Privacy-Preserving Detection of Anomalous Phenomena in Crowdsourced Environmental Sensing

Mihai Maruseac[1], Gabriel Ghinita[1(✉)], Besim Avci[2], Goce Trajcevski[2], and Peter Scheuermann[2]

[1] University of Massachusetts, Boston, MA 02125, USA
{mmarusea,gghinita}@cs.umb.edu
[2] Northwestern University, Evanston, IL 60208, USA
{besim,goce,peters}@eecs.northwestern.edu

Abstract. Crowdsourced environmental sensing is made possible by the wide-spread availability of powerful mobile devices with a broad array of features, such as temperature, location, velocity, and acceleration sensors. Mobile users can contribute measured data for a variety of purposes, such as environmental monitoring, traffic analysis, or emergency response. One important application scenario is that of detecting anomalous phenomena, where sensed data is crucial to quickly acquire data about forest fires, environmental accidents or dangerous weather events. Such cases typically require the construction of a *heatmap* that captures the distribution of a certain parameter over a geospatial domain (e.g., temperature, CO_2 concentration, water polluting agents, etc.).

However, contributing data can leak sensitive private details about an individual, as an adversary may be able to infer the presence of a person in a certain location at a given time. In turn, such information may reveal information about an individual's health, lifestyle choices, and may even impact the physical safety of a person. In this paper, we propose a technique for privacy-preserving detection of anomalous phenomena, where the privacy of the individuals participating in collaborative environmental sensing is protected according to the powerful semantic model of *differential privacy*. Our techniques allow accurate detection of phenomena, without an adversary being able to infer whether an individual provided input data in the sensing process or not. We build a differentially-private index structure that is carefully customized to address the specific needs of anomalous phenomenon detection, and we derive privacy-preserving query strategies that judiciously allocate the privacy budget to maintain high data accuracy. Extensive experimental results show that the proposed approach achieves high precision of identifying anomalies, and incurs low computational overhead.

1 Introduction

Environmental sensing using crowdsourcing is a promising direction due to the widespread availability of mobile devices with positioning capabilities and a

C. Claramunt et al. (Eds.): SSTD 2015, LNCS 9239, pp. 313–332, 2015.
DOI: 10.1007/978-3-319-22363-6_17

broad array of sensing features, e.g., audio and video capture, temperature, velocity, acceleration, etc. In addition, mobile devices can easily interface with external sensors and upload readings for many other environmental parameters (e.g., CO_2, water pollution levels, atmospheric pressure). The growing trend towards crowdsourcing environmental sensing is beneficial for a wide range of applications, such as pollution levels monitoring or emergency response. In such a setting, authorities can quickly and inexpensively acquire data about forest fires, environmental accidents or dangerous weather events.

One particular task that is relevant to many application domains is that of detecting anomalous phenomena. Such cases typically require to determine a *heatmap* capturing the distribution of a certain sensed parameter (e.g., temperature, CO_2 level) over a geospatial domain. When the parameter value in a certain region reaches a predefined threshold, then an alarm should be triggered, signaling the occurrence of an anomaly. Furthermore, the alarm should identify with good accuracy the region where the dangerous event occurred, so that countering measures can be deployed to that region.

However, there are important privacy concerns related to crowdsourced sensing. Contributed data may reveal sensitive private details about an individual's health, lifestyle choices, and may even impact the physical safety of a person. To protect against such disclosure, the state-of-the-art model of *differential privacy (DP)* adds noise to data in a way that prevents an adversary from learning whether the contribution of an individual is present in a dataset or not. Several DP-compliant techniques for protecting location data have been proposed in [1,16,17]. However, these approaches consider only simple, general-purpose count queries, and rely on simplifying assumptions that make them unsuitable for our considered problem of anomalous phenomenon detection.

Consider the example of a forest fire, where mobile users report air temperature in various regions. To model the fire spread, one needs to plot the temperature distribution, which depends on the values reported by individual users, and the users' reported locations. With existing techniques, one could partition the dataspace according to a regular grid and split the available privacy budget between two aggregate query types, one counting user locations in each grid cell, and the other summing reported values. Next, a temperature heatmap is obtained by averaging the temperature for each cell. As we show in our experimental evaluation, this approach results to useless data, due to the high amount of noise injected. This is the result of a more fundamental limitation of existing approaches that are designed only for general-purpose queries, and do not take into account correlations that are specific to more complex data processing algorithms.

In this paper, we propose an accurate technique for privacy-preserving detection of anomalous phenomena in crowdsourced sensing. We also adopt the powerful semantic model of *differential privacy*, but we devise a tailored solution, specifically designed for privacy-preserving heatmap construction. Our technique builds a flexible data indexing structure that can provide query results at arbitrary levels of granularity. Furthermore, the sanitization process fuses together distinct types of information (e.g., user count, placement and reported value

scale) to obtain an effective privacy-preserving data representation that can help decide with high accuracy whether the sensed value in a certain geographical region exceeds the threshold or not. To the best of our knowledge, this is the first work that addresses the problem of value heatmap construction within the differential privacy framework. Our specific contributions are:

1. We introduce a hierarchical differentially-private structure for representing sensed data collected by mobile users. The structure is customized to address the specific requirements of value heatmap construction, and accurately supports queries at variable levels of granularity.
2. We examine the impact of structure parameters and privacy budget allocation on data accuracy, and devise algorithms for parameter selection and tuning.
3. We investigate techniques for reducing the impact of DP-injected noise, and devise effective voting strategies during data processing that increase accuracy of anomalous phenomenon detection.
4. We perform an extensive experimental evaluation which shows that the proposed techniques accurately detect anomalous phenomena, and clearly outperform existing general-purpose sanitization methods that fare poorly when applied to the studied problem.

The paper is organized as follows: Sect. 2 provides background information on differential privacy. In Sect. 3, we introduce the system model, and the metrics used to characterize anomalous phenomenon detection accuracy. Section 4 presents the proposed privacy-preserving data indexing structure and analytical models for characterizing query accuracy. We introduce strategies for anomaly detection in Sect. 5, followed by experimental evaluation results in Sect. 6. We present related work in Sect. 7, and conclude with directions for future work in Sect. 8.

2 Background

2.1 Differential Privacy

Differential privacy (DP) [2,3] addresses the limitation of syntactic privacy models (e.g., k-anonymity [19], ℓ-diversity [12], t-closeness [9]) which are vulnerable against background knowledge attacks. DP is a semantic model which argues that one should minimize the risk of disclosure that arises from an individual's participation in a dataset.

