Random Sampling Plus Fake Data: Multidimensional Frequency Estimates With Local Differential Privacy

Héber H. Arcolezi
heber.hwang_arcolezi@univ-fcomte.fr
Femto-ST Institute, Univ. Bourg. Franche-Comté, CNRS
Belfort, France

Jean-François Couchot
jean-francois.couchot@univ-fcomte.fr
Femto-ST Institute, Univ. Bourg. Franche-Comté, CNRS
Belfort, France

Bechara Al Bouna
bechara.albouna@ua.edu.lb
TICKET Lab., Antonine University
Hadat-Baabda, Lebanon

Xiaokui Xiao
xkxiao@nus.edu.sg
School of Computing, National University of Singapore
Singapore, Singapore

ABSTRACT

With local differential privacy (LDP), users can privatize their data and thus guarantee privacy properties before transmitting it to the server (a.k.a. the aggregator). One primary objective of LDP is frequency (or histogram) estimation, in which the aggregator estimates the number of users for each possible value. In practice, when a study with rich content on a population is desired, the interest is in the multiple attributes of the population, that is to say, in multidimensional data ($d \geq 2$). However, contrary to the problem of frequency estimation of a single attribute (the majority of the works), the multidimensional aspect imposes to pay particular attention to the privacy budget. This one can indeed grow extremely quickly due to the composition theorem. To the authors' knowledge, two solutions seem to stand out for this task: 1) splitting the privacy budget for each attribute, i.e., send each value with $\frac{\epsilon}{d}$-LDP (*Spl*), and 2) random sampling a single attribute and spend all the privacy budget to send it with ϵ-LDP (*Smp*). Although *Smp* adds additional sampling error, it has proven to provide higher data utility than the former *Spl* solution. However, we argue that aggregators (who are also seen as attackers) are aware of the sampled attribute and its LDP value, which is protected by a "less strict" e^{ϵ} probability bound (rather than $e^{\epsilon/d}$). This way, we propose a solution named Random Sampling plus Fake Data (RS+FD), which allows creating *uncertainty* over the sampled attribute by generating fake data for each non-sampled attribute; RS+FD further benefits from amplification by sampling. We theoretically and experimentally validate our proposed solution on both synthetic and real-world datasets to show that RS+FD achieves nearly the same or better utility than the state-of-the-art *Smp* solution.

CCS CONCEPTS

• **Security and privacy → Privacy-preserving protocols**.

KEYWORDS

Local differential privacy, Multidimensional data, Frequency estimation, Sampling

ACM Reference Format:
Héber H. Arcolezi, Jean-François Couchot, Bechara Al Bouna, and Xiaokui Xiao. 2021. Random Sampling Plus Fake Data: Multidimensional Frequency Estimates With Local Differential Privacy. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3459637.3482467

1 INTRODUCTION

1.1 Background

In recent years, differential privacy (DP) [18, 19] has been increasingly accepted as the current standard for data privacy [1, 4, 20, 38]. In the centralized model of DP, a trusted curator has access to compute on the entire raw data of users (e.g., the Census Bureau [2, 26]). By 'trusted', we mean that curators do not misuse or leak private information from individuals. However, this assumption does not always hold in real life [33]. To address non-trusted services, with the local model of DP (LDP) [28], each user applies a DP mechanism to their own data before sending it to an untrusted curator (a.k.a. the aggregator). The LDP model allows collecting data in unprecedented ways and, therefore, has led to several adoptions by industry. For instance, big tech companies like Google, Apple, and Microsoft, reported the implementation of LDP mechanisms to gather statistics in well-known systems (i.e., Google Chrome browser [23], Apple iOS and macOS [39], and Windows 10 operation system [15]).

1.2 Problem statement

On collecting data, in practice, one is often interested in multiple attributes of a population, i.e., multidimensional data. For instance, in cloud services, demographic information (e.g., age, gender) and user habits could provide several insights to further develop solutions to specific groups. Similarly, in digital patient records, users might be linked with both their demographic and clinical information.

In this paper, we focus on the problem of private frequency (or histogram) estimation on multiple attributes with LDP. This is a primary objective of LDP, in which the data collector decodes all the privatized data of the users and can then estimate the number of users for each possible value. The single attribute frequency

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3482467

estimation task has received considerable attention in the literature [3, 5, 15, 23, 27, 32, 41, 48, 50] as it is a building block for more complex tasks (e.g., heavy hitter estimation [12, 13, 44], estimating marginals [24, 35, 37, 49], frequent itemset mining [36, 43]).

