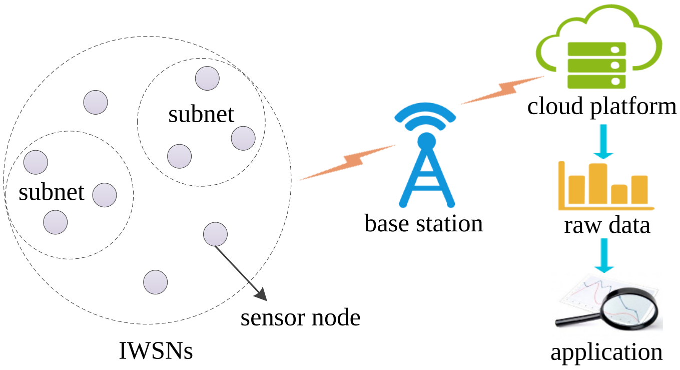


**Figure 1.** Illustration of node location data aggregation.

Compared with wired networks, IWSNs have a higher possibility of privacy leaks when they integrate with cloud computing and big data technologies. Consequently, effective privacy-preserving technologies are needed in IWSNs. The work in [11] analyses existing privacy protection approaches in WSNs from data and sensor [12,13], and it investigates the approaches, such as anonymization [14], supported in large-scale industrial environments. The widespread use of geospatial data should be coupled with greater security [15], such as security controllability and strictly provable security [16]. Traditional methods based on anonymity model have risks of privacy disclosure [17–19]. Differential privacy is a relatively novel concept [20,21] and can hide any single record in the output by perturbing the data, which makes an adversary fail to infer the presence of a record with high probability [22,23]. How to balance between data quality and privacy level is a key problem for research [24].

In order to improve the utility of released geospatial data, a number of data decomposition methods based on tree structure were proposed [25]. In [26–29], a domain partition scheme based on quadtree or *kd*-tree is proposed to enhance the utility of released data. However, these works have not analyzed optimal partition granularity of data domains, so how to choose the appropriate tree depth or partition granularity is the key point for data decomposition strategy.

To the best of our knowledge, the work in [7] first uses a granularity partitioning scheme to create differentially private geospatial data. It assumes that the shape of query is square and the non-uniformity error is proportional to the number of location points in the cells, which fall on the border of the query shape [7]. However, the shape of data query perhaps is rectangle in actuality, i.e., the length is not equal with the width of query. Meanwhile, the larger the intersection between the border of query rectangle and cells gets, the bigger the non-uniformity error becomes. Inspired by this, we point out that the non-uniformity error is proportional to the intersection area between the border of query rectangle and cells, and a novel granularity partitioning model of data domain based on the global analysis of noise error and non-uniformity error is proposed. A bucket sort based cells merging strategy is adopted to enhance the utility of released data further, and it merges all of the similar cells contained in the data domain and differs from the scheme in [29], which just aggregates the four sub nodes of quadtree.

The contributions of this paper are as follows:

- We propose a novel granularity partitioning model of data domain, which is effective to balance the noise error and the non-uniformity error. The partition granularity is proportional to the area of data domain, privacy budget and coefficient $k$.
- We adopt a cells merging strategy based on bucket sort, which groups all the similar cells of data domain into a partition, in order to decrease the noise added to each cell. This merging strategy further raises the accuracy of query.

- We conduct evaluations using two real-world datasets to verify the effectiveness of the granularity partitioning model and the similar cells merging strategy, and results show that the proposed approach has better query accuracy and enhances the utility of released data.

The rest of this paper is organized as follows: Section 2 introduces the related work about differentially private data release, including privacy spatial decomposition (PSD) method and differentially private grids approach; Section 3 provides the preliminaries on differential privacy and problem definition; Section 4 presents the granularity partitioning model and the corresponding data release approaches; Experimental results and analysis are presented in Section 5; The conclusions and future work are finally presented in Section 6.

## 2. Related Work

The work in [30] studies the network isolation problem in group-based IWSNs, and Ref. [31] researches the dangerous area of toxic gases with WSNs. These works mostly focused on the effectiveness of network and the safety of application. As cloud and communication technologies (such as 5G and Internet) are integrated into IWSNs, more private data and information are produced. Researchers and experts are facing the serious problem of how to mine useful information from perturbed data, so data validity and privacy should be considered deeply.

In order to enhance the utility of perturbed data, PSD based on tree structure is adopted [26]. The PSD divides the geospatial dataset into $m \times m$ independent cells via horizontal and vertical lines, and gets the number of points in each cell. There are two types of error in query: one is noise error introduced by perturbation, and another is non-uniformity error generated by assuming that the location points are distributed uniformly. These two errors both affect the accuracy of query results and depend on the partition granularity $m$. The finer value of $m$ implies a fewer non-uniformity error.

Based on the *kd*-tree structure, Xiao et al. [27,28] allocated half of the privacy budget to the raw data and constructed the *kd*-tree by using the sanitized data to guarantee user privacy. However, it leads to a relatively larger error [26].

Cormode et al. [26] proposed a quadtree based method Quad-opt to enhance the utility, which is called Qopt for short in this paper. Qopt splits the data domain into a complete quadtree with a predefined tree depth. Firstly, data space is divided into equal quadrants. Then, the subspace is further split into four equal pieces until tree depth reaches the predefined value $h$. Different from the existing uniform budgeting strategy, Cormode et al. proposed a novel geometric budgeting strategy, which allocates $\varepsilon_i$ for each level of the quadtree ($\sum_{i=0}^{h} \varepsilon_i = \varepsilon$). Notably, proportional factor is $2^{1/3}$ and constrained inference [32] is employed to post-process the output of query with the purpose of higher utility. However, directly adding noise into each cell will result in a larger error when the data is sparse.

Fan et al. [29] proposed to aggregate similar cells into a partition to overcome the data sparsity issue. First, each cell is pre-classified by domain knowledge, i.e., the cell is sparse or dense type. Next, a node is split into four equal quadrants until the predefined depth value $h$ is reached or the node is homogeneous, i.e., all the cells within the node belong to the same type. This method needs to pre-classify the type of cells relying on the specialized knowledge, which may lead to misjudgment. Meanwhile, merging of cells just restricts in quadtree node and fails to extend to the entire data domain.