Two datasets $\mathcal{D}$ and $\mathcal{D}'$ are said to be *siblings* if they differ in a single record r, i.e., $\mathcal{D}' = \mathcal{D} \cup \{r\}$ or $\mathcal{D}' = \mathcal{D} \setminus \{r\}$. An algorithm $\mathcal{A}$ is said to satisfy differential privacy with parameter ε (called *privacy budget*) if the following condition is satisfied [2]:

Definition 1 (ε-indistinguishability). *Consider algorithm $\mathcal{A}$ that produces output $\mathcal{O}$ and let $\epsilon > 0$ be an arbitrarily-small real constant. Algorithm $\mathcal{A}$ satisfies* ε-indistingui-shability *if for every pair of sibling datasets $\mathcal{D}, \mathcal{D}'$ it holds that*

$$\left| \ln \frac{Pr[\mathcal{A}(\mathcal{D}) = \mathcal{O}]}{Pr[\mathcal{A}(\mathcal{D}') = \mathcal{O}]} \right| \leq \varepsilon \tag{1}$$

In other words, an attacker is not able to learn, with significant probability, whether output $\mathcal{O}$ was obtained by executing $\mathcal{A}$ on input $\mathcal{D}$ or $\mathcal{D}'$. To date, two prominent techniques have been proposed to achieve ε-indistinguishability [3,13]: the *Laplace mechanism* (and the closely related geometric mechanism for integer-valued data) and the *exponential mechanism*. Both mechanisms are closely related to the concept of *sensitivity*:

Definition 2 (L_1-sensitivity [3]). *Given any two sibling datasets $\mathcal{D}$, $\mathcal{D}'$ and a set of real-valued functions $\mathcal{F} = \{f_1, \ldots, f_m\}$, the* L_1-sensitivity *of $\mathcal{F}$ is measured as* $\Delta_{\mathcal{F}} = \max_{\forall \mathcal{D}, \mathcal{D}'} \sum_{i=1}^{m} |f_i(\mathcal{D}) - f_i(\mathcal{D}')|$.

The *Laplace mechanism* is used to publish the results to a set of statistical queries. A statistical query set $\mathcal{Q} = \{Q_1, \ldots, Q_m\}$ is the equivalent of a set of real-valued functions, hence the sensitivity definition immediately extends to such queries. According to [3], to achieve DP with parameter ε it is sufficient to add to each query result random noise generated according to a Laplace distribution with mean $\Delta_{\mathcal{Q}}/\varepsilon$. For COUNT queries that do not overlap in the data domain (e.g., finding the counts of users enclosed in disjoint grid cells), the sensitivity is 1.

An important property of differentially-private algorithms is *sequential composability* [13]. Specifically, if two algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ executing in isolation on dataset $\mathcal{D}$ achieve DP with privacy parameters ε_1 and ε_2 respectively, then executing both $\mathcal{A}_1$ and $\mathcal{A}_2$ on $\mathcal{D}$ in sequence achieves DP with parameter $(\varepsilon_1 + \varepsilon_2)$. In contrast, *parallel composability* specifies that executing $\mathcal{A}_1$ and $\mathcal{A}_2$ on disjoint partitions of the dataset achieves DP with parameter $\max(\varepsilon_1, \varepsilon_2)$.

2.2 Private Spatial Decompositions (PSD)

The work in [1] introduced the concept of *Private Spatial Decompositions (PSD)* to release spatial datasets in a DP-compliant manner. A PSD is a spatial index transformed according to DP, where each index node is obtained by releasing a noisy count of the data points enclosed by that node's extent. Various index types such as grids, quad-trees or k-d trees [18] can be used as a basis for PSD.

Accuracy of PSD is heavily influenced by the type of PSD structure and its parameters (e.g., height, fan-out). With space-based partitioning PSD, the split position for a node does not depend on data point locations. This category includes flat structures such as grids, or hierarchical ones such as BSP-trees (Binary Space Partitioning) and quad-trees [18]. The privacy budget ϵ needs to be consumed only when counting the users in each index node. Typically, all nodes at same index level have non-overlapping extents, which yields a constant and low sensitivity of 1 per level (i.e., adding/removing a single location in the data may affect at most one partition in a level). The budget ϵ is best distributed across levels according to the *geometric allocation* [1], where leaf nodes receive more budget than higher levels. The sequential composition theorem applies across nodes on the same root-to-leaf path, whereas parallel composition applies

to disjoint paths in the hierarchy. Space-based PSD are simple to construct, but can become unbalanced.

Object-based structures such as k-d trees and R-trees [1] perform splits of nodes based on the placement of data points. To ensure privacy, split decisions must also be done according to DP, and significant budget may be used in the process. Typically, the exponential mechanism [1] is used to assign a merit score to each candidate split point according to some cost function (e.g., distance from median in case of k-d trees), and one value is randomly picked based on its noisy score. The budget must be split between protecting node counts and building the index structure. Object-based PSD are more balanced in theory, but they are not very robust, in the sense that accuracy can decrease abruptly with only slight changes of the PSD parameters, or for certain input dataset distributions.

The recent work in [16] compares tree-based methods with multi-level grids, and shows that two-level grids tend to perform better than recursive partitioning counterparts. The paper also proposes an *Adaptive Grid (AG)* approach, where the granularity of the second-level grid is chosen based on the noisy counts obtained in the first-level (sequential composition is applied). AG is a hybrid which inherits the simplicity and robustness of space-based PSD, but still uses a small amount of data-dependent information in choosing the granularity for the second level.

All these methods assume general-purpose and homogeneous queries (i.e., find counts of users in various regions of the dataspace), and, as we show later in this paper, are not suitable for the problem of anomalous phenomenon detection. We compare against state-of-the-art PSD techniques in our experimental evaluation (Sect. 6).

3 System Model and Evaluation Metrics

We consider a two-dimensional geographical region and a phenomenon characterized by a scalar value (e.g., temperature, CO_2 concentration) within domain $[0, M]$. A number of N mobile users measure and report phenomenon values recorded at their location. If a regular grid is super-imposed on top of the data domain, then the histogram obtained by averaging the values reported within each grid cell provides a *heatmap* of the observed phenomenon. Since our focus is on detecting anomalous phenomena, the actual value in each grid cell is not important; instead, what we are concerned with is whether a cell value is above or below a given threshold T, $0 < T < M$.

Mobile users report sensed values to a trusted data collector, as illustrated in Fig. 1. The collector sanitizes the set of reported values according to differential privacy with parameter ε, and outputs as result a data structure representing a noisy index of the data domain, i.e., a PSD. This PSD is then released to data recipients (i.e., general public) for processing. Based on the PSD, data recipients are able to answer queries with arbitrary granularity that is suitable for their specific data uses. Furthermore, each data recipient has flexibility to choose a different threshold value T in their analysis. In practice, the trusted collector

role can be fulfilled by cell phone companies, which already know the locations of mobile users, and may be bound by contractual obligations to protect users' location privacy. The collector may charge a small fee to run the sanitization process, or can perform this service free of charge, and benefit from a tax break, e.g., for supporting environmental causes.

According to differential privacy, the goal of the protection mechanism is to hide whether a certain individual contributed to the set of sensed values or not. To achieve protection, noise is added to the values of individual value reports. Furthermore, fake value reports may have to be inserted, and some actual readings may have to be deleted from the dataset. Inherently, protection decreases data accuracy.