In the LDP setting, the aggregator already knows the users' identifiers, but not their private data. We assume there are d attributes $A = \{A_1, A_2, ..., A_d\}$, where each attribute A_j with a discrete domain $\mathcal{D}_j$ has a specific number of values $|A_j| = k_j$. Each user u_i for $i \in \{1, 2, ..., n\}$ has a tuple $\mathbf{v}^{(i)} = (v_1^{(i)}, v_2^{(i)}, ..., v_d^{(i)})$, where $v_j^{(i)}$ represents the value of attribute A_j in record $\mathbf{v}^{(i)}$. Thus, for each attribute A_j, the analyzer's goal is to estimate a k_j-bins histogram, including the frequency of all values in $\mathcal{D}_j$.

1.3 Context of the problem

Regarding multiple attributes, as also noticed in the recent survey work on LDP in [48], most studies for collecting multidimensional data with LDP mainly focused on numerical data (e.g., [17, 34, 40, 46]). Unlike the single attribute frequency estimation problem (the majority of the works), the multidimensional setting needs to consider the allocation of the privacy budget. To the authors' knowledge, there are mainly two solutions for satisfying LDP by randomizing $\mathbf{v}$. We will simply omit the index notation $\mathbf{v}^{(i)}$ in the analysis as we focus on one arbitrary user u_i here. On the one hand, due to the composition theorem [20], users can split the privacy budget for each attribute and send all randomized values with $\frac{\epsilon}{d}$-LDP to the aggregator (*Spl*). The other solution is based on random sampling a single attribute and spend all the privacy budget to send it (*Smp*). More precisely, each user tells the aggregator which attribute is sampled, and what is the perturbed value for it ensuring ϵ-LDP; the aggregator would not receive any information about the remaining $d - 1$ attributes.

Although the later *Smp* solution adds sampling error, in the literature [7, 34, 40, 41, 46], it has proven to provide higher data utility than the former *Spl* solution. However, aggregators (who are also seen as attackers) are aware of the sampled attribute and its LDP value, which is protected by a "less strict" e^{ϵ} probability bound (rather than $e^{\epsilon/d}$). In other words, while both solutions provide ϵ-LDP, we argue that using the *Smp* solution may be unfair with some users. For instance, on collecting multidimensional health records (i.e., demographic and clinical data), users that randomly sample a demographic attribute (e.g., gender) might be less concerned to report their data than those whose sampled attribute is "disease" (e.g., if positive for human immunodeficiency viruses - HIV).

This way, there is a privacy-utility trade-off between the *Spl* and *Smp* solutions. With these elements in mind, we formulate the problematic of this paper as: *For the same privacy budget ϵ, is there a solution for multidimensional frequency estimates that provides better data utility than Spl and more protection than Smp?*

1.4 Purpose and contributions

In this paper, we intend to solve the aforementioned problematic by answering the following question: *What if the sampling result (i.e., the selected attribute) was **not** disclosed with the aggregator?* Since the sampling step randomly selects an attribute $j \in [1, d]$ (we slightly abuse the notation and use j for A_j), we propose that users add uncertainty about the sampled attribute through generating $d - 1$ *fake data*, i.e., one for each non-sampled attribute.

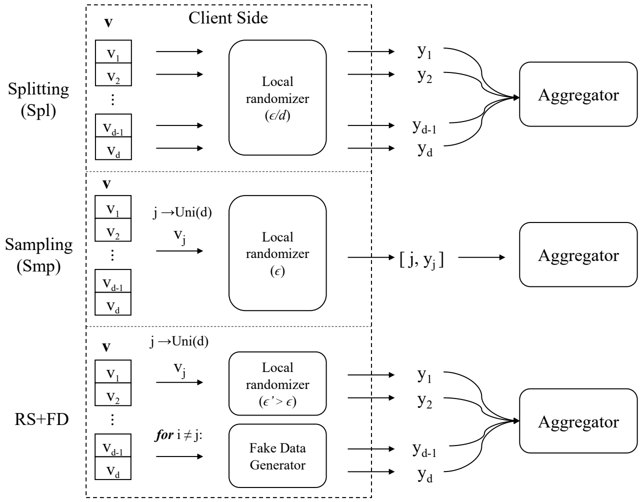

Figure 1: Overview of our random sampling plus fake data (RS+FD) solution in comparison with two known solutions, namely, *Spl* and *Smp*, where $Uni(d) = Uniform(\{1, 2, ..., d\})$.