For all above methods based on PSD, the utility of perturbed data is related to tree depth $h$ or partition granularity $m$, so how to choose the right value of $h$ or $m$ is the key point. Qardaji et al. [7] proposed a granularity partitioning model of data domain to solve this problem and presented a uniform method UG based on this model. The partition granularity is $\sqrt{N\varepsilon/c}$, where $N$ is the number of data points in cell, $c$ is a small constant (generally $c$ = 10), and $\varepsilon$ is the privacy budget. Qardaji et al. further proposed an adaptive grid method AG. It splits the data domain into $m_1 \times m_1$ independent cells, where $m_1 = max(10, 0.25 \times \lceil N\varepsilon/c \rceil)$ . The cell will be further divided into $m_2 \times m_2$ independent cells if the noisy count of the cell is bigger than the given threshold value, where $m_2 = \lceil 2N'(1-\xi)\varepsilon/c \rceil$, and $\xi$ is a parameter determined by user. To et al. [33] adopted the AG method to solve the spatial crowdsourcing specific requirements, and they modified the parameter $c$ to decrease the system overhead, which is just beyond our research. Note that literature [7] assumes that the length and width of query rectangle are equal to each other, which may affect the utility of perturbed data.

In this paper, the proposed method Ugrid adopts a novel granularity partitioning model of data domain. There is no need to assume that the length and width of query rectangle are equal to each other, and the partition granularity is $\lceil \sqrt{4kHL\varepsilon/\sqrt{2}} \rceil$ , where $k$ is the proportionality coefficient, $H$ and $L$ are the width and length of data domain, respectively, and $\varepsilon$ is the privacy budget. We further introduce a merging grids release approach, which groups all of the similar cells into a partition and adds Laplace noise to each partition. The aggregation of the similar cells has improved the accuracy of data query.

## 3. Preliminaries

In this section, we formally introduce the basic concept of differential privacy and present the problem definition.

### 3.1. Differential Privacy

The formalized definition of differential privacy is as follows.

**Definition 1.** *Differential privacy [6]: Given two neighboring datasets $D$ and $D'$, which differ at most one tuple, and a randomized algorithm $A : \mathcal{D} \to R$. Let $O$ be the set of all possible outputs of algorithm $A$, and $A$ is said to satisfy $\varepsilon$-differential privacy for any subset $\sigma \subseteq O$ if*

$$\frac{Pr[A(D) \in \sigma]}{Pr[A(D') \in \sigma]} \leq exp(\varepsilon).$$

The parameter $\varepsilon$ is called privacy budget, which is used to control the ratio of output of algorithm $A$ in neighboring datasets $D$ and $D'$ [25]. Smaller $\varepsilon$ yields a stronger privacy guarantee because the output probabilities of algorithm $A$ in $D$ and $D'$ are approximately the same, which makes the adversary fail to judge whether the tuple is present in the dataset or not. However, the smaller $\varepsilon$ is, the lower the utility will be, as the adding noise is bigger. Detail of noise addition can be seen in the part of the Laplace mechanism. The value of $\varepsilon$ is usually small, such as 0.1, 0.5, 1, etc. [25].

Differential privacy owns a significant composable property, which plays an important role in demonstrating whether an algorithm satisfies $\varepsilon$-differential privacy or not.

**Property 1.** *Parallel composition [34]: Let random algorithm $A_i$ each provide $\varepsilon_i$-differential privacy for disjoint subsets $D_i$, and the sequence of $A_i(D_i)$ provides $max(\varepsilon_i)$- differential privacy.*

**Property 2.** *Sequential composition [34]: Let random algorithm $A_i$ each provide $\varepsilon_i$-differential privacy. Then, a sequence of $A_i(D)$ over the database $D$ provides $\Sigma\varepsilon_i$- differential privacy.*

Laplace mechanism is a differentially private implementation scheme, which masks the real data by adding random noise following Laplace distribution to the output. The value of noise is related to privacy budget $\varepsilon$ and global sensitivity.

**Definition 2.** *Global sensitivity [6]: Given a function $f : \mathcal{D} \to R^d$, the global sensitivity of $f$ is defined as follows:*

$$\tau(f) = \max_{D,D'} ||f(D) - f(D')||_1.$$

The parameter $D$ and $D'$ are neighboring datasets; $R$ is the real space; $d$ is the dimension; and $||f(D) - f(D')||_1$ is the first-order norm distance [25]. For instance, the global sensitivity of count function is 1.

**Definition 3.** *Laplace mechanism [35]: Given a dataset D and a function* $f : \mathcal{D} \to R^d$*, if the adding noise follows Laplace distribution, i.e., noise* $\sim Lap(\tau(f)/\varepsilon)$*, where location parameter is 0, scale parameter is* $\tau(f)/\varepsilon$*; then, random algorithm* $A(D) = f(D) + noise$ *provides ε-differential privacy.*

The exponential mechanism addresses the non-numeric case in which adding noise makes no sense. It is another method to construct differentially private algorithm over any quality function $u(D, r)$.

**Definition 4.** *Exponential mechanism [36]: Given a dataset D, a privacy parameter ε, a quality function* $u(D, r)$*, and the global sensitivity* $\tau(u)$ *of* $u(D, r)$*, random algorithm A provides ε-differential privacy if algorithm A chooses an outcome r from the range R with probability*

$$A(D, u) = \{r : |Pr[r \in R] \propto exp(\frac{\varepsilon}{2\tau(u)} u(D, r))\}.$$

Let $Lap(b)$ be a Laplace distribution, where location parameter $\mu = 0$, scale parameter is $b$, and its probability density function is $p(x) = exp(-|x|/b)/2b$. According to the function $p(x)$, the bigger $b$ gets, the bigger perturbed noise becomes.

If $noise \sim Lap(b)$, let $\sigma(x)$ denotes standard deviation, $D(x)$ denote variance, $\sigma(x) = \sqrt{D(x)}$, $D(x) = 2b^2$, and $b = \tau(f)/\varepsilon$; then, $D(x) = 2b^2 = 2\tau(f)^2/\varepsilon^2$, $\sigma(x) = \sqrt{D(x)} = \sqrt{2\tau(f)^2/\varepsilon^2} = \sqrt{2}\tau(f)/\varepsilon$ [25].

### 3.2. Problem Definition

Let $L$ and $H$ be the domain length and width of geospatial dataset $D$ of monitored objects in IWSNs; $a$ and $b$ are the length and width of data query $Q$ as shown in Figure 2. Splitting the data domain into $m \times m$ cells $\{c_1, c_2, ..., c_i, ..., c_{m\times m}\}$, point count $x_i$ of cell $c_i$ is perturbed by random noise following Laplace distribution. Then, the perturbed differentially privacy dataset $\widetilde{D} = \{\widetilde{x}_1, \widetilde{x}_2, ..., \widetilde{x}_i, ..., \widetilde{x}_{m\times m}\}$ is released.

