Privacy-Preserving Outlier Detection for Data Streams

Jonas Böhler[1(✉)], Daniel Bernau[1], and Florian Kerschbaum[2]

[1] SAP Research, Karlsruhe, Germany
{jonas.boehler,daniel.bernau}@sap.com
[2] University of Waterloo, Waterloo, Canada
florian.kerschbaum@uwaterloo.ca

Abstract. In cyber-physical systems sensors data should be anonymized at the source. Local data perturbation with differential privacy guarantees can be used, but the resulting utility is often (too) low. In this paper we contribute an algorithm that combines local, differentially private data perturbation of sensor streams with highly accurate outlier detection. We evaluate our algorithm on synthetic data. In our experiments we obtain an accuracy of 80% with a differential privacy value of $\epsilon = 0.1$ for well separated outliers.

1 Introduction

In cyber-physical systems, e.g. smart metering, connected cars or the Internet of Things, sensors stream data to a sink, e.g. a database in the cloud, which is commonly controlled by a different entity. The subjects observed by the sensors have a vested interest in preserving their privacy towards the other entity. A technical means to preserve privacy is to anonymize the data (at the source). However, the data itself may be personally identifiable information as was, e.g., shown for smart meter readings [10]. Local data perturbation with differential privacy guarantees [5] can be used to protect against such exploitation and can be applied by the sensor. However, the resulting utility in this non-interactive model is often much lower than in the interactive, trusted curator model of differential privacy. So far, successful, differentially private outlier detection was only achieved in the interactive model [8,17,18].

Our algorithm contributed in this paper shows that local data perturbation of sensor streams combined with highly accurate outlier detection is feasible. We achieve this by using a relaxed version of differential privacy and a privacy-preserving correction method. The relaxation is to adapt the sensitivity to the set of data excluding the outliers [4]. We assume a scenario where outliers are subject to subsequent investigation which requires precise data, e.g. a broken power line or water pipe. Our privacy-preserving correction method uses distribution of trust between a correction server and an analyst server (the database). The correction server never learns the real measurements, but only the random

G. Livraga and S. Zhu (Eds.): DBSec 2017, LNCS 10359, pp. 225–238, 2017.
DOI: 10.1007/978-3-319-61176-1_12

noise added by the data perturbation (with indices of data values). The analyst server never learns the random noise, but only indices of data values whose outlier status – false positives and false negatives – the correction server has adjusted[1]. The result provides an improved outlier detection and preserves differential privacy towards the data analyst, since data perturbation is applied at the source (independent of the algorithm). Furthermore, the correction server never learns enough information to reconstruct any of the data.

Our non-interactive data perturbation is applied once for all subsequent analyses and does not require a privacy budget distributed over a series of queries which is critical in many applications [6,20]. We evaluate our algorithm on synthetic data. In our experiments we detect 80% of outliers in a subset of 10% of all points with a differential privacy value of $\epsilon = 0.1$ on data sets with well separated outliers. Our error correction method has an average runtime of less than 40 ms on 100,000 data points.

2 Related Work

We perform outlier detection on sensor data perturbed with relaxed differential privacy at the source and correct the detection errors due to perturbation. We are not aware of any related work on this specific problem, however, there has been extensive work in related areas: releasing differentially private topographical information, relaxations of differential privacy and separation of outliers and non-outliers.

In the area of releasing topological proximities under differential privacy a foundation for privately deriving cluster centers is provided in [1,16,21]. Their approaches have two drawbacks due to the use of the interactive model: The complex determination of ϵ for an assumed number of iterations until convergence and the limitation to aggregated cluster centers. An approach towards non-interactive differential privacy in clustering through a hybrid approach of non-interactive and interactive computations is formulated in [22]. A foundation for increasing differential privacy utility by sensitivity optimizations is introduced in [17]. Furthermore, the authors formulate a differentially private approach to release near optimal k-means cluster centers with their *sample-and-aggregate* framework. In [18] the sample-and-aggregate approach is extended to detect the minimal ball B enclosing approximately 90% of the points; everything outside B is presumed to be an outlier. Their approach is formulated in the interactive model and requires to apply the calculated ball B to the original data for outlier identification. The work in [8] is similar in the desire to produce a sanitized data set representation allowing an unlimited series of adaptive queries. Their non-interactive approach for producing private coresets (a weighted subset of points capturing geometric properties of its superset) suitable for k-means and k-median queries is proven theoretically efficient, but does not allow to identify

[1] The analyst server learns also parameters about the data set which are however computable from the output of the algorithm.

individual outliers. Chances for great accuracy improvements in differentially private analysis are identified in [18] if outliers were identified and removed before analysis.

Several relaxations for differential privacy have been suggested. Either by adapting the sensitivity [17], additional privacy loss [3], by distinguishing between groups with different privacy guarantees [11,14], or by relaxing the adversary [19,23]. In [3] (ϵ,δ)-differential privacy is presented where the privacy loss does not exceed ϵ with probability at most $1-\delta$ where δ is negligible. For our scenario in which outliers should be treated as a separate group, δ would become very large. Instead, we argue for a noise distribution with different ϵ guarantees for outliers and non-outliers. By relaxing the assumed adversary knowledge about the data the work [23] shows that utility gains in Genome-Wide Association Studies are achievable. This relaxation is not discriminating between different groups found in the data set as in our case. Additionally, we avoid relaxing the adversary and instead decrease the privacy guarantee for outliers.