To measure the accuracy of sanitization, we need to quantify the extent to which the outcome for certain regions changes from above the threshold to below, or vice-versa. Given an arbitrary-granularity regular grid, we define the following metrics:

ϕ_{both}: number of grid cells above the threshold according to *both* actual and sanitized readings.
ϕ_{either}: number of grid cells above the threshold according to *either* actual or sanitized readings.
ϕ_{flip}: number of grid cells above the threshold in one dataset and below in the other.
ϕ_{all}: total number of grid cells.

It results immediately from the metric definitions that $\phi_{either} = \phi_{flip} + \phi_{both}$. Hence, we can define two additional metrics with domain $[0, 1]$ and ideal value of 1 (i.e., perfect accuracy). **FlipRatio (FR)** quantifies the proportion of cells that change their outcome due to sanitization:

$$FR = 1 - \frac{\phi_{flip}}{\phi_{all}}$$

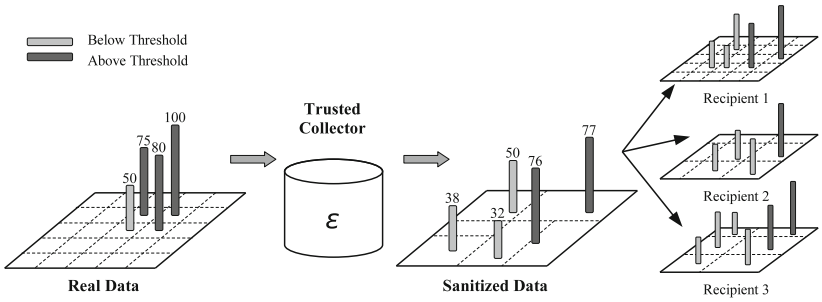


Fig. 1. System model

The **Jaccard (J)** metric, derived from *the Jaccard similarity coefficient* [2], measures the dissimilarity between the real and sanitized datasets:

$$J = \frac{\phi_{both}}{\phi_{either}}$$

The FR and J metrics have the advantage of being less dependent on the grid granularity, i.e., the ϕ_{all} values, so they maintain their relevance across a broad range of query granularities. However, only the J metric captures the local impact of the sanitization method. Interchanging the state of two random cells will not change the values of any other metrics than J, so they are not sufficient to determine the accuracy of the heatmap. Therefore, in the rest of the paper, we focus on the J metric. Formally, our problem statement is:

Problem 1. Given N users moving within a two-dimensional space, a phenomenon characterized by a scalar value with domain range $[0, M]$, an anomaly threshold T, $0 < T < M$ and privacy budget ε, determine an ε-differentially-private release such that the Jaccard metric between the real and sanitized dataset is maximized.

4 PSD for Anomalous Phenomenon Detection

Constructing an appropriate PSD is an essential step, since the accuracy of the entire solution depends on the structure properties. Furthermore, due to the specific requirements of our problem, general-purpose PSDs such as the ones optimized for count queries [1,16,17] are not suitable.

The anomalous phenomenon detection may be performed with respect to a regular grid of arbitrarily fine-grained granularity. On the other hand, creating a PSD that is too fine-grained is not a suitable approach. According to the Laplace mechanism, each cell's query result is added with random noise of magnitude independent of the actual value. Therefore, PSDs with small cells and PSDs that do not adapt to data density are not appropriate, as the resulting inaccuracy is high. Instead, we construct a flexible structure, based on which the threshold condition can be answered for arbitrary regular grids, as illustrated on the right side of Fig. 1.

The PSD must keep track of two measures necessary to determine phenomena heatmaps: sensor counts[1] and phenomenon value sums, which together provide average values for each cell. We denote the actual values for sensor count and value sum in a cell by n and s, respectively (we use subscript indices to distinguish the n and s values across cells). We denote the sanitized counts and sums by n^* and s^*. The sensitivity of n is 1, whereas the sensitivity of s is M (adding a new sensor in a cell can increase n by 1 and s by M). Hence, if n is answered using privacy budget ε_n and s is answered using privacy budget ε_s, the variance of n^* is $\frac{2}{\varepsilon_n^2}$, whereas the variance of s^* is $\frac{2M^2}{\varepsilon_s^2}$.

[1] In the rest of the paper, the terms *mobile user* and *sensor* are used interchangeably.

To simplify presentation, we introduce our PSD in incremental fashion: first, we outline the main concepts and parameters for a single-level regular grid. Next, we extend our findings to a two-level structure, and then generalize to a multiple-level structure. Table 1 summarizes the notations used.

Single-level Grid. Assume a regular grid of $N_0 \times N_0$ cells spanning over a data domain of size $w \times w$. Similar to other work on PSD [10,16], we assume that a negligible fraction of the privacy budget is spent to estimate n_0^*, the total number of sensors, and s_0^*, the sum of all sensed values. Granularity N_0 must be chosen to minimize the expected error over all rectangular queries (since any query can be decomposed into non-overlapping rectangular regions). The error has two sources:

- *Laplace error* within a single cell due to noise addition by the Laplace mechanism. These errors are added for all cells covered by the query.
- *Non-uniformity error* caused by non-uniformity of sensor distribution within a grid cell. These errors occur only for cells which are partially covered by the query rectangle. In such a case, we output a value proportional to the fraction of the cell that overlaps the query.

Furthermore, errors occur for both sensor counts and sensed values. Since the threshold T is expected to be proportional to scale M, we normalize the error for sensed values to account for the skew introduced by M. The error expression subject to minimization becomes the sum of all count errors plus $\frac{1}{M}$ of the sum of all value sum errors.

Table 1. Symbols and notations used in the paper.

Symbol	Description
n, s	Real count and sum of values of sensors in a cell
n^*, s^*	Noisy count and sum of values of sensors in a cell
n', s'	Count and sum of values of sensors in a cell after weighted averaging
$\overline{n}$, $\overline{s}$	Count and sum of values of sensors in a cell after mean consistency step
ε	Privacy budget
$\varepsilon_n, \varepsilon_s$	Privacy budget used for answering count and, respectively, sum queries in the cell
α	Proportion of available privacy budget to use at current PSD level
β	Proportion of privacy budget for the current level used for answering count queries
N_u	Split factor for cell u
M	Maximum value of a sensor's scale
T	Threshold for the anomalous heatmap
N_t	Threshold for minimum (noisy) number of sensors in a cell
K	Non-uniformity constant

Consider an arbitrary rectangle query of size rw^2, $r \in (0,1)$. The query will cover approximately rN_0^2 cells. The total variance of the query result is $\frac{2rN_0^2}{\varepsilon_n^2}$ for n and $\frac{2M^2rN_0^2}{\varepsilon_s^2}$ for s. Hence, the count error is expressed as $\sqrt{2r}\frac{N_0}{\varepsilon_n}$, and the sum error as $\sqrt{2r}\frac{MN_0}{\varepsilon_s}$. The total Laplace error is $\sqrt{2r}N_0\left(\frac{1}{\varepsilon_n}+\frac{1}{\varepsilon_s}\right)$.