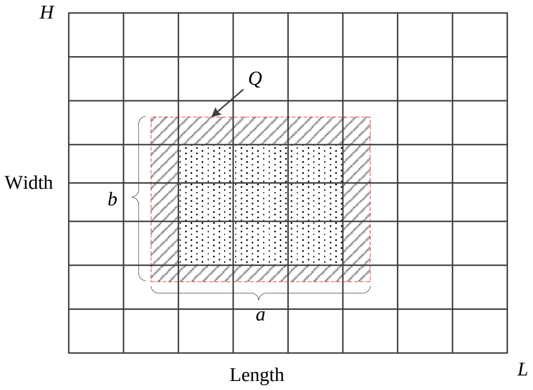


**Figure 2.** Example of data query.

Let $e_n$ be the noise error introduced by the addition noise, and let $e_u$ be the non-uniformity error caused by the assumption of uniform distribution. As shown in Figure 2, there is an intersection between cells and query rectangle $Q$, and some cells are partly contained in it, such as cells filled with oblique lines. For these cells, we calculate the number of data points in the intersected part based on the uniformity assumption. For instance, let $I_i$ be the intersected part between cell $c_i$ and query $Q$, and let $x_i'$ be the estimated count of data points in $I_i$ . Then, $x_i' = x_i \times area(I_i)/area(c_i)$, where $x_i$ is the count of data points in $c_i$, $area(I_i)$ is the area of $I_i$, and $area(c_i)$ is the area of $c_i$. We conclude that $x_i' = x_i$ if $area(I_i) = area(c_i)$, $x_i' \neq x_i$ if not.

Intuitively, a bigger value of partition granularity implies a smaller non-uniformity error and a larger noise error when splitting the data domain. In contrast, a smaller value of partition granularity means a smaller noise error and a larger non-uniformity error because the value of partition granularity $m$ is the key. Qardaji et al. inferred the value of $m$ based on the assumption $a = b$ [7], which is needless in this paper.

From the above analysis, given a geospatial dataset $D$ and privacy budget $\varepsilon$, how to choose an optimal partition granularity $m$ to minimize the error of query $Q$ is the research question in this paper. The formalized definition is defined as follows:

$$\min_{m}(e_n + e_u).$$

Table 1 is a summary of the primary symbols used in this paper.

**Table 1.** Symbols.

| Symbol | Description |
|---|---|
| $D$ | two-dimensional geospatial dataset |
| $\widetilde{D}$ | sanitized dataset |
| $c_i$ | cell |
| $x_i$ | count of data points in cell $c_i$ |
| $k$ | proportionality coefficient |
| $Q$ | data query |
| $I_i$ | intersection between query $Q$ and cell $c_i$ |
| $\widetilde{x}_i$ | noisy count of cell $c_i$ |
| $L$ | domain length of $D$ |
| $H$ | domain width of $D$ |
| $a$ | length of $Q$ |
| $b$ | width of $Q$ |

## 4. Sanitized Data Release Based on Grid Partition

This section mainly states the sanitized data release approach Ugrid and Mgrid. In particular, the granularity partitioning model of data domain is first presented; then, Ugrid and Mgrid based on this model are presented.

### *4.1. Uniform Grid Release Approach*

#### 4.1.1. Granularity Partitioning Model of Data Domain

We give the optimal value of partition granularity $m$ based on the overall analysis of perturbed data's noise error and non-uniformity error. Details of derivation are as follows.

As illustrated in Figure 2, let cells that are filled with oblique lines be $I$, the non-uniformity error is 0 when the area of $I$ is 0; now, cells in $Q$ are completely contained in it. The non-uniformity error becomes bigger with the area of $I$ increasing when cells in $Q$ are not contained in it. Motivated by this, we propose that the relative error $\beta$ based on the uniformity assumption is proportional to the area $\alpha$ of $I$. Let $k$ be the proportionality coefficient between $\beta$ and $\alpha$, $\beta_i$ and $\alpha_i$ are the $i$th sampling values; let $\widetilde{\beta}$ and $\widetilde{\alpha}$ be the mean value of relative error and area of $I$, and we can infer the value of $k$ through linear-regression analysis, and the value of $k$ can be calculated by least square estimation; the formula is as follows:

$$k = \sum(\alpha_i - \bar{\alpha})(\beta_i - \bar{\beta}) \Big/ \sum(\alpha_i - \bar{\alpha})^2.$$

For instance, we have that relative error satisfies $\beta = 0.1291\alpha + const$ through linear-regression analysis in checkin dataset, where $k = 0.1291$. Notably, the relative error is 0 when the area of $I$ is 0. We finally set $k = 0.1314$ after using the coordinate origin (0, 0) to correct the value of $k$. Details of checkin dataset can be seen in Section 5.

The bigger the area of $I$ gets, the bigger the non-uniformity error becomes. The value of the non-uniformity error $e_u$ reaches a maximum in theory, and the area of $I$ is the region of cells that intersected with the border of the query $Q$. The value of $e_u$ is $k(2aH + 2bL)/m$.

The analysis of non-uniformity error: the number of cells that intersected with the border of $Q$ is defined as $Num = 2am/L + 2bm/H$, where $2am/L$ is the number of cells that intersected with the top and bottom borders of $Q$, and $2bm/H$ is the number of cells intersected with the left and right borders of $Q$. The area of each cell is $LH/m^2$, where $LH$ is the total area of data domain and $m^2$ is the total number of cells. Then, we deduce the total area of the cells that intersected with the border of $Q$ is $Num \times LH/m^2 = (2am/L + 2bm/H) \times LH/m^2$, and non-uniformity error that is proportional to the total area is $k(2am/L + 2bm/H) \times LH/m^2$.

The noise error is caused by the added Laplace noise and is affected by the value of partition granularity $m$. The value of $e_n$ is $\sqrt{2}rm/\varepsilon$, and $r = ab/LH$.

The analysis of noise error: the added random noise follows Laplace distribution and has a standard deviation $\sqrt{2}/\varepsilon$. The number of cells included in the query $Q$ is $Num' = (ab/LH)m^2$, where $ab/LH$ is the ratio of the area of query $Q$ to the area of data domain and $m^2$ is the total number of cells. Thus, the standard deviation of the total noise error is $\sqrt{2Num'}/\varepsilon = \sqrt{2(ab/LH)m^2}/\varepsilon = \sqrt{2}rm/\varepsilon$, and $r = ab/LH$.