The discussion on separation of outliers and non-outliers has been addressed in [11] by questioning the equal right for privacy for all (i.e. citizens vs. terrorists). Their work is close to ours in enforcing privacy guarantees to differentiate between a protected and a target subpopulation. However, in [11] the original data is maintained and query answers are perturbed interactively with a trusted curator. We avoid giving access to original data and enforce perturbation at the source. It is concluded in [15] that sparse domains incorporate a high risk of producing outliers in the perturbed data and thus argue for the need of outlier identification and removal in the unperturbed data set. In contrast, we preserve outliers and enable the detection of outliers in the perturbed data. Tailored differential privacy is defined in [14] and aims to provide *stronger* ϵ-differential privacy guarantees to outliers. We decided to evaluate the opposite by granting them *less* protection since we see outliers as faulty systems or sensors one needs to detect.

3 Preliminaries

We model a database (or data set) D as a collection of records from $\mathcal{D}^n$, i.e. $D \in \mathcal{D}^n$, where each entry D_i of D represents one participant's information. The *Hamming distance* $d_H(\cdot,\cdot)$ between two databases $x, y \in \mathcal{D}^n$ is $d_H(x,y) = |\{i : x_i \neq y_i\}|$, i.e. the number of entries in which they differ. Databases x, y are called *neighbors* or *neighboring* if $d_H(x,y) = 1$.

Definition 1 (Differential Privacy). *A perturbation mechanism $\mathcal{M}$ provides ϵ-differential privacy if for all neighboring databases D_1 and D_2, and all $S \subseteq Range(\mathcal{M})$,*

$$Pr[\mathcal{M}(D_1) \in S] \leq \exp(\epsilon) \cdot Pr[\mathcal{M}(D_2) \in S].$$

The protection for an individual in the database is measured by the privacy level ϵ. While a small ϵ offers higher protection for individuals involved in the

computation of a statistical function f, a larger ϵ offers higher accuracy on f. In case an individual is involved in a series of n statistical functions perturbed by a corresponding mechanism $\mathcal{M}_i$, where each function is requiring ϵ_i, her protection is defined as $\epsilon = \sum_{i=1}^{n} \epsilon_i$ by the basic sequential composition theorem of Dwork et al. [3,16]. A data owner can limit the privacy loss by specifying a maximum for ϵ called *privacy budget* [4]. Depending on the mutual agreement the exhaustion of the privacy budget can require the original data to be destroyed as mentioned in [20] since the privacy guarantee no longer holds.

The noise level of $\mathcal{M}$ in differential privacy is dependent on the *sensitivity* of f. For an overview of different notions of sensitivity with respect to the l_1-metric see Table 1. The global sensitivity GS_f of a function $f : \mathcal{D}^n \to \mathbb{R}^k$ determines the worst-case change that the omission or inclusion of a single individual's data can have on f. For example, if f is a counting query the removal of an individual can only change the result by 1. GS_f has to cover *all neighboring* databases, whereas local sensitivity $LS_f(D)$ covers *one fixed* database instance $D \in \mathcal{D}^n$ and all its neighbors. In certain cases databases with low local sensitivity can have neighbors with high local sensitivity, thereby allowing an attacker to distinguish them by the sensitivity-dependent noise magnitude alone. In contrast, smooth sensitivity $SS_f(D)$ compares a fixed database instance D with *all other database* instances but with respect to the distance between them and a privacy parameter β[2]. Using the notation from Table 1, the parameters that differ in the various notions are: allowed distance between neighboring databases D_1, D_2 (1 for GS_f and LS_f, unrestricted for SS_f) and choice of databases D_1 (a single fixed database instance for LS_f and SS_f, unrestricted for GS_f). In Sect. 4 we introduce a new notion of sensitivity, *relaxed sensitivity*, where the choice of databases is generalized to allow the selection of a subgroup of all possible databases.

Table 1. Comparison of different sensitivity notions

Global [4]	$GS_f = \max_{D_1, D_2 :\, d_H(D_1,D_2)=1;\; D_1, D_2 \in \mathcal{D}^n} \|f(D_1) - f(D_2)\|_1$
Local	$LS_f(D_1) = \max_{D_2 :\, d_H(D_1,D_2)=1;\; D_2 \in \mathcal{D}^n} \|f(D_1) - f(D_2)\|_1$
Smooth [17]	$SS_{f,\beta}(D_1) = \max_{D_2 \in \mathcal{D}^n} \left(LS_f(D_2) e^{-\beta d_H(D_1,D_2)} \right)$

Two models for computation of a mechanism $\mathcal{M}$ have been suggested by [5]. In the *interactive* model a data analyst receives noisy answers to functions evaluated on unperturbed data D as long as the privacy budget is not exhausted. In contrast, the original data D can be discarded in the *non-interactive* model by producing a sanitized version $D' = \mathcal{M}(D, f)$ of D and results are calculated with

[2] SS_f needs to be a *smooth upper bound* S as defined in [17], i.e. $\forall D \in \mathcal{D}^n : S(D) \geq LS_f(D)$ and $\forall D_1, D_2 \in \mathcal{D}^n, d_H(D_1, D_2) = 1 : S(D_1) \leq e^{\beta} \cdot S(D_2)$. These requirements can be fulfilled by $S(D) = GS_f$ with $\beta = 0$. SS_f, however, is the *smallest* function to satisfy these requirements with $\beta > 0$.