We call our solution *Random Sampling plus Fake Data (RS+FD)*. Fig. 1 illustrates the overview of RS+FD in comparison with the aforementioned known solutions, namely, *Spl* and *Smp*. More precisely, with RS+FD, the client-side has two steps: local randomization and fake data generation. First, an LDP mechanism preserves privacy for the data of the sampled attribute. Second, the fake data generator provides fake data for each $d - 1$ non-sampled attribute. This way, the privatized data is "hidden" among fake data and, hence, the sampling result is not disclosed along with the users' report (and statistics).

What is more, we notice that RS+FD can enjoy privacy amplification by sampling [9, 10, 14, 28, 31]. That is, if one randomly sample a dataset without replacement using a sampling rate $\beta < 1$, it suffices to use a privacy budget $\epsilon' > \epsilon$ to satisfy ϵ-DP, where $\frac{e^{\epsilon'}-1}{e^{\epsilon}-1} = \frac{1}{\beta}$ [31]. This way, given that the sampled dataset for each attribute has non-overlapping users, i.e., each user selects an attribute with sampling probability $\beta = \frac{1}{d}$, to satisfy ϵ-LDP, each user can apply an LDP mechanism with $\epsilon' = \ln\left(d \cdot (e^{\epsilon} - 1) + 1\right) \geq \epsilon$.

To summarize, this paper makes the following contributions:

- We propose a novel solution, namely, RS+FD for multidimensional frequency estimates under LDP, which is generic to be used with any existing LDP mechanism developed for single-frequency estimation.
- Using state-of-the-art LDP mechanisms, we develop four protocols within RS+FD and analyze them analytically and experimentally.
- We conducted a comprehensive and extensive set of experiments on both synthetic and real-world datasets. Under the same privacy guarantee, results show that our proposed protocols with RS+FD achieve similar or better utility than using the state-of-the-art *Smp* solution.

Paper's outline: The remainder of this paper is organized as follows. In Section 2, we revise the privacy notions that we are considering, i.e., LDP, the LDP mechanisms we further analyze in this paper, and amplification by sampling. In Section 3, we introduce our RS+FD solution, the integration of state-of-the-art LDP mechanisms within RS+FD, and their analysis. In Section 4, we present experimental results. In Section 5, we discuss our results and review related work. Lastly, in Section 6, we present the concluding remarks and future directions.

2 PRELIMINARIES

In this section, we briefly recall LDP (Subsection 2.1), the LDP mechanisms we will apply in this paper (Subsection 2.2), and amplification by sampling (Subsection 2.3).

2.1 Local differential privacy

Local differential privacy, initially formalized in [28], protects an individual's privacy during the data collection process. A formal definition of LDP is given in the following:

Definition 1 (ϵ-Local Differential Privacy). *A randomized algorithm $\mathcal{A}$ satisfies ϵ-LDP if, for any pair of input values $v_1, v_2 \in Domain(\mathcal{A})$ and any possible output y of $\mathcal{A}$:*

$$\Pr[\mathcal{A}(v_1) = y] \leq e^{\epsilon} \cdot \Pr[\mathcal{A}(v_2) = y].$$

Similar to the centralized model of DP, LDP also enjoys several important properties, e.g., immunity to post-processing ($F(\mathcal{A})$ is ϵ-LDP for any function F) and composability [20]. That is, combining the results from m differentially private mechanisms also satisfies DP. If these mechanisms are applied separately in disjointed subsets of the dataset, $\epsilon = max(\epsilon_1\text{-}, \ldots, \epsilon_m)$-LDP (parallel composition). On the other hand, if these mechanisms are sequentially applied to the same dataset, $\epsilon = \sum_{i=1}^{m} \epsilon_i$-LDP (sequential composition).

2.2 LDP mechanisms

Randomized response (RR), a surveying technique proposed by Warner [47], has been the building block for many LDP mechanisms. Let $A_j = \{v_1, v_2, ..., v_{k_j}\}$ be a set of $k_j = |A_j|$ values of a given attribute and let ϵ be the privacy budget, we review two state-of-the-art LDP mechanisms for single-frequency estimation (a.k.a. frequency oracles) that will be used in this paper.