**Lemma 1.** *The total error of data query is at a minimum when $m = \lceil \sqrt{4kHL\varepsilon/\sqrt{2}} \rceil$, where k is the proportional coefficient, H and L are the domain width and length of dataset D, and ε is the privacy budget.*

**Proof.** The total error is the sum of non-uniformity error $e_u$ and noise error $e_n$, where $e_u = k(2am/L + 2bm/H) \times LH/m^2 = k(2aH + 2bL)/m \geq 2k\sqrt{4aH \times bL}/m = 4k\sqrt{HL} \times \sqrt{rHL}/m = 4kHL\sqrt{r}/m, e_n = \sqrt{2}rm/\varepsilon$, according to the above analysis. To minimize the total error, according to $4kHL\sqrt{r}/m = \sqrt{2}rm/\varepsilon$, we deduce that $m = \sqrt{4kHL\varepsilon/\sqrt{2}}$, and round it up to a whole number; then, $m = \lceil \sqrt{4kHL\varepsilon/\sqrt{2}} \rceil$. □

4.1.2. Ugrid Method

We propose a uniform grid release approach based on this granularity partitioning model. First of all, Ugrid splits the data domain into $m \times m$ independent cells according to the value of partition granularity $m$, where $m = \lceil \sqrt{4kHL\varepsilon/\sqrt{2}} \rceil$. Next, calculate the count $x_i$ of data points in each cell $c_i$ through a traversal of $D$ and getting the point counts of all cells $\{x_1, x_2, ..., x_i, ..., x_{m\times m}\}$. Then, obtain the noised count $\widetilde{x}_i = x_i + Lap(1/\varepsilon)$ by adding Laplace noise $Lap(1/\varepsilon)$. Finally, share the sanitized dataset $\widetilde{D} = \{\widetilde{x}_1, \widetilde{x}_2, ..., \widetilde{x}_i, ..., \widetilde{x}_{m\times m}\}$ for query services.

The pseudo code description of Ugrid is presented in Algorithm 1. Steps $1 \sim 5$ of Ugrid conduct domain division in geospatial dataset $D$; step 6 sets point count $x_i = |c_i|$; in steps $7 \sim 9$, the Laplace noise is added into each count $x_i$ of $c_i$; and the last step generates the differentially privacy sanitized dataset $\widetilde{D}$.

**Algorithm 1** Ugrid

```
Input: geospatial dataset $D$, privacy budget $\varepsilon$, partition granularity $m$
Output: sanitized dataset $\widetilde{D} = \{\widetilde{x}_1, \widetilde{x}_2, ..., \widetilde{x}_i, ..., \widetilde{x}_{m\times m}\}$
1:  for $(o = 1; o \leq D.size(); o++)$ do
2:      if $(point_o \in c_i)$ then
3:          add $point_o$ to cell $c_i$
4:      end if
5:  end for
6:  set point count $x_i = |c_i|$
7:  for $(i = 1; i \leq m \times m; i++)$ do
8:      noisy count $\widetilde{x}_i = x_i + Lap(1/\varepsilon)$
9:  end for
10: sanitized dataset $\widetilde{D} = \{\widetilde{x}_1, \widetilde{x}_2, ..., \widetilde{x}_i, ..., \widetilde{x}_{m\times m}\}$
```

**Theorem 1.** *Algorithm Ugrid satisfies $\varepsilon$-differential privacy.*

**Proof.** According to Property 1 parallel composition, Laplace noise $Lap(1/\varepsilon)$ is added into $m \times m$ independent cells; then, Ugrid provides $max(\varepsilon_i)$ differential privacy. As $\varepsilon_i = \varepsilon$, Ugrid satisfies $\varepsilon$-differential privacy. □

**Theorem 2.** *Given geospatial dataset D and partition granularity m, the time complexity of Ugrid is $O(|D| + m^2)$.*

**Proof.** In steps 1–5 of algorithm, dataset $D$ has $|D|$ data points, and the cost is $|D|$. In steps 7–9, there are $m^2$ cells, and the cost is $m^2$. Therefore, the total cost is $O(|D| + m^2)$. □

To balance the noise error and non-uniformity error, Ugrid sets $m = \lceil \sqrt{4kHL\varepsilon/\sqrt{2}} \rceil$ to minimize the total error. However, the output of query may contain a large mass of noise when some cells are extremely sparse. For example, let real count $x_i$ of cell $c_i$ be 1, the added noise is 20, and then the noise error is 20, which remains with little valuable information in it. In order to decrease data query error and enhance the utility of perturbed data, we further introduce a merging grids release approach Mgrid. It aggregates all similar cells into a partition employing the bucket sort based cell merging strategy, which reduces the noise error by adding noise into each partition.

## *4.2. Merging Grids Release Approach*

### 4.2.1. Merging of Similar Cells

The formalized definition of similar cells merging is defined as follows:

**Definition 5.** *Similar cells merging: Given the point count of all cells $\{x_1, x_2, ..., x_i, ..., x_{m \times m}\}$, they are similar if the count of these cells:*

$$|x_i.hash - x_j.hash| < c, i, j \in N^*.$$

The parameter $c$ is a constant, and corresponding cells will be merged into a partition if the difference between every two hash value of these cells is smaller than the given threshold $c$. BKDRHash maps the binary string(3-bits are a group) of $x_i$ to $x_i.hash$.

To find the similar cells, clustering algorithm affinity propagation [37] has the advantage of not needing to specify the cluster "number". However, the algorithm is more complex, and the time complexity is $O(m^4 \log m^2)$. To enhance the efficiency, the bucket sort based similar cells merging is adopted. Given mapping function $f(x_i) = x_i.hash/c, i \in [1, m^2]$, $c$ is a small value, and is related to the BKDRHash. Mapping each cell with count $x_i$ to the corresponding bucket, the cells in each bucket just are similar cells. Bucket sort based similar cells merging traverses all cells only just once, and its time complexity is $O(m^2)$.

After the merging of similar cells, we get the partition dataset $\{p_1, p_2, ..., p_l, ..., p_L\}$. As shown in Figure 3, cells filled with point symbols, oblique lines, diagonal grids, horizontal and vertical grids are merged into the partitions $p_1$, $p_2$, $p_3$, $p_l$, respectively. The random noise $Lap(1/\varepsilon)$ is added to each partition $p_l$; then, the noise added to each cell contained in partition is decreased. Note that BKDRHash is used in the process of similar cells merging to protect data privacy of the real count of all cells.