D'. While [17] assumes that the majority of mechanisms utilize the interactive model the findings of [2] suggest that D' is inefficient but potentially useful for many classes of queries if computational constraints are ignored. However, the non-interactive model also has its benefits: First, there is no need for a curator who requires access to the sensitive D, analyzes and permits queries and adjusts the privacy budget. Second, storage constrained sensors do not need to retain D and instead release a locally sanitized D'. Third, the data owner is not left with the administrative decision on how to handle exhausted privacy budgets (e.g. destroy D or refresh budget periodically as discussed in [16,21]).

4 Relaxed Differential Privacy

Differential privacy is a strong privacy guarantee due to its two worst-case assumptions: the adversary is assumed to have complete knowledge about the data set except for a single record and all possible data sets are covered by the guarantee. To relax differential privacy one has to relax these assumptions. The first assumption was relaxed in [19] by using a weaker but more realistic adversary and bounding the adversary's prior and posterior belief. In [17] the second assumption is relaxed by their notion of smooth sensitivity. We focused on the latter approach due to the fact that we are concerned with the discovery of outliers: We do not need the guarantee to hold for all records equally.

4.1 Relaxed Sensitivity

Our following new notion of *relaxed sensitivity* allows for different privacy guarantees for groups $\mathcal{N}$ (non-outliers), $\mathcal{O}$ (outliers) within a single dataset $\mathcal{D}^n = \mathcal{N} \cup \mathcal{O}$.

Definition 2 (Relaxed sensitivity). *Let $\mathcal{D}^n = \mathcal{N} \cup \mathcal{O}$ then the* relaxed sensitivity *of a function $f : \mathcal{D}^n \to \mathbb{R}^k$ is*

$$RS_f^{\mathcal{N},\mathcal{D}^n} = \max_{\substack{D_1,D_2 \in \mathcal{N} \\ D_2 :\, d_H(D_1,D_2)=1}} \|f(D_1) - f(D_2)\|_1.$$

In the following we abuse notation slightly and say that in the case that $\mathcal{D}$ consists of multiple, independent columns the sensitivity and perturbation are calculated per column and not the entire database at once. While local sensitivity only holds for *one fixed* database instance the relaxed sensitivity covers *all* databases from the subset $\mathcal{N}$. Let LS_f^X, GS_f^X denote local and global sensitivity respectively over a database set X.

Theorem 1. *Relaxed sensitivity compares to local and global sensitivity as follows:*

$$LS_f^{\mathcal{N}}(D) \leq RS_f^{\mathcal{N},\mathcal{D}^n} = GS_f^{\mathcal{N}} \leq GS_f^{\mathcal{D}^n}$$

where $D \in \mathcal{N} \subseteq \mathcal{D}^n$.

The proof is omitted due to space constraints. We will omit the dataset in the sensitivity notation in the following when it is not explicitly needed. The privacy guarantee is enforced by noise whose magnitude is controlled by privacy parameter ϵ and the sensitivity. We adapt the popular Laplace mechanism [5] to allow its invocation with different sensitivity notions.

Definition 3 Laplace mechanism. *Given any function $f : \mathcal{D}^n \to \mathbb{R}^k$, the Laplace mechanism is defined as*

$$\mathcal{M}_L(x, f(\cdot), GS_f/\epsilon) = f(x) + (Y_1, \ldots, Y_k)$$

where Y_i are independent and identically distributed random variables drawn from the Laplace distribution Laplace(GS_f/ϵ).

To relax differential privacy, we will adapt the scaling parameter to RS_f/ϵ, thus sampling noise from Laplace(RS_f/ϵ). We view a database as consisting of multiple columns and the perturbation is performed *per column*: The Laplace mechanism receives a column, i.e. a vector, as input and outputs a perturbed vector.

Theorem 2. *Let $\mathcal{D}^n = \mathcal{N} \cup \mathcal{O}$ and f be a function $f : \mathcal{D}^n \to \mathbb{R}^k$. The Laplace mechanism $\mathcal{M}_L(x, f, RS_f/\epsilon)$ preserves ϵ'-differential privacy for $x \in \mathcal{D}^n$ and preserves ϵ-differential privacy for $x \in \mathcal{N}$, where $\epsilon' = \epsilon \cdot GS_f/RS_f \geq \epsilon$.*

The proof is omitted due to space limitations.

4.2 Approximation of Relaxed Sensitivity

We do not want to restrict the queries an analyst can perform in the non-interactive model. Therefore, we choose to evaluate the *identity function* $f_{\text{id}}(x) = x$ for sensitivity determination. The sensitivity for f_{id} can be unbounded depending on the input domain. In the following we assume elements in $\mathcal{N}$ to be bounded real numbers. With f_{id} as our function and bounded $\mathcal{N}$, we can express the relaxed sensitivity as $RS_{f_{\text{id}}} = \max(\mathcal{N}) - \min(\mathcal{N})$, i.e. the gap between possible databases, that we seek to close. We see sensors measurements as point coordinates and use the terms interchangeably.