The query rectangle might partially cover some cells. The number of such cells is of the order $\mathcal{O}(\sqrt{r}N_0)$ (determined by the perimeter of the query rectangle). Hence, we can assume that the number of points in partially covered cells is of the order $\mathcal{O}(\sqrt{r}N_0\frac{n_0^*}{N_0^2}) = K\sqrt{r}\frac{n_0^*}{N_0}$, where K is a constant. Assuming uniform sensor density, the error for value sum in partially covered cells is $K\sqrt{r}\frac{s_0^*}{N_0}$. Hence, the non-uniformity error is $K\frac{\sqrt{r}}{N_0}\left(n_0^*+\frac{s_0^*}{M}\right)$.
Thus, we must minimize the expression:

$$\sqrt{2r}N_0\left(\frac{1}{\varepsilon_n}+\frac{1}{\varepsilon_s}\right)+K\frac{\sqrt{r}}{N_0}\left(n_0^*+\frac{s_0^*}{M}\right) \tag{2}$$

According to the sequential composition property (Sect. 2), the available privacy budget ε must be split between ε_n and ε_s. We capture this split with parameter $\beta \in (0,1)$, defined as the fraction used by the count sanitization: $\varepsilon_n = \beta\varepsilon$ and $\varepsilon_s = (1-\beta)\varepsilon$. Minimizing Eq. (2) with respect to N_0, we obtain the optimal single-level granularity

$$N_0 = \sqrt{\varepsilon \times \frac{K}{\sqrt{2}} \times \beta(1-\beta)\left(n_0^*+\frac{s_0^*}{M}\right)} \tag{3}$$

Two-level Grid. Starting with the optimal single-level N_0 setting, we further divide each cell according to its noisy n^* and s^*. The privacy budget must be split between the two levels according to sequential composition. We model this split with parameter $\alpha \in (0,1)$, which quantifies the budget fraction allocated to the level 1 grid. Levels 1 and 2 receive respectively budgets $\varepsilon_1 = \alpha\varepsilon$ and $\varepsilon_2 = (1-\alpha)\varepsilon$. Each level budget is further divided between counts and sums using parameter $\beta \in (0,1)$:

$$\varepsilon_{n1} = \beta\varepsilon_1, \varepsilon_{s1} = (1-\beta)\varepsilon_1, \varepsilon_{n2} = \beta\varepsilon_2, \varepsilon_{s2} = (1-\beta)\varepsilon_2 \tag{4}$$

Since each level-1 cell is further divided, we define N_0 as a fraction of the value in Eq. (3) (later in this section, Eq. (11) shows how to choose η):

$$N_0 = \frac{1}{\eta}\sqrt{\varepsilon \times \frac{K}{\sqrt{2}} \times \beta(1-\beta)\left(n_0^*+\frac{s_0^*}{M}\right)} \tag{5}$$

For each cell u in the first level we use budgets ε_{n1} and ε_{s1} to determine n_{u1}^* and, respectively, s_{u1}^*. Based on these values, we split cell u into N_u^2 cells. For each cell $v \in child(u)$, we use ε_{n2} and ε_{s2} to determine n_{v2}^* and, respectively, s_{v2}^* (the subscript indicates the level of the grid where the value is computed).

Since the actual sensor count in a cell at level 1 is the same as the sum of the sensor counts in all of its children at level 2 (and the same holds for the sums), we perform a constrained inference procedure with the purpose of improving accuracy. Based on the values n^*_{u1}, s^*_{u1}, n^*_{v2}, s^*_{v2} we determine $\overline{n_{u1}}$, $\overline{s_{u1}}$, $\overline{n_{v2}}$ and $\overline{s_{v2}}$ such that

$$\overline{n_{u1}} = \sum_{v \in child(u)} \overline{n_{v2}}$$

$$\overline{s_{u1}} = \sum_{v \in child(u)} \overline{s_{v2}}$$

and $\forall u$, the variances of $\overline{n_{u1}}$ and $\overline{s_{u1}}$ are minimized. Note that, since all input values are already sanitized, no budget is consumed in the constrained inference step, and differential privacy is still enforced.
We determine these values in two steps:

1. We determine the **weighted average** estimators n'_{u1} and s'_{u1} with minimal variance. We average the values of n^*_{u1} and $\sum_{v \in child(u)} n^*_{v2}$ to determine n'_{u1} and the corresponding ones for s'_{u1}. To do so, we are using the fact that the variance of the weighted average of two random variables X and Y with variances $Var(X)$ and $Var(Y)$ is minimized by the value
$$\frac{Var(Y)}{Var(X) + Var(Y)} \times X + \frac{Var(X)}{Var(X) + Var(Y)} \times Y \tag{6}$$
In our case, X is n'_{u1} (s'_{u1}) and Y is $\sum_{v \in child(u)} n^*_{v2}$ (respectively $\sum_{v \in child(u)} s^*_{v2}$).
2. We update the values to ensure **mean consistency** according to:
$$\overline{n_{u1}} = n'_{u1} \;\; \overline{n_{v2}} = n'_{v2} + \frac{1}{N_u^2}\left(\overline{n_{u1}} - \sum_{v \in child(u)} n'_{v2}\right) \tag{7}$$
$$\overline{s_{u1}} = s'_{u1} \;\; \overline{s_{v2}} = s'_{v2} + \frac{1}{N_u^2}\left(\overline{s_{u1}} - \sum_{v \in child(u)} s'_{v2}\right) \tag{8}$$

The effects of the constrained inference so far concern only queries which partially cover level-1 cells. Suppose that a query covers $i \times j$ sub-cells of cell u, where $i, j \in \{1, 2, \ldots N_u\}$. Then, the effect of the constrained inference is that $\min(i \times j, N_u^2 - i \times j)$ level-2 cells will be used to answer the query. On average, the number of level-2 cells required to answer a query is:

$$\frac{1}{N_u^2 - 1} \sum_{i=1}^{N_u} \sum_{j=1}^{N_u} \min(i \times j, N_u^2 - i \times j) \approx \frac{N_u^2}{5} + \mathcal{O}(N_u)$$

Hence, the total variances are $\frac{2N_u^2}{5\varepsilon_{n2}^2}$ and $\frac{2M^2N_u^2}{5\varepsilon_{s2}^2}$, and the resulting total Laplace error is $\frac{\sqrt{10}N_u}{5}\left(\frac{1}{\varepsilon_{n2}} + \frac{1}{\varepsilon_{s2}}\right)$.