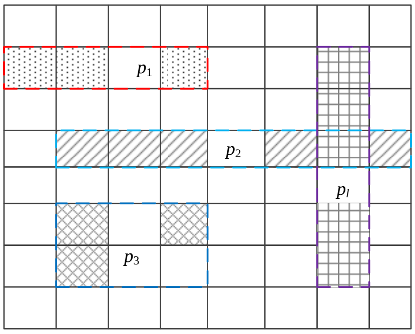


**Figure 3.** Example of merging cells.

### 4.2.2. Mgrid Method

We further propose the merging grids release approach based on the granularity partitioning model and the similar cells merging strategy. Similar to the method Ugrid, firstly, Mgrid splits the dataset into $m \times m$ cells $\{c_1, c_2, ..., c_i, ..., c_{m\times m}\}$ based on the granularity $m$. Secondly, it traverses $D$ and calculates the hash value $x_i.hash$ of $x_i, i \in [1, m^2]$ based on the BKDRHash. Thirdly, similar cells are selected according to the mapping function $f$. Furthermore, Laplace noise $Lap(1/\varepsilon)$ is added to each partition $p_l, l \in [1, L]$, and noisy count $\widetilde{x}_i$ of each cell $c_i$ in $p_l$ is defined as $\widetilde{x}_i = (|p_l| + Lap(1/\varepsilon))/p_l.size()$, where $|p_l|$ is the count of data points located in $p_l$. Finally, the sanitized dataset $\widetilde{D} = \{\widetilde{x}_1, \widetilde{x}_2, ..., \widetilde{x}_l, ..., \widetilde{x}_L\}$ is released for query services.

Algorithm 2 states the pseudo code of Mgrid. Steps 1–5 of the Mgrid conduct the cell division in geospatial dataset $D$; in steps 7–9, each count $x_i$ is mapped to corresponding bucket according to the mapping function $f(x_i) = x_i.hash/c$; step 10 gets the partition dataset $\{p_1, p_2, ..., p_l, ..., p_L\}$ after the merging of similar cells; in steps 11–13, the Laplace noise is added into each partition $p_l$; the final step generates the sanitized differentially privacy dataset $\widetilde{D}$.

**Algorithm 2** Mgrid

```
Input: geospatial dataset $D$, privacy budget $\varepsilon$, partition granularity $m$, threshold $c$
Output: sanitized dataset $\widetilde{D} = \{\widetilde{x}_1, \widetilde{x}_2, ..., \widetilde{x}_l, ..., \widetilde{x}_L\}$
 1: for $(o = 1; o \leq D.size(); o++)$ do
 2:     if $(point_o \in c_i)$ then
 3:         add $point_o$ to cell $c_i$
 4:     end if
 5: end for
 6: set point count $x_i = |c_i|$
 7: for $(i = 1; i \leq m \times m; i++)$ do
 8:     select similar cells by mapping function $f(x_i) = x_i.hash/c$
 9: end for
10: set partition dataset is $\{p_1, p_2, ..., p_l, ..., p_L\}$, where $p_l$ = {similar cells}
11: for $(l = 1; l \leq L; l++)$ do
12:     noisy count $\widetilde{x}_i = |p_l| + Lap(1/\varepsilon)$
13: end for
14: sanitized dataset $\widetilde{D} = \{\widetilde{x}_1, \widetilde{x}_2, ..., \widetilde{x}_l, ..., \widetilde{x}_L\}$
```

Mgrid also satisfies $\varepsilon$-differential privacy, and the proof is similar to Ugrid's. Note that the Laplace noise is added to each partition, bucket sort does not consume the privacy budget, and the Laplace mechanism consumes $\varepsilon$.

**Theorem 3.** *Data query error is less than or equal to the value before the merging of similar cells.*

**Proof.** Given the geospatial dataset, before the merging of similar cells, the noise error $e_n$ of query $Q$ is $\sum_{i=1}^{\sqrt{rm}} \sqrt{2}/\varepsilon$, and non-uniformity error $e_u$ is $k(2am/L + 2bm/H) \times LH/m^2 = k(2aH + 2aL)/m$, the data query error is then $error = e_n + e_u = \sum_{i=1}^{\sqrt{rm}} \sqrt{2}/\varepsilon + k(2aH + 2bL)/m$.

After the merging of similar cells, the noise error of query $Q$ is $e'_n = \sum_{i=1}^{\sqrt{rm}} \sqrt{2}/\varepsilon p_l.size()$, $l \in [1, L], p_l.size() \geq 1$, and non-uniformity error is $e'_u = k(2am/L + 2bm/H) \times LH/m^2 = k(2aH + 2aL)/m$; then, the data query error is $error' = e'_n + e'_u = \sum_{i=1}^{\sqrt{rm}} \sqrt{2}/\ \varepsilon p_l.size() + k(2aH + 2bL)/m$.

The difference between $error'$ and $error$ is $(\sum_{i=1}^{\sqrt{rm}} \sqrt{2}/\ \varepsilon)(1/p_l.size() - 1) \leq 0$, so we deduce that data query error is less than or equal to the value before the merging of similar cells. □

**Theorem 4.** *Given geospatial dataset D and partition granularity m, then the time complexity of Mgrid is* $O(|D| + m^2 + L)$.

**Proof.** In steps 1–5 of algorithm, dataset $D$ has $|D|$ data points, and the cost is $|D|$. There are $m^2$ cells in steps 7–9, and the cost is $m^2$. In steps 11–13, the number of all partitions is $L$. Therefore, the total cost is $O(|D| + m^2 + L)$.

## 5. Experimental Results and Analysis

In this section, we begin with the introducing of the metric standard of perturbed data. Then, we present the experimental datasets used in this paper. Finally, detailed analysis of experimental results is presented.