With historical data and domain knowledge the approximation can be tailored more precisely to individual data sets. With knowledge about what measurements can be considered physically possible one can approximate a bound for $\mathcal{N}$. We approximate $RS_{f_{\text{id}}}$ in the following with q-th percentiles ρ_q, $q \in [0, 100]$ of $\mathcal{D}$. We denote with p_o the percentage of outliers in the data set – alternatively, it can be seen as a bound on $\mathcal{N}$. We set $q_{\max} = 100 - p_o/2$, $q_{\min} = 100 - q_{\max}$, and approximate $RS_{f_{\text{id}}}$ with

$$\widehat{RS_{f_{\text{id}}}} = \rho_{q_{\max}}(\mathcal{D}) - \rho_{q_{\min}}(\mathcal{D}). \tag{1}$$

For this approximation we assume the following characteristics regarding our datasets:

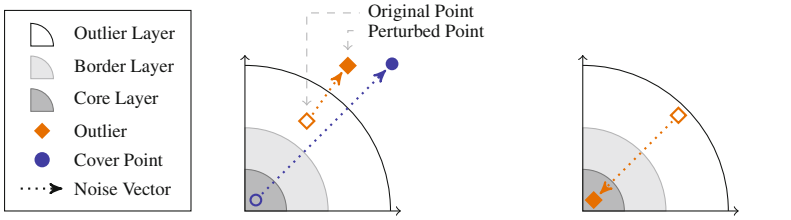
1. We define outliers as points on an outer layer surrounding non-outliers,
2. the percentage of outliers or a bound for $\mathcal{N}$ can be approximated,
3. the data set contains only one cluster.

Assumption 1 is also used in depth-based outlier detection algorithms. Regarding Assumption 2, one can learn the outlier percentage or bounds via historical data and the range of plausible (i.e. non-faulty, physically possible) measurements. If multiple clusters exists the data can be split in cluster groups thus fulfilling Assumption 3. The split is either performed by the data owner or a third party in a privacy-preserving manner (see Sect. 2 for clustering approaches consuming a portion of ϵ).

$\widehat{RS_f}$ implicitly defines $\mathcal{N}'$, which is an estimation of $\mathcal{N}$. An estimation $\widehat{RS_f} > RS_f$ leads to $\mathcal{N} \subset \mathcal{N}'$, i.e. more elements than necessary are protected which does not decrease privacy for elements from $\mathcal{N}$. However, $\widehat{RS_f} < RS_f$ implicitly defines $\mathcal{N}' \subset \mathcal{N}$, i.e. elements in $\mathcal{N} \backslash \mathcal{N}'$ could suffer a privacy loss since they receive less noise than needed. We want to stress that even for an inaccurate approximation, the non-outliers are still protected and receive a privacy level of $\epsilon' = \epsilon \cdot RS_{f_{\text{id}}} / \widehat{RS_{f_{\text{id}}}} \geq \epsilon$ (see Theorem 2). Furthermore, we correct errors introduced by the perturbation (or estimation $\widehat{RS_f} > RS_f$). For this we classify and detect the errors based on their change in distance to the center after and before perturbation as described in the following sections.

5 Outliers and False Negative Types

Let f_{outlier} be an outlier detection function $f_{\text{outlier}} : \mathcal{T} \rightarrow \{1, \ldots, |\mathcal{T}|\}$ which returns the indices (i.e. row numbers) of $\mathcal{T}$ which are outliers. We will refer to outliers detected in the unperturbed data set as *outliers* or $\mathcal{O} = f_{\text{outlier}}(\mathcal{T})$. When referring to the perturbed version of $\mathcal{T}$, denoted as $\mathcal{T}'$, we will use *presumed outliers* or $\mathcal{O}' = f_{\text{outlier}}(\mathcal{T}')$. We define outliers as points on an outer layer surrounding a (denser) core similar to [18]. Our goal is to find a small subset containing $\mathcal{O}$ on the perturbed data without having access to the original data.



(a) Legend (b) Outlier undetected due to perturbation of other points. (c) Outlier undetected due to its perturbation.

Fig. 1. Types of false negatives after perturbation. Layers correspond to unperturbed data points.

We perturb the data set with the adapted Laplace mechanism using approximated relaxed sensitivity per column. For well-separated outliers and non-outliers and with our relaxed sensitivity notion $\mathcal{O}$ and $\mathcal{O}'$ can be equal. However, this is not necessarily the case and therefore we present a correction algorithm in Sect. 6 to find the *false negatives* i.e. missing outliers from $\mathcal{O}$ that are not in $\mathcal{O}'$. The presumed outliers in $\mathcal{O}'$ can be separated in two sets: *false positives*, i.e. presumed outliers in $\mathcal{O}'$ that are not in $\mathcal{O}$ and *true positives*, i.e. outliers in $\mathcal{O}$ that are also in $\mathcal{T}'$.