2.2.1 *Generalized randomized response (GRR).*

The k-Ary RR [27] mechanism extends RR to the case of $k_j \geq 2$ and is also referred to as direct encoding [41] or generalized RR (GRR) [43, 45, 49]. Throughout this paper, we use the term GRR for this LDP mechanism. Given a value $B = v_i$, GRR(v_i) outputs the true value v_i with probability $p = \frac{e^{\epsilon}}{e^{\epsilon}+k_j-1}$, and any other value v_l for $l \neq i$ with probability $q = \frac{1-p}{k_j-1} = \frac{1}{e^{\epsilon}+k_j-1}$. GRR satisfies ϵ-LDP since $\frac{p}{q} = e^{\epsilon}$.

The estimated frequency $\hat{f}(v_i)$ that a value v_i occurs, for $i \in [1, k_j]$, is calculated as [41, 43]:

$$\hat{f}(v_i) = \frac{N_i - nq}{n(p-q)}, \tag{1}$$

in which N_i is the number of times the value v_i has been reported and n is the total number of users. In [41], it is shown that this is an unbiased estimation of the true frequency, and the variance of this estimation is $Var[\hat{f}(v_i)] = \frac{q(1-q)}{n(p-q)^2} + \frac{f(v_i)(1-p-q)}{n(p-q)}$. In the case of small $f(v_i) \sim 0$, this variance is dominated by the first term. Thus, the *approximate variance* of this estimation for GRR is [41]:

$$Var[\hat{f}_{GRR}(v_i)] = \frac{e^{\epsilon} + k_j - 2}{n(e^{\epsilon} - 1)^2}. \tag{2}$$

2.2.2 *Optimized unary encoding (OUE).*

For a given value v, $B = Encode(v)$, where $B = [0, 0, ..., 1, 0, ...0]$, a k_j-bit array in which only the v-th position is set to one. Subsequently, the bits from B are flipped, depending on two parameters p and q, to generate a privatized vector B'. More precisely, $Pr[B'[i] = 1] = p$ if $B[i] = 1$ and $Pr[B'[i] = 1] = q$ if $B[i] = 0$. This unary-encoding (UE) mechanism satisfies ϵ-LDP for $\epsilon = ln\left(\frac{p(1-q)}{(1-p)q}\right)$ [23]. Wang et al. [41] propose optimized UE (OUE), which selects "optimized" parameters ($p = \frac{1}{2}$ and $q = \frac{1}{e^{\epsilon}+1}$) to minimize the *approximate variance* of UE-based mechanisms while still satisfying ϵ-LDP. The estimation method used in (1) equally applies to OUE. As shown in [41], the OUE *approximate variance* is calculated as:

$$Var[\hat{f}_{OUE}(v_i)] = \frac{4e^{\epsilon}}{n(e^{\epsilon} - 1)^2}. \tag{3}$$

2.2.3 *Adaptive LDP mechanism.*

Comparing (2) with (3), elements $k_j - 2 + e^{\epsilon}$ is replaced by $4e^{\epsilon}$. Thus, as highlighted in [41], when $k_j < 3e^{\epsilon} + 2$, the utility loss with GRR is lower than the one of OUE. Throughout this paper, we will use the term adaptive (ADP) to denote this best-effort and dynamic selection of LDP mechanism.

2.3 Privacy amplification by sampling

One well-known approach for increasing the privacy of a DP mechanism is to apply the mechanism to a random subsample of the dataset [9, 10, 14, 28, 31]. The intuition is that an attacker is unable to distinguish which data samples were used in the analysis. Li et al. [31, Theorem 1] theoretically prove this effect.

Theorem 1. ***Amplification by Sampling*** *[31]. Let $\mathcal{A}$ be an ϵ'-DP mechanism and $\mathcal{S}$ to be a sampling algorithm with sampling rate β. Then, if $\mathcal{S}$ is first applied to a dataset $\mathbb{D}$, which is later privatized with $\mathcal{A}$, the derived result satisfies DP with $\epsilon = \ln\left(1 + \beta(e^{\epsilon'} + 1)\right)$.*

3 RANDOM SAMPLING PLUS FAKE DATA

In this section, we present the overview of our RS+FD solution (Subsection 3.1), and the integration of the local randomizers presented in Subsection 2.2 within RS+FD (Subsections 3.2, 3.3, and 3.4).

3.1 Overview of RS+FD

We consider the local DP model, in which there are two entities, namely, users and the aggregator (an untrusted curator). Let n be the total number of users, d be the total number of attributes, $\mathbf{k} = [k_1, k_2, ..., k_d]$ be the domain size of each attribute, $\mathcal{A}$ be a local randomizer, and ϵ be the privacy budget. Each user holds a tuple $\mathbf{v} = (v_1, v_2, ..., v_d)$, i.e., a private value per attribute.