### 5.1. Utility Metric

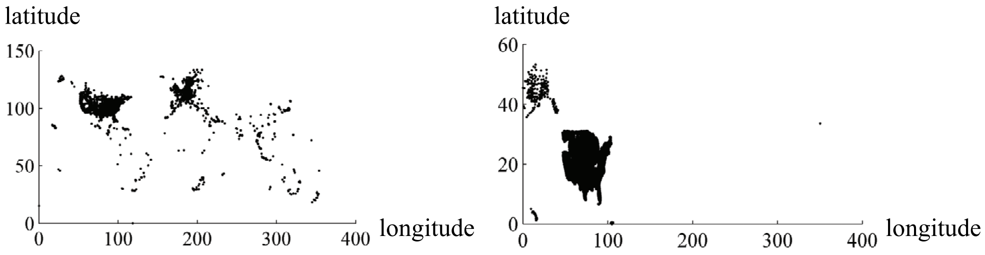
There is no specific utility metric standard in existing literature for differential privacy, and the existing methods usually adopt variance [26,38], relative error [7,26,29], absolute error [7], etc. to evaluate the utility of sanitized data. In this paper, we adopt relative error and absolute error to estimate the utility of sanitized data. Given a query $Q$, let $Q(D)$ denote the real query result, let $Q(\widetilde{D})$ denote the noisy query result, and then the relative error is defined as:

$$Error(Q) = \left|Q(D) - Q(\tilde{D})\right| \Big/ \max\{Q(D), |D| \big/ 4^6\}.$$

The parameter $|D|$ is the count of nodes contained in dataset $D$, and the divisor is $|D|/4^6$ when $Q(D) = 0$, which avoids dividing by zero. The smaller the relative error is, the more accurate the query will be. Meanwhile, in order to intuitively observe the value of added noise, we also present the comparison results of absolute error.

### 5.2. Experimental Datasets

In Figure 4, the shapes of datasets are presented by plotting the point of monitored object directly, and the coordinate of base point starts from (0, 0) for intuition and convenience.



(a) Checkin dataset (b) Areas dataset

**Figure 4.** Illustration of datasets.

The first dataset is named as Checkin [39] obtained from a location-based social networking website Gowalla as illustrated in Figure 4a. It mainly includes the location point, check-in-time, and location id with about 6,442,890 records and only part of the location point information is used. The length and width of the data query $Q$ increase by 10 in this dataset.

The second dataset Areas [40] is obtained from the U.S. Census Bureau as illustrated in Figure 4b. Dataset Areas is the legislative areas national geodatabase, and we only employ the location point data. The length of $Q$ increases by 3 and the width increases by 2 in this dataset.

The query error is verified under different privacy budgets, such as 0.1, 0.5 and 1. Table 2 presents the parameters information about these datasets.

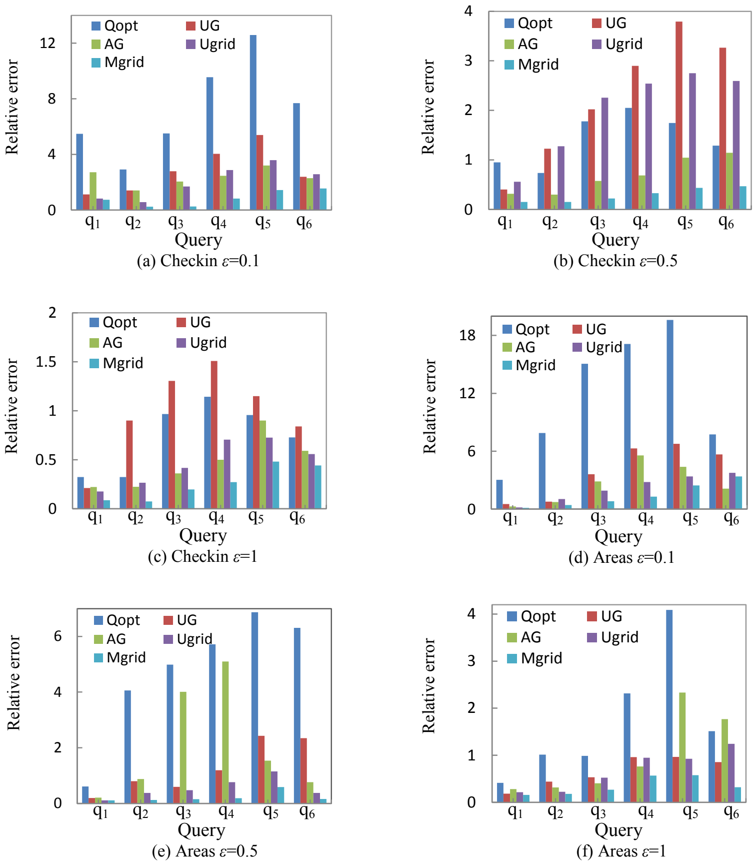
**Table 2.** Parameter information about datasets.

| Dataset | Num of Points | Domain Size | Query Size |
|---|---|---|---|
| Checkin | 625,123 | 354 × 133 | $q_i = 10(i+1) \times 10(i+1), i \in [1,6]$ |
| Areas | 179,371 | 351 × 54 | $q_i = 3(i+5) \times 2(i+5), i \in [1,6]$ |

We randomly generate 500 data queries for each query size and finally calculate the mean value. The experiments were conducted on Intel i5 CPU (santa clara, CA, USA) with 3 GB RAM, the programming platform is Eclipse3.5 (Eclipse Foundation, Inc., Ottawa, Canada), and the programming language is JAVA.

### 5.3. Experimental Results

This subsection presents the experimental results of UG [7], AG [7], Ugrid, Mgrid, and Qopt under different query $Q$.



(a) Checkin ε=0.1

(b) Checkin ε=0.5

(c) Checkin ε=1

(d) Areas ε=0.1

(e) Areas ε=0.5

(f) Areas ε=1

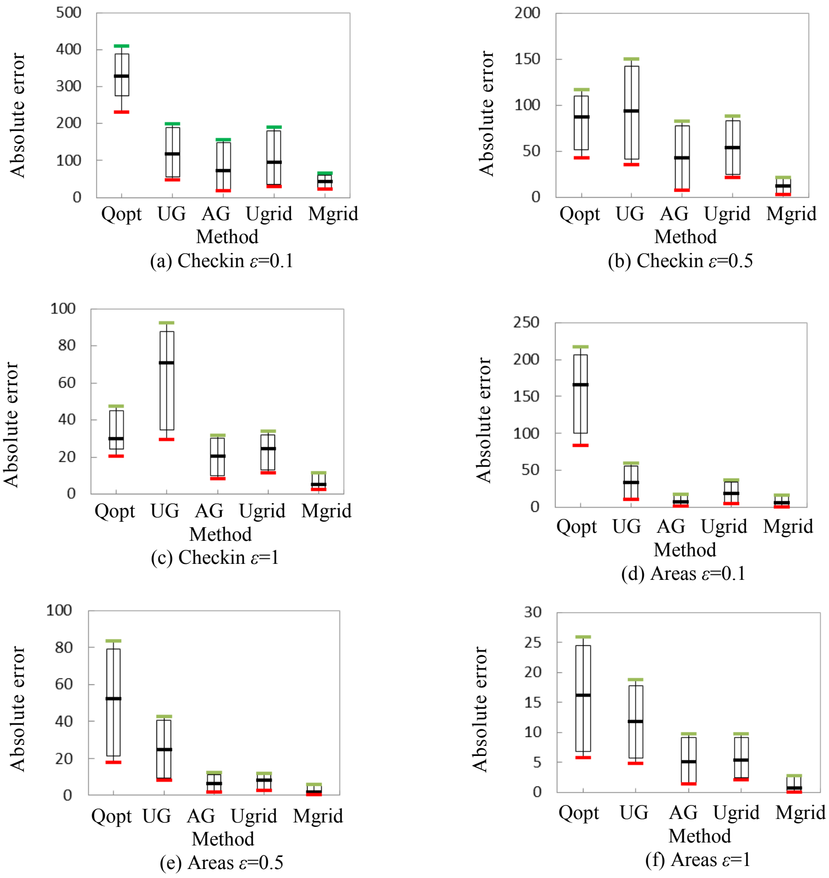
**Figure 5.** Relative error of query.