We distinguish between two different types of false negatives visualized in Fig. 1. Non-outliers lie in the core layer, outliers in the outlier layer and the empty border layer separates the two. The layers for the unperturbed data differ from the perturbed layers which are omitted in Fig. 1. The two types of false negatives can occur as follows: First, a non-outlier can become a *cover point* after perturbation, i.e. "cover" a real outlier to produce a false negative as shown in Fig. 1b. Second, an outlier can also become a false negative on its own when it lands in a non-outlier region after perturbation, e.g. a dense core as in Fig. 1c, where it will not be detected in the perturbed data.

6 Relaxed Differentially Private Outlier Detection and Correction

Given the data $\mathcal{T}'$, a relaxed differentially private version of $\mathcal{T}$, we want to find the outliers corresponding to the unperturbed $\mathcal{T}$. We use the *semi-honest model* introduced in [9] where corrupted protocol participants do not deviate from the protocol but gather everything created during the run of the protocol. (e.g. message transcripts, temporary memory). Furthermore, we assume that only one participant can be corrupted; an alternative assumption is that participants do not share their knowledge. The assumption that parties do not share their knowledge is similar to the interactive model of differential privacy, i.e. different analysts do not collaborate by combining their privacy budgets.

In the following we view data sets as consisting of two columns, one column per measured attribute. Let $\mathcal{T}$, $\mathcal{T}'$ be the unperturbed resp. perturbed data set. For the perturbation each column receives independently drawn noise from the Laplace mechanism with approximated relaxed sensitivity. We use sets of *indices* corresponding to rows in $\mathcal{T}$ and $\mathcal{T}'$. This is convenient since an index identifies the same point before and after perturbation. Let I be the set of indices from $\mathcal{T}$ (and thereby also $\mathcal{T}'$). We denote with $\mathcal{T}[i]$ the row at index $i \in I$ and with $\mathcal{T}[, j]$ the selection of column j. Recall that we denote with $\mathcal{O}$ the set of all indices corresponding to rows with outliers in $\mathcal{T}$ detected by f_{outlier}, i.e. $\mathcal{O} = f_{\text{outlier}}(\mathcal{T})$, and with $\mathcal{O}'$ the set of all presumed outlier indices from $\mathcal{T}'$. As before, *false negatives* are missing outliers from $\mathcal{O}$ that are not found in $\mathcal{O}'$, *false positives* are presumed outliers in $\mathcal{O}'$ that are not in $\mathcal{O}$, and *true positives* are outliers in $\mathcal{O}$ that are also in $\mathcal{O}'$. We denote the Euclidean distance of two points c, x as $d(c, x) = d_c(x)$ where c is the center of $\mathcal{T}$, determined by averaging each column.

Definition 4 (Distance Difference d_{diff}). *The distance difference between a point in $\mathcal{T}$ and $\mathcal{T}'$ at index i and the center c after and before perturbation respectively is*

$$d_{diff}[i] = d_c(\mathcal{T}'[i]) - d_c(\mathcal{T}[i]).$$

We denote with $w_{\mathcal{O}}$ the width of the outlier layer (visualized in Fig. 1) in the unperturbed data $\mathcal{T}$. We denote with $\mathcal{FN}_{Lj}$ a set of indices for different layers $j \in \{1, 2, 3\}$ of presumed false negatives. These are spatial layers based on the false negative types of Fig. 1 used in the correction phase.

6.1 Correction Algorithm

Our goal is to detect presumed outliers on relaxed differentially private data $\mathcal{T}'$, find the undetected false negatives and remove additionally detected false positives. The algorithm presented in Fig. 2 operates as follows: Each sensor $\mathcal{S}$ has as input the data set $\mathcal{T}$, the privacy parameter ϵ and approximated relaxed sensitivities $\widehat{RS_{f_{\text{id}},j}}$ for each data column j. The correction server $\mathcal{CS}$ has as input the outlier layer width $w_{\mathcal{O}}$. The values for $\widehat{RS_{f_{\text{id}},j}}$ and $w_{\mathcal{O}}$ are determined with historical data, i.e. knowledge about past outliers, and bounds for the non-outliers, e.g. normal, non-faulty measurement values for sensors. $\mathcal{S}$ scales the data in line 1a and generates the perturbed data set $\mathcal{T}'$ via perturbation of each column j with the adapted Laplace mechanism parameterized with $\widehat{RS_{f_{\text{id}},j}}/\epsilon$. Then it sends $\mathcal{T}'$ to the analyst $\mathcal{A}$ and the distance differences d_{diff} to the correction server $\mathcal{CS}$. In line 3 the server $\mathcal{CS}$ filters $\mathcal{O}'$ in two sets $\mathcal{FP}$ and $\mathcal{TP}$ for presumed false positives and true positives respectively. The filtering is based on comparison of d_{diff} against a threshold – the distance difference with the biggest change between sorted distance differences. We use the fact that false positives, i.e. non-outliers that were detected as outliers in the perturbed set, have a higher distance difference, i.e. receive more noise, than true positives when non-outliers and outliers are well-separated. We do not want to remove true positives (actual outliers) under any circumstances. Thus, we err on the side of removing not enough false positives if the separation between outliers and non-outliers is low. In line 4 we reduce the set of indices to check for potential false negatives. Without the reduction true negatives can land in $\mathcal{FN}_{Lj}$, since they have the same distance difference (d_{diff}) but not the same core distance (d_c) as false negative candidates. The server $\mathcal{CS}$ detects false negatives in line 5, i.e. outliers not contained in $\mathcal{O}'$, in three spatial layers $\mathcal{FN}_{L1}$, $\mathcal{FN}_{L2}$, and $\mathcal{FN}_{L3}$, where the first corresponds to the false negative type from Fig. 1c and the latter to 1b. The use of two layers $\mathcal{FN}_{L2}$ and $\mathcal{FN}_{L3}$ for one false negative type is due to the row reduction and a simplification for $\mathcal{FN}_{L2}$ explained in the following.