Client-Side. The client-side is split into two steps, namely, local randomization and fake data generation (cf. Fig. 1). Initially, each user samples a unique attribute j uniformly at random and applies an LDP mechanism to its value v_j. Indeed, RS+FD is generic to be applied with any existing LDP mechanisms (e.g., GRR [27], UE-based protocols [23, 41], Hadamard Response [3]). Next, for each $d-1$ non-sampled attribute i, the user generates one random fake data. Finally, each user sends the (LDP or fake) value of each attribute to the aggregator, i.e., a tuple $\mathbf{y} = (y_1, y_2, ..., y_d)$. This way, the sampling result is not disclosed with the aggregator. In summary, Alg. 1 exhibits the pseudocode of our RS+FD solution.

Aggregator. For each attribute $j \in [1, d]$, the aggregator performs frequency (or histogram) estimation on the collected data by removing bias introduced by the local randomizer and fake data.

Algorithm 1 Random Sampling plus Fake Data (RS+FD)

```
Input : tuple $\mathbf{v} = (v_1, v_2, ..., v_d)$, domain size of attributes $\mathbf{k} = [k_1, k_2, ..., k_d]$, privacy parameter $\epsilon$, local randomizer $\mathcal{A}$.
Output : privatized tuple $\mathbf{y} = (y_1, y_2, ..., y_d)$.
$\epsilon' = \ln(d \cdot (e^{\epsilon} - 1) + 1)$                ▹ amplification by sampling [31]
$j \leftarrow Uniform(\{1, 2, ..., d\})$                ▹ Selection of attribute to privatize
$B_j \leftarrow v_j$
$y_j \leftarrow \mathcal{A}(B_j, k_j, \epsilon')$                ▹ privatize data of the sampled attribute
for $i \in \{1, 2, ..., d\}/j$ do                ▹ non-sampled attributes
    $y_i \leftarrow Uniform(\{1, ..., k_i\})$                ▹ generate fake data
end for
return : $\mathbf{y} = (y_1, y_2, ..., y_d)$                ▹ sampling result is not disclosed
```

Privacy analysis. Let $\mathcal{A}$ be any existing LDP mechanism, Algorithm 1 satisfies ϵ-LDP, in a way that $\epsilon' = \ln(d \cdot (e^{\epsilon} - 1) + 1)$. Indeed, we observe that our scenario is equivalent to sampling a dataset $\mathbb{D}$ without replacement with sampling rate $\beta = \frac{1}{d}$ in the centralized setting of DP, which enjoys privacy amplification (cf. Subsection 2.3). With the local model, users privatize their data locally with a DP model. This way, to satisfy ϵ-LDP, an amplified privacy parameter $\epsilon' > \epsilon$ can be used.

Limitations. Similar to other sampling-based methods for collecting multidimensional data under LDP [17, 34, 40, 46], our RS+FD solution also entails *sampling error*, which is due to observing a sample instead of the entire population. In addition, in comparison with the *Smp* solution, RS+FD requires more computation on the user side because of the fake data generation part. Yet, communication cost is still equal to the *Spl* solution, i.e., each user sends one message per attribute. Lastly, while RS+FD utilizes an amplified $\epsilon' \geq \epsilon$, there is also bias generated from uniform fake data that may require a sufficient number of users n to eliminate the noise.

3.2 RS+FD with GRR

Client side. Integrating GRR as the local randomizer $\mathcal{A}$ into Alg. 1 (RS+FD[GRR]) requires no modification. Initially, on the client-side, each user randomly samples an attribute j. Next, the value v_j is privatized with GRR (cf. Subsection 2.2.1) using the size of the domain k_j and the privacy parameter ϵ'. In addition, for each non-sampled $d-1$ attribute i, the user also generates fake data uniformly at random according to the domain size k_i. Lastly, the user transmits the privatize tuple $\mathbf{y}$, which includes the LDP value of the true data "hidden" among fake data.

Aggregator RS+FD[GRR]. On the server-side, for each attribute $j \in [1, d]$, the aggregator estimates $\hat{f}(v_i)$ for the frequency of each value $i \in [1, k_j]$ as:

$$\hat{f}(v_i) = \frac{N_i d k_j - n(d - 1 + q k_j)}{n k_j (p - q)}, \tag{4}$$

in which N_i is the number of times the value v_i has been reported, $p = \frac{e^{\epsilon'}}{e^{\epsilon'} + k_j - 1}$, and $q = \frac{1-p}{k_j - 1}$.