For non-uniformity errors, assume r is the ratio between the area used to answer the query and the total area of the cell. We know from the single-level case that the non-uniformity errors are $K\sqrt{r}\frac{n_u^*}{N_u}$ and $K\sqrt{r}\frac{s_u^*}{N_u}$. To eliminate the $\sqrt{r}$ factor, we integrate over its domain ((0, 0.5]) and compute the expected value of the total non-uniformity error. Since $\frac{\int_0^{0.5}\sqrt{r}dr}{\int_0^{0.5}dr} = \frac{\sqrt{2}}{3}$ we get that the total non-uniformity error is $\frac{\sqrt{2}K}{3N_u}\left(n_u^* + \frac{s_u^*}{M}\right)$.
Thus, we must minimize the expression

$$\frac{\sqrt{10}N_u}{5}\left(\frac{1}{\varepsilon_{n2}} + \frac{1}{\varepsilon_{s2}}\right) + \frac{\sqrt{2}K}{3N_u}\left(n_u^* + \frac{s_u^*}{M}\right)$$

and we obtain

$$N_u = \sqrt{\frac{\sqrt{5}}{3}\varepsilon K\beta(1-\beta)(1-\alpha)\left(n_u^* + \frac{s_u^*}{M}\right)} \tag{9}$$

where we can approximate $\frac{\sqrt{10}}{3}$ by 1. This also provides a value for η (Eq.(5)), such that:

$$N_0 = \sqrt{\varepsilon \times \frac{K}{\sqrt{2}} \times \beta(1-\beta)\alpha\left(n_0^* + \frac{s_0^*}{M}\right)} \tag{10}$$

$$N_u = \sqrt{\varepsilon \times \frac{K}{\sqrt{2}} \times \beta(1-\beta)(1-\alpha)\left(n_u^* + \frac{s_u^*}{M}\right)} \tag{11}$$

Generalization to Multiple Levels. The analysis for two levels can be extended to a multiple-level structure, where the privacy budget is split across levels (keeping $\alpha\varepsilon$ for the current level and dividing privacy budget between count and sum using β, as before), and the granularity for each new level is determined based on the sanitized data and variance analysis at the previous level. However, we must carefully decide when to end the recursion, as having too many levels will decrease the budget per level, and consequently decrease accuracy. Because of this, we implement two stopping mechanisms: first, we introduce a maximum depth of the PSD, *max_depth*, to prevent excessive reduction of per-level privacy budget. Second, we introduce a threshold, N_t such that a cell u is divided only if its estimated sensor count satisfies inequality $n_u^* > N_t$.
The number N_u of children nodes of u is given by:

$$N_u = \sqrt{\varepsilon_u \times \frac{K}{\sqrt{2}} \times \beta(1-\beta)(1-\alpha)\left(n_u^* + \frac{s_u^*}{M}\right)} \tag{12}$$

We illustrate the proposed multiple-level PSD approach with a running example, in parallel with the description of the pseudocode provided in Algorithm 1. The PSD is built in three phases. First, the PSD structure is determined (i.e., the spatial extent of each index node), by splitting cells according to Eq. (12),

and noisy values are computed for sensor counts and value sums. This is the only step that accesses the real dataset of readings, and hence the only step that consumes privacy budget. The recursive procedure `buildPSD` (Algorithm 1) summarizes this process.

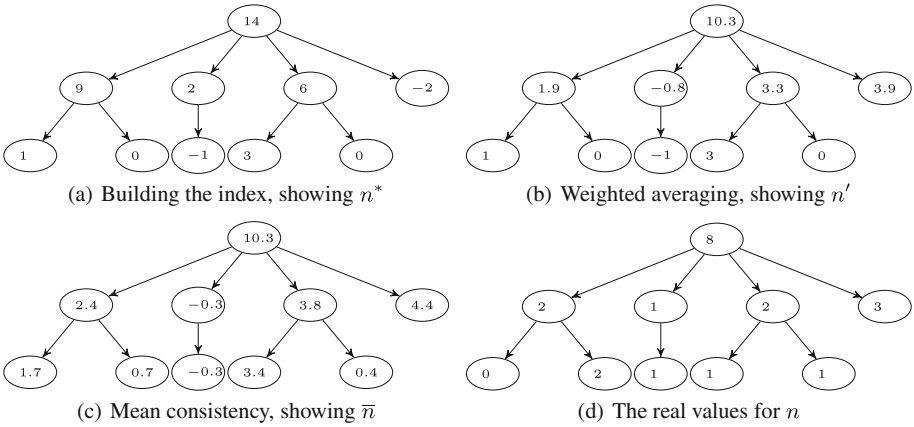
Algorithm 1. Splitting a PSD cell u at depth *depth*, with privacy budget ε

```
1: function BUILDPSD(ε, u, depth)
2:     if depth == max_depth then
3:         ε_crt ← ε
4:     else
5:         ε_crt ← αε
6:     end if
7:     ε_n ← βε_crt
8:     ε_s ← (1 − β)ε_crt
9:     (n, s) ← GETREALVALUES(u)
10:    n* ← n + LAPLACE(1/ε_n)
11:    s* ← s + LAPLACE(M/ε_s)
12:    N_u ← COMPUTESPLIT(ε, n*, s*)
13:    if N_u < Nt then
14:        ε_n ← β(1 − α)ε
15:        ε_s ← (1 − β)(1 − α)ε
16:        n'* ← n + LAPLACE(1/ε_n)
17:        s'* ← s + LAPLACE(M/ε_s)
18:        n' ← AVERAGE(n*, n'*)
19:        s' ← AVERAGE(s*, s'*)
20:    end if
21:    for all v ∈ SPLITCELL(u, N_u, depth) do
22:        BUILDPSD((1 − α)ε, v, depth + 1)
23:    end for
24: end function
```

Figure 2 illustrates PSD construction with $\alpha = 0.2$, $\beta = 0.5$ and $\varepsilon = 1.6$. The root node will receive a budget of $\varepsilon_{n,root} = 0.5 \times 0.2 \times 16 = 0.16$ (lines 2–8 of Algorithm 1). Line 9 computes the real values for the count and sum of sensor values inside the cell (the sensor counts for the running example are presented in Fig. 2(d)). Lines 10–11 add Laplace noise, resulting in a value of $n^*_{root} = 14$. The split granularity for next level is determined as in Eq. (12). Assume we obtain $N_u = 4$, larger than the threshold $N_t = 2$. The root is split into four cells, and the procedure is recursively applied to each of them with $\varepsilon_1 = (1 - \alpha)\varepsilon = 0.8 \times 1.6 = 1.28$.

The budget for level 1 is further split between sum and count values, to obtain $\varepsilon_{n,1} = 0.128$ (lines 2–8). Adding the corresponding Laplace noise to the real values of 2, 1, 2 and 3 (Fig. 2(d)) (lines 10–11), results in noisy counts 9, 2, 6 and, respectively, -2 (Fig. 2(a)).

The cells with values 9 and 6 are further split, while the one with $n^*_1 = -2$ is not, due to the value of N_t. In case no further splits are performed, the remaining



(a) Building the index, showing n^* (b) Weighted averaging, showing n'

(c) Mean consistency, showing $\overline{n}$ (d) The real values for n

Fig. 2. Representation of PSD construction, including weighted averaging and mean consistency.

budget is used by running lines 13–20 of Algorithm 1, which compute new noisy estimates which are averaged to determine n' and, respectively, s'.

Since the remaining cells are at the maximal depth allowed by the method, the remaining privacy budget of $\varepsilon_{n,2} = 0.512$ is used to compute the remaining noisy values. The result of the algorithm is shown in Fig. 2(a).