Figure 5 shows the relative error in different queries. Compared with other methods, Mgrid outweighs other methods on a whole. The relative error of method Ugrid is comparatively large as shown in Figure 5b, and we believe this kind of situation is caused by excessive sparse cells contained in query $Q$. Mgrid further enhances the query accuracy by the similar cells merging strategy, which merges the similar cells into a partition to decrease the added noise to each cell. Experimental results verify the effectiveness of the granularity partitioning model and the similar cells merging strategy.

Note that the experimental results of UG outperform the results of AG in Figure 5e. This situation may be caused by the privacy budget allocation and the sparsity of dataset. AG allocates half of privacy budget to the first-level cells and the other half of the budget to the second-level cells; different allocation strategy will have an impact on the query results.

Figure 6 shows the profile of absolute error, which is displayed by candlestick chart, in different queries. Note that the top horizontal line of the candlestick is the maximum value, the bottom horizontal line of the candlestick is the minimum value, and the middle horizontal line of the candlestick is the arithmetic mean. The top of the box is 95% of the maximum value, and the bottom of the box is 120% of the minimum value.

As shown in Figure 6, just considering the absolute error of the added noise, we also deduce that Mgrid outperforms the four other methods. Note that the results of Qopt in dataset Areas are getting worse, which shows that a granularity partition based method works well when dealing with the sparse data.



(a) Checkin $\varepsilon$=0.1

(b) Checkin $\varepsilon$=0.5

(c) Checkin $\varepsilon$=1

(d) Areas $\varepsilon$=0.1

(e) Areas $\varepsilon$=0.5

(f) Areas $\varepsilon$=1

**Figure 6.** Absolute error of query.

## 6. Conclusions

The location information of monitored objects is an important privacy attribute in IWSNs, which has become a significant research direction. To protect node location privacy and enhance the utility of perturbed data, we propose a novel granularity partitioning model based on the overall analysis of two types of errors. This model considers that the shape of query $Q$ is a rectangle, which is closer to the requirement of actual queries. Ugrid and Mgrid are proposed based on this model and are validated through two real world datasets. Experimental results show that Mgrid has a good utility and query accuracy.

As for future work, we will continue to improve the granularity partitioning model and make the model to be in better accordance with the self-characteristics of datasets. For instance, using area and point count two factors to construct the model. In addition, we will explore the distributed application for geospatial datasets—for example, multiserver differentially private data release.

**Acknowledgments:** This work is supported by the National Natural Science Foundation of China (No. 61772562, No. 61272497), the Hubei Provincial Natural Science Foundation of China for Distinguished Young Scholars (No. 2017CFA043), the Applied Basic Research Project of Wuhan (2017060201010162), the Fundamental Research Funds for the Central Universities (CZZ17003, CZP17043, 2042017gf0038), and the Youth Elite Project of State Ethnic Affairs Commission of China.

**Author Contributions:** In this paper, S.L. conceived and designed the experiments; J.W. performed the experiments and wrote the paper; R.Z. and Z.C. provided supervision and support for the research work.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this paper:

| | |
|---|---|
| IWSNs | Industrial Wireless Sensor Networks |
| WSNs | Wireless Sensor Networks |
| PSD | Privacy Spatial Decomposition |

## References

1. Posada, J.; Toro, C.; Barandiaran, I.; Oyarzun, D.; Stricker, D.; de Amicis, R.; Pinto, E.B.; Eisert, P.; Döllner, J.; Vallarino, I., Jr. Visual computing as a key enabling technology for Industrie 4.0 and Industrial Internet. *IEEE Comput. Graph. Appl.* **2015**, *35*, 26–40.
2. Chu, W.C.; Ssu, K.F. Location-free boundary detection in mobile wireless sensor networks with a distributed approach. *Comput. Netw.* **2014**, *70*, 96–112.
3. Li, X.; Li, D.; Wan, J.; Vasilakos, A.; Lai, C.; Wang, S. A review of industrial wireless networks in the context of Industry 4.0. *Wirel. Netw.* **2017**, *23*, 23–41.
4. Salam, H.A.; Khan, B.M. IWSN—Standards, challenges and future. *IEEE Potentials* **2016**, *35*, 9–16.
5. Francis, T.; Madiajagan, M.; Kumar, V. Privacy issues and techniques in E-health systems. In Proceedings of the ACM SIGMIS Conference on Computers and People Research, Newport Beach, CA, USA, 4–6 June 2015; pp. 113–115.
6. Dwork, C. Differential privacy. In Proceedings of the 33rd International Colloquium on Automata, Languages and Programming, Venice, Italy, 10–14 July 2006; pp. 1–12.
7. Qardaji, W.; Yang, W.N.; Li, N.H. Differentially private grids for geospatial data. In Proceedings of the IEEE 29th International Conference on data Engineering, Brisbane, Australia, 8–12 April 2013; pp. 757–768.
8. Abiola, S.O.; Portman, E.; Kautz, H.; Dorsey, E.R. Node view: A mHealth real-time infectious disease interface disease interface. In Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and 2015 ACM International Symposium on Wearable Computers, Osaka, Japan, 7–11 September 2015; pp. 297–300.
9. De Montjoye, Y.A.; Radaelli, L.; Singh, V.K.; Pentland, A.S. Unique in the shopping mall: On the reidentifiability of credit card metadata. *Science* **2015**, *347*, 536–539.