Theorem 2. *For $j \in [1, d]$, the estimation result $\hat{f}(v_i)$ in* (4) *is an unbiased estimation of $f(v_i)$ for any value $v_i \in \mathcal{D}_j$.*

Proof 2

$$\begin{aligned} E[\hat{f}(v_i)] &= E\left[\frac{N_i d k_j - n(d - 1 + q k_j)}{n k_j (p - q)}\right] \\ &= \frac{d}{n(p-q)} E[Ni] - \frac{d - 1 + q k_j}{k_j (p - q)}. \end{aligned}$$

Let us focus on

$$\begin{aligned} E[N_i] &= \frac{1}{d}\left(pnf(v_i) + q(n - nf(v_i))\right) + \frac{d-1}{d k_j} n \\ &= \frac{n}{d}\left(f(v_i)(p - q) + q + \frac{d-1}{k_j}\right). \end{aligned}$$

Thus,

$$E[\hat{f}(v_i)] = f(v_i).$$

Theorem 3. *The variance of the estimation in* (4) *is:*

$$\begin{aligned} \mathrm{VAR}(\hat{f}(v_i)) &= \frac{d^2 \delta (1 - \delta)}{n(p-q)^2}, \text{ where} \\ \delta &= \frac{1}{d}\left(q + f(v_i)(p - q) + \frac{(d-1)}{k_j}\right). \end{aligned} \tag{5}$$

Proof 3

Thanks to (4) we have

$$\mathrm{VAR}(\hat{f}(v_i)) = \frac{\mathrm{VAR}(N_i) d^2}{n^2 (p-q)^2}. \tag{6}$$

Since N_i is the number of times value v_i is observed, it can be defined as $N_i = \sum_{z=1}^{n} X_z$ where X_z is equal to 1 if the user z, $1 \leq z \leq n$ reports value v_i, and 0 otherwise. We thus have $\mathrm{VAR}(N_i) = \sum_{z=1}^{n} \mathrm{VAR}(X_z) = n\,\mathrm{VAR}(X)$, since all the users are independent. According to the probability tree in Fig. 2,

$$P(X = 1) = P(X^2 = 1) = \delta = \frac{1}{d}\left(q + f(v_i)(p - q) + \frac{(d-1)}{k_j}\right).$$

We thus have $\mathrm{VAR}(X) = \delta - \delta^2 = \delta(1 - \delta)$ and, finally,

$$\mathrm{VAR}(\hat{f}(v_i)) = \frac{d^2 \delta (1 - \delta)}{n(p-q)^2}. \tag{7}$$

3.3 RS+FD with OUE

Client side. To use UE-based protocols (OUE in our work) as local randomizer $\mathcal{A}$ in Alg. 1, there is, first, a need to define the fake data generation procedure. We propose two solutions: (i) RS+FD[OUE-z] in Alg. 2, which applies OUE to $d-1$ zero-vectors, and (ii) RS+FD[OUE-r] in Alg. 3, which applies OUE to $d-1$ one-hot-encoded fake data (uniform at random). We note that, on the one hand, introducing fake data through privatizing zero-vectors introduces less noise as there is only one parameter to perturb each bit, i.e., $Pr[0 \rightarrow 1] = q$. On the other hand, starting with zero-vectors