The second phase of the index building method is **weighted averaging**. We average for each internal node the two estimates and compute n' and s' according to Eq. (6). For each node, we keep track of the variance of the noisy variables and the averaged values, since they will be needed in the higher levels of the tree. The resulting tree at the end of this phase is shown in Fig. 2(b).

Finally, the last phase performs **mean consistency**, which ensures that the estimate from one node is the same as the sum of the estimates from its children. We use Eqs. (7) and (8) in a top-down traversal of the tree, the result of which is shown in Fig. 2(c).

5 PSD Processing and Heatmap Construction

As illustrated in Fig. 1 (Sect. 3), after the PSD is finalized at the trusted collector, it is distributed to data recipients who process it according to their own granularity and threshold requirements. The objective of the data recipient is to obtain a binary heatmap that captures areas with anomalous phenomena, i.e., regions of the geographical domain where the measured values are above the recipient-specified threshold.

We assume that the recipient is interested in building a heatmap according to a *recipient resolution grid* (*rrg*). Recall that our solution is designed to be flexible with respect to recipient requirements, and each recipient may have its own *rrg* of arbitrary granularity. In this section, we show how a recipient is able to accurately determine a phenomenon heatmap given as input the PSD, the

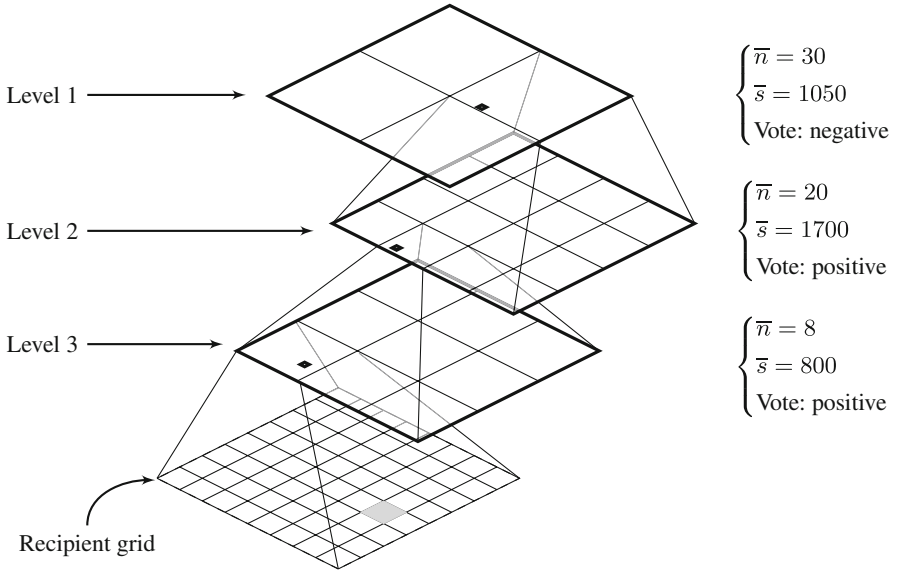


Fig. 3. Construction of heatmap at the data recipient site

recipient-defined rrg and threshold T. The objective of heatmap construction is to determine for each rrg cell a binary outcome: *positive* if the value derived for the cell is above T, and *negative* otherwise.

Figure 3 shows an example of rrg superimposed on the PSD index. The PSD has four levels, out of which only three are shown (the root is split into four cells, and it is omitted from the diagram due to space considerations). The bottom layer in the diagram represents the rrg. The shaded cell in the rrg layer represents the cell for which we are currently determining the outcome. In this example, we illustrated a high-resolution rrg, so most rrg cells are completely enclosed within a PSD cell at each index level. However, in general, there may be cases when a rrg cell overlaps with several PSD cells. We consider both cases below.

Since the recipient has no other information other than the PSD, we assume that the count and sum values inside a PSD cell are uniformly distributed over the cell's extent. Hence, for each rrg cell we compute n and s in proportion to the overlap between the rrg and PSD cells, normalized by the PSD cell area. If one rrg cell overlaps two or more PSD cells, the values for n and s are determined as the weighted sum of the values corresponding to each PSD cell, where the weight is represented by the overlap amount.

Note that, even if the above procedure may result in values for n and s for each rrg cell which are not too far apart from the actual values, there is another important source of inaccuracy due to the fact that the outcome for an rrg cell is obtained by dividing the noisy s and n values. The ratio can be significantly affected even if the noise is not very high. Furthermore, even though the leaf

cells of the PSD are likely to be closer in resolution to the rrg grid, considering solely leaf nodes in the outcome evaluation may have undesirable effects, due to the fact that the noise added to leaf nodes is more significant compared to their actual values compared to PSD nodes that are higher in the hierarchy (i.e., relative errors are higher closer to the leaf level).

In our solution, we account for these factors. Instead of naïvely dividing estimates for n and s in each rrg grid cell (which may have low accuracy), we evaluate individually the outcome based on information at each PSD level, and then combine the outcomes through a voting process in order to determine the outcome for each individual rrg cell. Returning to the example in Fig. 3, assume that threshold $T = 80$. We determine the outcome of the gray cell at the rrg layer by using the outcomes for all the marked PSD cells on the three levels shown (cells are marked using a small black square). Specifically, the Level 1 PSD cell containing the shaded grid cell has $\overline{n} = 30$ and $\overline{s} = 1050$, resulting in a phenomenon value $\overline{\rho} = \frac{\overline{s}}{\overline{n}} = 35$, below the threshold $T = 80$. Hence, the root cell's vote would be negative, meaning that with the information from that layer, the grayed grid cell does not present an anomalous reading.

However, at Level 2 of the PSD, we have $\overline{n} = 20$ and $\overline{s} = 1700$, resulting in a value of 85, greater than the threshold. Hence, this layer will contribute a positive vote. Similarly, at Level 3, $\overline{n} = 8$ and $\overline{s} = 800$ which also results in a positive vote.

The resulting outcome for any rrg cell depends on the distribution of the votes it has received. We could use the difference between positive and negative votes, but this will report a biased result for grid cells overlapping multiple PSD cells at the same level. A better solution is to use the ratio of positive votes to the total votes. In our example, the grayed cell got two positive votes and a single negative one, hence it would be marked as anomalous.

An alternative approach is to use only the number of positive votes that have been received. For instance, a rrg cell would receive a positive outcome if at least two PSD cells vote positively. This approach has two advantages: first, it captures locality better than the previous strategy. If the region where the phenomenon has an anomalous value is small, majority voting would tend to flatten the heatmap at higher levels, and the sharp spike may be missed. The two-vote strategy, however, may correctly identify the spike if both the leaf level PSD and another level above vote positively. Second, the two-vote strategy may prevent false alarms, caused by small PSD cells that may receive a high amount of random noise. By having a second level confirm the reading, many of the false negatives are eliminated, as it is unlikely that two PSD cells at different levels that overlap each other both receive very high noise due to the Laplace mechanism.