The inequalities are based on the unperturbed position of outliers in respect to non-outliers (i.e. on an outer, less dense layer) and the distance difference after perturbation. The reasoning for $\mathcal{FN}_{L1}$ is that non-outliers, who are by definition closer to the center c, are distanced further away from c after perturbation due to the noise magnitude. Whereas outliers, who are already further away, can

Input: Each sensor $\mathcal{S}_{id}$ has data $\mathcal{T}$, privacy level ϵ and approximated relaxed sensitivities $\widehat{RS_{f_{\text{id}},j}}$ per column j. Correction server $\mathcal{CS}$ has outlier layer width $w_{\mathcal{O}}$ of sensor domain.

1. **Each sensor** $\mathcal{S}_{id}$
   (a) Scales each column $\mathcal{T}[,j]$: subtraction of mean and division of standard deviation.
   (b) Perturbs each column to $\mathcal{T}'[,j] = \mathcal{M}_L(\mathcal{T}[,j], f_{\text{id}}, \widehat{RS_{f_{\text{id}},j}}/\epsilon)$ and sends its id and perturbed data, i.e. $(id, \mathcal{T}')$, to analyst $\mathcal{A}$.
   (c) Calculates the data center c by averaging every dimension, calculates the distance differences $d_{\text{diff}} = d_c(\mathcal{T}') - d_c(\mathcal{T})$ and sends (id, d_{diff}) to correction server $\mathcal{CS}$.
2. **Analyst** $\mathcal{A}$ performs standard outlier detection on $\mathcal{T}'$ to get the list $\mathcal{O}'$ of presumed outliers indices and sends $(id, \mathcal{O}')$ to $\mathcal{CS}$.
3. **Correction Server** $\mathcal{CS}$
   (a) Calculates the threshold index t for the biggest change between ascending d_{diff} values of presumed outliers: $t = \arg\max_{j_k \in J} d_{\text{diff}}[j_{k+1}] - d_{\text{diff}}[j_k]$ where $J = \{j_1, j_2, \dots\}$ are the indices from $\mathcal{O}'$ sorted according to ascending d_{diff} values. (For convenience we define $j_{k+1} = j_k$ for $k + 1 > |J|$.)
   (b) Separates indices via t into false positives $\mathcal{FP} = \{i \in \mathcal{O}' \mid d_{\text{diff}}[i] > d_{\text{diff}}[t]\}$ and true positives $\mathcal{TP} = \mathcal{O}' \backslash \mathcal{FP}$ and calculates $d_{\mathcal{TP}} = \min_{i \in \mathcal{TP}} d_{\text{diff}}[i]$.
   (c) Sends $(id, d_{\mathcal{TP}}, d_{\mathcal{TP}} + w_{\mathcal{O}})$ to $\mathcal{A}$.
4. **Analyst** $\mathcal{A}$ creates

$$I_2 = \{i \in I \backslash \mathcal{O}' \mid d_c(\mathcal{T}'[i]) \geq d_{\mathcal{TP}}\},$$
$$I_3 = \{i \in I \backslash \mathcal{O}' \mid d_c(\mathcal{T}'[i]) \geq d_{\mathcal{TP}} + w_{\mathcal{O}}\},$$

   and sends (id, I_2, I_3) to $\mathcal{CS}$.
5. **Correction Server** $\mathcal{CS}$ creates false negatives layer sets

$$\mathcal{FN}_{L1} = \{i \in I \backslash \mathcal{O}' \mid d_{\text{diff}}[i] < 0\},$$
$$\mathcal{FN}_{L2} = \{i \in I_2 \mid 0 \leq d_{\text{diff}}[i] \leq d_{\mathcal{TP}}\},$$
$$\mathcal{FN}_{L3} = \{i \in I_3 \mid d_{\mathcal{TP}} \leq d_{\text{diff}}[i] \leq d_{\mathcal{TP}} + w_{\mathcal{O}}\}.$$

Output: $\mathcal{CS}$ outputs $(id, \mathcal{TP}, \mathcal{FN}_{L1}, \mathcal{FN}_{L2}, \mathcal{FN}_{L3})$ to $\mathcal{DO}$.