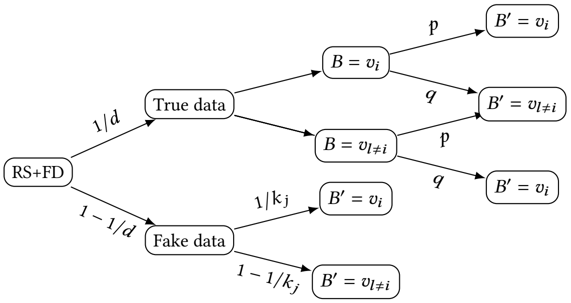

Figure 2: Probability tree for Eq. (4) **(RS+FD[GRR]).**

Algorithm 2 RS+FD[OUE-z]

```
Input : tuple $\mathbf{v} = (v_1, v_2, ..., v_d)$, domain size of attributes $\mathbf{k} = [k_1, k_2, ..., k_d]$, privacy parameter $\epsilon$, local randomizer OUE.
Output : privatized tuple $\mathbf{B}' = (B'_1, B'_2, ..., B'_d)$.
$\epsilon' = \ln(d \cdot (e^{\epsilon} - 1) + 1)$                ▷ amplification by sampling [31]
$j \leftarrow Uniform(\{1, 2, ..., d\})$                ▷ Selection of attribute to privatize
$B_j = Encode(v_j) = [0, 0, ..., 1, 0, ...0]$                ▷ one-hot-encoding
$B'_j \leftarrow OUE(B_j, \epsilon')$                ▷ privatize real data with OUE
for $i \in \{1, 2, ..., d\}/j$ do                ▷ non-sampled attributes
    $B_i \leftarrow [0, 0, ..., 0]$                ▷ initialize zero-vectors
    $B'_i \leftarrow OUE(B_i, \epsilon')$                ▷ randomize zero-vector with OUE
end for
return : $\mathbf{B}' = (B'_1, B'_2, ..., B'_d)$                ▷ sampling result is not disclosed
```

Algorithm 3 RS+FD[OUE-r]

```
Input : tuple $\mathbf{v} = (v_1, v_2, ..., v_d)$, domain size of attributes $\mathbf{k} = [k_1, k_2, ..., k_d]$, privacy parameter $\epsilon$, local randomizer OUE.
Output : privatized tuple $\mathbf{B}' = (B'_1, B'_2, ..., B'_d)$.
$\epsilon' = \ln(d \cdot (e^{\epsilon} - 1) + 1)$                ▷ amplification by sampling [31]
$j \leftarrow Uniform(\{1, 2, ..., d\})$                ▷ Selection of attribute to privatize
$B_j = Encode(v_j) = [0, 0, ..., 1, 0, ...0]$                ▷ one-hot-encoding
$B'_j \leftarrow OUE(B_j, \epsilon')$                ▷ privatize real data with OUE
for $i \in \{1, 2, ..., d\}/j$ do                ▷ non-sampled attributes
    $y_i \leftarrow Uniform(\{1, ..., k_i\})$                ▷ generate fake data
    $B_i \leftarrow Encode(y_i)$                ▷ one-hot-encoding
    $B'_i \leftarrow OUE(B_i, \epsilon')$                ▷ randomize fake data with OUE
end for
return : $\mathbf{B}' = (B'_1, B'_2, ..., B'_d)$                ▷ sampling result is not disclosed
```

may not suffice to "hide" the sampled attribute if the perturbation probability q is too small. Studying this effect is out of the scope of this paper and is left as future work.

Aggregator RS+FD[OUE-z]. On the server-side, if fake data are generated with OUE applied to zero-vectors, as in Alg. 2, for each attribute $j \in [1, d]$, the aggregator estimates $\hat{f}(v_i)$ for the frequency of each value $i \in [1, k_j]$ as:

$$\hat{f}(v_i) = \frac{d(N_i - nq)}{n(p-q)}, \tag{8}$$

in which N_i is the number of times the value v_i has been reported, n is the total number of users, $p = \frac{1}{2}$, and $q = \frac{1}{e^{\epsilon'}+1}$.

Theorem 4. *For $j \in [1, d]$, the estimation result $\hat{f}(v_i)$ in* (8) *is an unbiased estimation of $f(v_i)$ for any value $v_i \in \mathcal{D}_j$.*

Proof 4

$$E[\hat{f}(v_i)] = E\left[\frac{d(N_i - nq)}{n(p-q)}\right] = \frac{d(E[N_i] - nq)}{n(p-q)}$$
$$= \frac{d}{n(p-q)} E[N_i] - \frac{dq}{p-q}.$$

We have successively

$$E[N_i] = \frac{n}{d}\left(pf(v_i) + q(1 - f(v_i))\right) + \frac{(d-1)nq}{d}$$
$$= \frac{n}{d}\left(f(v_i)(p-q) + dq\right).$$

Thus,

$$E[\hat{f}(v_i)] = f(v_i).$$

Theorem 5. *The variance of the estimation in* (8) *is:*

$$\text{VAR}(\hat{f}(v_i)) = \frac{d^2\delta(1-\delta)}{n(p-q)^2}, \textit{ where}$$
$$\delta = \frac{1}{d}\left(dq + f(v_i)(p-q)\right). \tag{9}$$

The proof for Theorem 5 follows *Proof 3* and is omitted here. In this case, δ follows the probability tree in Fig. 3.