6 Experiments

We evaluate experimentally the proposed technique for privacy-preserving detection of anomalous phenomena. We implemented a C prototype, and we ran our

experiments on an Intel Core i7-3770 3.4 GHz CPU machine with 8 GB of RAM running Linux OS. We first provide a description of the experimental settings used. Next, we evaluate the accuracy of our technique in comparison with benchmarks. Finally, we investigate the performance of our technique when varying fundamental system parameters.

Experimental Settings. We consider a square two-dimensional location space with size 100×100, and a phenomenon with range $M = 100$ and threshold $T = 80$. We consider between $10,000$ and $50,000$ mobile users (i.e., sensors), uniformly distributed over the location domain. The average non-anomalous phenomenon value is 20, and to simulate an anomaly we generate a Gaussian distribution of values with scale parameter 20, centered at a random focus point within the location domain.

We consider two benchmark techniques for comparison. The first method, denoted as *Uniform Grid (U)*, considers a single-level fixed-granularity regular grid. The parameters of the grid are chosen according to the calculations presented in the first part of Sect. 4. The second method, *Adaptive Grid (AG)*, implements the state-of-the-art technique for PSDs as introduced in [16]. Specifically, it uses a two-level grid, where the first grid granularity is chosen according to a fixed split as indicated in [16], whereas the second-level granularity is determined based on the data density in the first level.

Comparison with Competitor Methods. We measure the accuracy in detecting anomalous phenomena for the proposed tree-based technique (denoted as t) and the benchmarks U and AG when varying privacy budget ε. For fairness, we consider the *1-vote* decision variant, which is supported by all methods. Figure 4(a) shows that our technique (presented with two distinct depth settings) clearly outperforms both benchmarks with respect to the Jacard metric. The U and AG method are only able to achieve values around 0.1 or less. Furthermore, they are not able to make proper use of the available privacy budget, and sometimes accuracy decreases when ε increases. The reason for this behavior is that the procedure for grid granularity estimation proposed in [16] has some built-in constants that are only appropriate for specific datasets and query types. In our problem setting, the granularity of these choices increases when ε increases, and the noise injected offsets the useful information in each cell.

To validate the superiority of the proposed technique beyond the J metric, Fig. 4(b) and (c) provide visualization of the heatmap obtained for the U method and our technique, respectively (the heatmap obtained for AG is similar to that of U). The anomalous phenomenon in the real data is shown using the circle area (i.e., points inside the circle are above the threshold). The heatmap produced by the U method is dominated by noise, and indicates that there are small regions with above-the-threshold values randomly scattered over the data domain. In contrast, our technique accurately identifies a compact region that overlaps almost completely with the actual anomalous region. Furthermore, for the t technique we consider two distinct maximum depth settings, $d = 3$ and $d = 4$. We observe that, although both variants outperform the benchmarks, as the height of the structure increases, a potentially negative effect occurs due to

the fact that the privacy budget per level decreases. Hence, it is not advisable to increase too much the PSD depth.

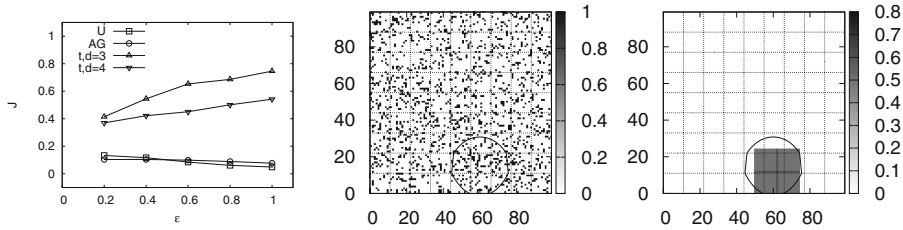
Both the UG and the AG method are unable to maintain data accuracy, and return virtually unusable data, without the ability to detect the occurrence of anomalous phenomena. In the rest of the experiments, we no longer consider competitor methods, and we focus on the effect of varying system parameters on the accuracy of the proposed technique. We also note that our method incurs low performance overhead, similar to that of the U method (between 2 and 4 s to sanitize and process the entire dataset). The AG method requires slightly longer, in the range of 15–20 s.

Effect of Varying System Parameters. We perform experiments to measure the accuracy of the proposed technique when varying fundamental system parameters, such as budget split parameters α, β and sensor count N.

Figure 5 shows the accuracy of our method when varying α, the budget split fraction across levels. Each graph illustrates several distinct combinations of budget ε and count-sum budget split β. For smaller α values, a smaller fraction of the budget is kept for the current level, with the rest being transferred for the children cells. Since the root node and the high levels of the tree have large spans, a smaller budget does not have a significant effect on accuracy, so it is best when a larger fraction is used in the lower-levels. For $\alpha = 0.2$, the proposed method reaches close to perfect J metric value.

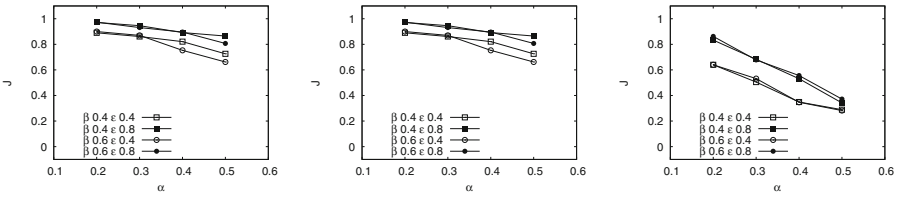
We also illustrate the effect of the various decision variants based on voting. Comparing Fig. 5(a) and (b), we can see that the accuracy increases slightly for the 2-vote scenario. This confirms that the 2-vote approach is able to filter out cases where some large outlier noise in one of the lower-level cells creates a false positive. On the other hand, the majority-voting strategy from Fig. 5(c) obtains lower accuracy, as it suffers from a relatively high false negative rate. Even if some of the levels signal an alarm, it is possible that a large amount of noise on several levels flips the outcome to "below the threshold". We conclude that the 2-vote strategy is the best available option.

Figure 6 shows the effect of varying parameter β, which decides the privacy budget split between the counts and sums in the PSD. Similar to previous results, we observe that the majority voting strategy has lower accuracy, due to the



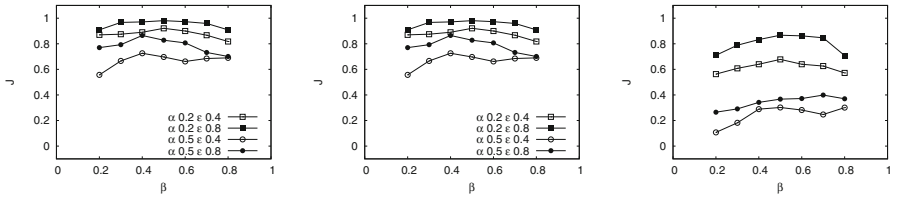
(a) Jacard metric, variable ε (b) Heatmap for U method (c) Heatmap for t method

Fig. 4. Accuracy evaluation in comparison with U and AG benchmarks.