Fig. 2. Algorithm for correction of outlier detection.

reduce their distance to the center as seen in Fig. 1c. Hence, we look for indices i fulfilling $d_{\text{diff}}[i] < 0$. The idea behind $\mathcal{FN}_{L2}$ is a simplification that all outliers lie on the same "orbit" around the center (same center distance). In this case the minimal distance difference d_{diff} to become a true positive is the minimal d_{diff} one can find from the set of presumed true positives $\mathcal{TP}$, i.e. $d_{\mathcal{TP}} = \min_{t \in \mathcal{TP}}(d_{\text{diff}}[t])$. Only unperturbed outliers with a d_{diff} greater than $d_{\mathcal{TP}}$ could be detected as an outlier after perturbation. Hence, the remaining undetected have d_{diff} larger than 0 (otherwise they land in $\mathcal{FN}_{L1}$) but smaller than $d_{\mathcal{TP}}$. However, not all outliers do lie on the same orbit. Therefore, we collect in $\mathcal{FN}_{L3}$ indices with distance difference greater than $d_{\mathcal{TP}}$. We also know that no undetected outlier's distance difference can be greater than the distance difference of false positives

in $\mathcal{FP}$, i.e. $d_{\mathcal{FP}} = \min_{f \in \mathcal{FP}}(d_{\text{diff}}[f])$. More precise is the outlier layer width $w_{\mathcal{O}}$ (line 4).

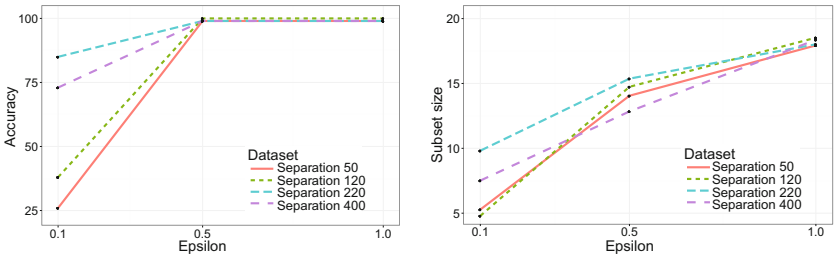
6.2 Privacy of the Correction Algorithm

The server $\mathcal{CS}$ knows the distance difference for points at a given index i, i.e. $d_{\text{diff}}[i]$, but $\mathcal{CS}$ does not know which perturbed point is identified by i since this knowledge remains at the analyst $\mathcal{A}$. The only information $\mathcal{CS}$ sends to $\mathcal{A}$ is $d_{\mathcal{TP}}, w_{\mathcal{O}}$, i.e. information regarding outliers which we like to detect. Even if $\mathcal{A}$ had access to all outlier information, including the noise used to perturb them, it would not lessen the protection of non-outliers. In exchange $\mathcal{A}$ sends $\mathcal{CS}$ I_{L2}, I_{L3}, i.e. sets of indices j of perturbed points with $d_c(\mathcal{T}'[j]) \geq d_{\mathcal{TP}}$ and $d_c(\mathcal{T}'[j]) \geq d_{\mathcal{TP}} + w_{\mathcal{O}}$ respectively. However, $\mathcal{CS}$ does not know which points correspond to these indices. If $\mathcal{A}$ and $\mathcal{CS}$ were to collaborate (i.e. not semi-honest) they could only narrow down the possible origin of a perturbed point. The information collaborating servers could learn can be reduced by using *frequency-hiding order-preserving encryption* as presented in [12] for the distance differences. Secure computation is not practical for our scenario (e.g. computation constrained IoT sensors) due to the bidirectional communication – although the communication complexity can be seen as almost independent of the network performance as shown in [13].

7 Outlier Detection Evaluation

We compared detected outliers on the original data with presumed outliers found on the perturbed data. Our algorithm was implemented in R 3.3.1 and run on a Apple MacBook Pro (Intel Core i7-4870HQ CPU, 16 GB main memory). We selected DBSCAN [7] to realize f_{outlier} and used the fpc package implementation. DBSCAN utilizes density and proximity of points, thus matching our spatial definition of the outlier topology where the outliers lie on an outer layer surrounding a denser core. DBSCAN is parameterized via *eps* for neighborhood reachability (point proximity and connections) and *minPts* (threshold density within the reachable neighborhood). We used the same DBSCAN parameters for the unperturbed and perturbed data. Our error correction logic only requires between 30 and 40 ms for 100,000 points. Four synthetic datasets were created to examine the impact of varying separation between outliers and non-outliers with the following characteristics: 100,000 points in $\mathbb{R}^2$ where each dimension is sampled independently from a normal distribution with standard deviation 3 and mean 0. Outlier percentage in the unperturbed data is 10%. After sampling the distances between outliers and non-outliers were increased to $\approx 50, 120, 220, 400$; we denote these data sets with *Separation 50*, etc. These separation distances were chosen based on decreasing probabilities that Laplace distributed noise preserves the outlier topology.

With *accuracy* we denote the percentage of all false negatives and true positives, i.e. all outliers, found with our approach. Furthermore, *subset size* is the



(a) Accuracy. (b) Subset size as percentage of all points.