10. Xue, A.Y.; Zhang, R.; Zheng, Y.; Xie, X.; Yu, J.H.; Tang, Y. DesTeller: A System for Destination Prediction Based on Trajectories with Privacy Protection. *Proc. VLDB Endow.* **2013**, *6*, 1198–1201.
11. Oualha, N.; Olivereau, A. Sensor and data privacy in industrial wireless sensor networks. In Proceedings of the Network and Information Systems Security, La Rochelle, France, 18–21 May 2011; pp. 1–8.
12. Anas, B.; Abdelshakour, A.; Ausif, M. Source anonymity in WSNs against global adversary utilizing low transmission rates with delay constraints. *Sensor* **2016**, *16*, 957.
13. Huang, C.; Ma, M.; Liu, Y.; Liu, A. Preserving source location privacy for energy harvesting WSNs. *Sensor* **2017**, *17*, 724.
14. Mukherjee, M.; Matam, R.; Shu, L.; Maglaras, L.; Ferrag, M.A.; Choudhry, N.; Kumar, V. Security and privacy in fog computing: Challenges. *IEEE Access* **2017**, *5*, 19293–19304.
15. Cicek, A.E.; Nergiz, M.E.; Saygin, Y. Ensuring location diversity in privacy-preserving spatio-temporal data publishing. *Int. J. Large Data Bases* **2014**, *23*, 609–625.
16. Xiao, Y.H.; Xiong, L. Protecting location with dynamic differential privacy under temporal correlations. In Proceedings of the 22nd ACM Audit and Control Conference on Computer and Communications Security, Denver, CO, USA, 12–16 October 2015; pp. 1298–1309.
17. Machanavajjhala, A.; Kifer, D.; Gehrke, J. L-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data* **2007**, *1*, 1–52.
18. Li, N.H.; Li, T.C.; Venkatasubramanian, S. T-closeness: Privacy beyond k-anonymity and l-diversity. In Proceedings of the IEEE 23rd International Conference on Data Engineering, Istanbul, Turkey, 15–20 April 2007; pp. 106–115.
19. Fabienne, E.; Aniket, K.; Matteo, M.; Francesca, P.; Ivan, P. Differentially private data aggregation with optimal utility. In Proceedings of the 30th Annual Computer Security Applications Conference, New Orleans, LA, USA, 8–12 December 2014; pp. 316–325.
20. Ebadi, H.; Sands, D.; Schneider, G. Differential privacy: Now it's getting personal. In Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Mumbai, India, 15–17 January 2015; pp. 69–81.
21. Fan, L.Y.; Bonomi, L.; Xiong, L.; Sunderam, V. Monitoring web browsing behavior with differential privacy. In Proceedings of the 23rd International Conference on World Wide Web, Seoul, Korea, 7–11 April 2014; pp. 177–188.
22. Dwork, C. A firm foundation for private data analysis. *Commun. ACM* **2011**, *54*, 86–95.
23. Soria-Comas, J.; Domingo-Ferrer, J.; Sánchez, D.; Martínez, S. Enhancing data utility in differential privacy via microaggregation-based k-anonymity. *VLDB J.* **2014**, *23*, 771–794.
24. Shu, L.; Chen, Y.; Huo, Z.; Bergmann, N.; Wang, L. When mobile crowd sensing meets traditional industry. *IEEE Access* **2017**, *5*, 15300–15307.
25. Wang, J.; Liu, S.B.; Li, Y.K. A review of differential privacy in individual data release. *Int. J. Distrib. Sens. Netw.* **2015**, *11*, 259682.
26. Cormode, G.; Procopiuc, C.; Srivastava, D.; Shen, E.; Yu, T. Differentially private spatial decompositions. In Proceedings of the IEEE 28th International Conference on Data Engineering, Washington, DC, USA, 1–5 April 2012; pp. 20–31.
27. Xiao, Y.H.; Xiong, L.; Yuan, C. Differentially private data release through multidimensional partitioning. In Proceedings of the 7th Very Large Data Base Workshop on Secure Date Management, Lecture Notes in Computer Science, Singapore, 17 September 2010; pp. 150–168.
28. Xiao, Y.H.; Gardner, J.; Xiong, L. Dpcube: Releasing differentially private data cubes for health information. In Proceedings of the IEEE 28th International Conference on Data Engineering, Washington, DC, USA, 1–5 April 2012; pp. 1305–1308.
29. Fan, L.Y.; Xiong, L.; Sunderam, V. Differentially private multi-dimensional time series release for traffic monitoring. In Proceedings of the 27th Annual IFIP WG 11.3 Conference, Newark, NJ, USA, 15–17 July 2013; pp. 33–48.
30. Shu, L.; Wang, L.; Niu, J.; Zhu, C.; Mukherjee, M. Releasing network isolation problem in group-based industrial wireless sensor networks. *IEEE Syst. J.* **2017**, *11*, 1340–1350.
31. Shu, L.; Chen, Y.; Sun, Z.; Tong, F.; Mukherjee, M. Detecting the dangerous area of toxic gases with wireless sensor networks. *IEEE Trans. Emerg. Top. Comput.* **2017**, 1, doi:10.1109/TETC.2017.2700358.

32. Hay, M.; Rastogi, V.; Miklau, G. Boosting the accuracy of differentially private histograms through consistency. *Proc. Large Data Base Endow.* **2010**, *3*, 1021–1032.
33. To, H.; Ghinita, G.; Shahabi, C. A framework for protecting worker location privacy in spatial crowdsourcing. *Proc. Large Data Base Endow.* **2014**, *7*, 919–930.
34. McSherry, F.; Liu, S.B.; Li, Y.K. Privacy integrated queries: An extensible platform for privacy-preserving data analysis. *Commun. ACM* **2010**, *53*, 89–97.
35. Dwork, C.; McSherry, F.; Nissim, K.; Smith, A. Calibrating noise to sensitivity in private data analysis. In Proceedings of the 3rd Conference on Theory of Cryptography, New York, NY, USA, 4–7 March 2006; pp. 265–284.
36. Mcsherry, F.; Talwar, K. Mechanism design via differential privacy. In Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science, Providence, RI, USA, 21–23 October 2007; pp. 94–103.
37. Frey, B.J.; Dueck, D. Clustering by passing messages between data points. *Science* **2007**, *315*, 972–976.
38. Xiao, Q.; Chen, R.; Tan, K.L. Differentially private network data release via structural inference. In Proceedings of the 20th ACM Knowledge Discovery and Data Mining International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 911–920.
39. Leskovec, J. Govalla. Available online: http://snap.stanford.edu/data/loc-gowalla.html (accessed on 28 January 2018).
40. Geodatabases. Available online: https://www.census.gov/geo/maps-data/data/tiger-geodatabases.html (accessed on 19 July 2015).