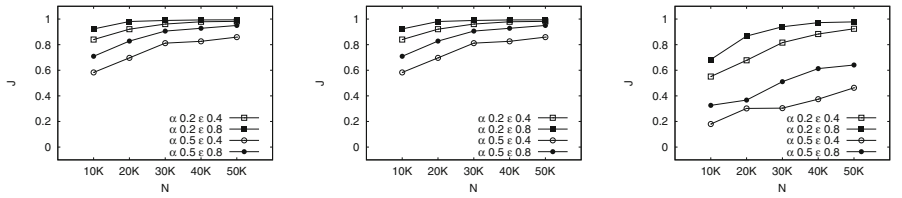
(a) 1 vote (b) 2 votes (c) 50% votes

Fig. 5. Impact of cross-level privacy budget split parameter α, $d = 3$



(a) 1 vote (b) 2 votes (c) 50% votes

Fig. 6. Impact of "count vs sum" privacy budget split parameter β, $d = 3$



(a) 1 vote (b) 2 votes (c) 50% votes

Fig. 7. Impact of number of mobile users N, $\beta = 0.5$, $d = 3$

increased occurrence of false negatives. The results also show that an equal split between counts and sums yields good results. As long as the β split is not severely skewed, the parameter does not significantly influence accuracy. However, when β is excessively low or high, one of the sum or count components gets very little budget, which causes large errors. In fact, this is one of the main reasons why competitor techniques fail to obtain good accuracy, as they do not consider the correlation between sum and count errors.

Finally, we consider the effect of varying number of sensors N. Figure 7 shows that the accuracy of the method increases slightly with N. This is expected, as a higher data density due to more reporting sensors benefits differential privacy, as the signal-to-noise ratio increases. In this case, we also notice a tendency of the majority voting strategy to underperform significantly compared to the 1-vote and 2-votes strategies.

7 Related Work

Collaborative sensing enables information extraction from a large number of wireless devices, spanning from smart phones to motes in a WSN. We focus on personal devices which are carried by users and may be used in sensing applications – from tracking to shapes-detection – in settings in which there are no WSNs available [11,15]. Such settings occur in many real-life applications in which the deployment of a WSN is either not possible or the WSN approach is not sustainable. We note that collaborative sensing is, in some sense, a broader paradigm than *participatory sensing* and *opportunistic sensing*, and when it comes to issues related to privacy protection, it subsumes the ones from the latter two paradigms in the risk of leaking personal/sensitive information [8]. While privacy-preserving computation has its history in domains such as cryptography and data mining, the existing methodologies cannot be straightforwardly mapped into the collaborative sensing applications.

Existing work addressed different aspects of the problem of detecting and representing spatial features of a particular monitored phenomenon [4,5]. Spatial summaries (e.g., isocontours [5]) may be constructed for energy-efficient querying. A natural trade-off is the precision of the aggregated representation vs the energy efficiency.

Location privacy has been studied extensively. Some techniques make use of cryptographic protocols such as private information retrieval [6]. Another category of methods focuses on location cloaking, e.g., using spatial k-anonymity [7,14], where a user hides among k other users. As discussed in Sect. 2, such techniques have serious security drawbacks. Closest to our work are the PSD construction techniques in [1,16,17]. As discussed in Sect. 4, these techniques are general-purpose, and our experimental evaluation shows that they are not suitable for anomalous phenomenon detection.

8 Conclusions

We proposed an accurate differentially-private technique for detection of anomalous phenomena in crowdsourced environmental sensing. Our solution consists of a PSD specifically-tailored to the requirements of phenomenon heatmap data, and strategies for flexible processing of sanitized datasets with values collected from mobile users. Experimental results show that the proposed technique is accurate, and clearly outperforms existing state-of-the-art in private spatial decompositions. In the future, we plan to extend our solution to continuous monitoring of phenomena, where multiple rounds of reporting are performed. This scenario is more challenging, as an adversary may correlate readings from multiple rounds to breach individual privacy.

References

1. Cormode, G., Procopiuc, C., Srivastava, D., Shen, E., Yu, T.: Differentially private spatial decompositions. In: ICDE, pp. 20–31 (2012)

2. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006)
3. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006)
4. Fayed, M., Mouftah, H.T.: Localised alpha-shape computations for boundary recognition in sensor networks. Ad Hoc Netw. **7**(6), 1259–1269 (2009)
5. Gandhi, S., Kumar, R., Suri, S.: Target counting under minimal sensing: complexity and approximations. In: Fekete, S.P. (ed.) ALGOSENSORS 2008. LNCS, vol. 5389, pp. 30–42. Springer, Heidelberg (2008)
6. Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C., Tan, K.L.: Private queries in location based services: anonymizers are not necessary. In: SIGMOD, pp. 121–132 (2008)
7. Gruteser, M., Grunwald, D.: Anonymous usage of location-based services through spatial and temporal cloaking. In: USENIX MobiSys (2003)
8. He, W., Liu, X., Nguyen, H.V., Nahrstedt, K., Abdelzaher, T.F.: PDA: privacy-preserving data aggregation for information collection. TOSN **8**(1), 6 (2011)
9. Li, N., Li, T., Venkatasubramanian, S.: T-closeness: privacy beyond k-anonymity and l-diversity. In: ICDE 2007, pp. 106–115. IEEE, Istanbul, Turkey (2007)
10. Li, N., Qardaji, W., Su, D., Cao, J.: Privbasis: frequent itemset mining with differential privacy. Proc. VLDB Endow. **5**(11), 1340–1351 (2012)
11. Li, W., Bao, J., Shen, W.: Collaborative wireless sensor networks: a survey. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Anchorage, Alaska, USA, 9–12 October 2011, pp. 2614–2619 (2011)
12. Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam, M.: l-diversity: privacy beyond k-anonymity. In: Proceedings of International Conference on Data Engineering (ICDE) (2006)
13. McSherry, F., Talwar, K.: Mechanism design via differential privacy. In: Proceedings of Annual IEEE Symposium on Foundations of Computer Science (FOCS), pp. 94–103 (2007)
14. Mokbel, M.F., Chow, C.Y., Aref, W.G.: The new casper: query processing for location services without compromising privacy. In: Proceedings of VLDB (2006)
15. Peralta, L.M.R., de Brito, L.M.P.L., Santos, J.F.F.: Improving users' manipulation and control on wsns through collaborative sessions. I. J. Knowl. Web Intell. **3**(3), 287–311 (2012)
16. Qardaji, W., Yang, W., Li, N.: Differentially private grids for geospatial data. In: Proceedings of IEEE Intlernational Conference on Data Engineering (ICDE) (2013)
17. Qardaji, W., Yang, W., Li, N.: Priview: practical differentially private release of marginal contingency tables. In: Proceedings of ACM SIGMOD (2014)
18. Samet, H.: The Design and Analysis of Spatial Data Structures. Addison-Wesley, Reading (1990)
19. Sweeney, L.: K-anonymity: a model for protecting privacy. Int. J. Uncertainty Fuzziness Knowl. Based Syst. **10**(5), 557–570 (2002)