Fig. 3. Accuracy and subset size for synthetic data sets; mean of 5 runs with $\epsilon \in \{0.1, 0.5, 1\}$.

size of the output of algorithm in Fig. 2, i.e. $\{\mathcal{TP}, \mathcal{FN}_{L1}, \mathcal{FN}_{L2}, \mathcal{FN}_{L3}\}$. Accuracy and subset size for $\epsilon \in \{0.1, 0.5, 1\}$ is shown in Fig. 3a and b respectively. The results for $\epsilon = 0.1$ indicate that our correction algorithm achieves meaningful accuracy of $\approx$75% for separation $\geq$220. However, $\epsilon = 0.1$ is a strong privacy guarantee and $\epsilon \in \{0.5, 1\}$ still offers meaningful protection. For $\epsilon = 0.5$ the found outlier percentage increases to 95–100% for all data sets. With a privacy guarantee of $\epsilon = 1$ our correction algorithm is not always needed since the separation between outliers and non-outliers is preserved even after perturbation. The subset size is always below 20% as is evident from Fig. 3b.

8 Conclusion

We implemented and evaluated an algorithm for detection of individual outliers on data perturbed in the local, non-interactive model of differential privacy, which is especially useful for IoT scenarios. We introduced a new notion of sensitivity, *relaxed sensitivity*, to provide different differential privacy guarantees for outliers in comparison to non-outliers. Furthermore, we presented a correction algorithm to detect false negatives and false positives. In our experiments we detect 80% of outliers in a subset of 10% of all points with a differential privacy value of $\epsilon = 0.1$ for data with well separated outliers.

Acknowledgments. This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 700294 (C3ISP) and 653497 (PANORAMIX).

References

1. Blum, A., Dwork, C., McSherry, F., Nissim, K.: Practical privacy: the SuLQ framework. In: Proceedings of the ACM Symposium on Principles of Database Systems (PODS) (2005)
2. Blum, A., Ligett, K., Roth, A.: A learning theory approach to noninteractive database privacy. J. ACM (JACM) **60**(2), 12 (2013)

3. Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M.: Our data, ourselves: privacy via distributed noise generation. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 486–503. Springer, Heidelberg (2006). doi:10.1007/11761679_29
4. 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). doi:10.1007/11681878_14
5. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. **9**(3–4), 211–407 (2014)
6. Erlingsson, Ú., Pihur, V., Korolova, A.: RAPPOR: randomized aggregatable privacy-preserving ordinal response. In: Proceedings of the Conference on Computer and Communications Security (CCS) (2014)
7. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD) (1996)
8. Feldman, D., Fiat, A., Kaplan, H., Nissim, K.: Private coresets. In: Proceedings of the ACM symposium on Theory of computing (STOC) (2009)
9. Goldreich, O.: Foundations of Cryptography. Basic Applications, vol. 2. Cambridge University Press, Cambridge (2009)
10. Jawurek, M., Johns, M., Kerschbaum, F.: Plug-in privacy for smart metering billing. In: International Symposium on Privacy Enhancing Technologies Symposium, pp. 192–210. Springer (2011)
11. Kearns, M., Roth, A., Wu, Z.S., Yaroslavtsev, G.: Privacy for the protected (only). ArXiv e-prints, May 2015
12. Kerschbaum, F.: Frequency-hiding order-preserving encryption. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 656–667. ACM (2015)
13. Kerschbaum, F., Dahlmeier, D., Schröpfer, A., Biswas, D.: On the practical importance of communication complexity for secure multi-party computation protocols. In: Proceedings of the 2009 ACM Symposium on Applied Computing, pp. 2008–2015. ACM (2009)
14. Lui, E., Pass, R.: Outlier privacy. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015. LNCS, vol. 9015, pp. 277–305. Springer, Heidelberg (2015). doi:10.1007/978-3-662-46497-7_11
15. Machanavajjhala, A., Kifer, D., Abowd, J., Gehrke, J., Vilhuber, L.: Privacy: theory meets practice on the map. In: Proceedings of the International Conference on Data Engineering (ICDE) (2008)
16. McSherry, F.: Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In: Proceedings of the ACM International Conference on Management of Data (SIGMOD) (2009)
17. Nissim, K., Raskhodnikova, S., Smith, A.: Smooth sensitivity and sampling in private data analysis. In: Proceedings of the ACM Symposium on Theory of Computing (STOC) (2007)
18. Nissim, K., Stemmer, U., Vadhan, S.: Locating a small cluster privately. In: Proceedings of the ACM Symposium on Principles of Database Systems (PODS) (2016)
19. Rastogi, V., Hay, M., Miklau, G., Suciu, D.: Relationship privacy: output perturbation for queries with joins. In: Proceedings of the ACM Symposium on Principles of Database Systems (PODS) (2009)
20. Roth, A.: New Algorithms for Preserving Differential Privacy. Ph.D. thesis, Carnegie Mellon University (2010)

21. Roy, I., Setty, S.T., Kilzer, A., Shmatikov, V., Witchel, E.: Airavat: security and privacy for mapreduce. In: Proceedings of the USENIX Conference on Networked Systems Design and Implementation (NSDI) (2010)
22. Su, D., Cao, J., Li, N., Bertino, E., Jin, H.: Differentially private k-means clustering. In: Proceedings of the ACM Conference on Data and Applications Security and Privacy (CODASPY) (2016)
23. Tramèr, F., Huang, Z., Hubaux, J.P., Ayday, E.: Differential privacy with bounded priors: reconciling utility and privacy in genome-wide association studies. In: Proceedings of the ACM Conference on Computer and Communications Security (CCS) (2015)