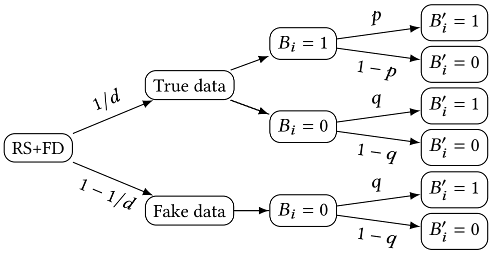

Figure 3: Probability tree for Eq. (8) **(RS+FD[OUE-z]).**

Aggregator RS+FD[OUE-r]. Otherwise, if fake data are generated with OUE applied to one-hot-encoded random data, as in Alg. 3, for each attribute $j \in [1, d]$, the aggregator estimates $\hat{f}(v_i)$ for the frequency of each value $i \in [1, k_j]$ as:

$$\hat{f}(v_i) = \frac{N_i d k_j - n\left[qk_j + (p-q)(d-1) + qk_j(d-1)\right]}{nk_j(p-q)}, \tag{10}$$

in which N_i is the number of times the value v_i has been reported, $p = \frac{1}{2}$, and $q = \frac{1}{e^{\epsilon'}+1}$.

Theorem 6. *For $j \in [1, d]$, the estimation result $\hat{f}(v_i)$ in* (10) *is an unbiased estimation of $f(v_i)$ for any value $v_i \in \mathcal{D}_j$.*

Proof 6

$$E[\hat{f}(v_i)] = E\left[\frac{N_i d k_j - n\left[qk_j + (p-q)(d-1) + qk_j(d-1)\right]}{nk_j(p-q)}\right]$$
$$= \frac{dE[N_i]}{n(p-q)} - \frac{(p-q)(d-1) + qdk_j}{k_j(p-q)}.$$

We have successively

$$
\begin{aligned}
E[N_i] &= \frac{n}{d}\left(pf(v_i) + q(1 - f(v_i))\right) + \frac{n(d-1)}{d}\left(\frac{p}{k_j} + \frac{k_j - 1}{k_j}q\right) \\
&= \frac{n}{d}\left(f(v_i)(p-q) + q\right) + \frac{n(d-1)}{dk_j}(p - q + k_j q).
\end{aligned}
$$

Thus,

$$E[\hat{f}(v_i)] = f(v_i).$$

Theorem 7. *The variance of the estimation in* (10) *is:*

$$
\begin{aligned}
\mathrm{VAR}(\hat{f}(v_i)) &= \frac{d^2\delta(1-\delta)}{n(p-q)^2}, \textit{ where} \\
\delta &= \frac{1}{d}\left(q + f(v_i)(p-q) + \frac{(d-1)}{k_j}\left(p + (k_j - 1)q\right)\right).
\end{aligned}
\tag{11}
$$

The proof for Theorem 7 follows *Proof 3* and is omitted here. In this case, δ follows the probability tree in Fig. 4.

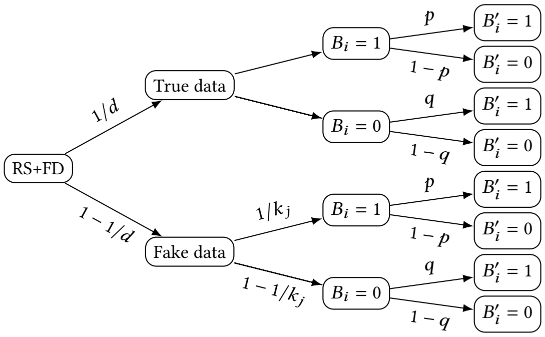

Figure 4: Probability tree for Eq. (10) **(RS+FD[OUE-r]).**

3.4 Analytical analysis: RS+FD with ADP

In a multidimensional setting with different domain sizes for each attribute, a dynamic selection of LDP mechanisms is preferred (as in Subsection 2.2.3). Because our estimators in (4), (8), and (10) are unbiased, their variance is equal to the mean squared error (MSE) that is commonly used in practice as an accuracy metric [42, 45, 46]. In this work, we analyze the *approximate variances* VAR_1 for RS+FD[GRR] in (5) and VAR_2 for RS+FD[OUE-z] in (9), in which $f(v_i) = 0$. This is, first, because under the local model of DP, the real frequency $f(v_i)$ is unknown and, second, because in real life the vast majority of values appear very infrequently. Notice that this is common practice in the literature (e.g., cf. [41, 46]), which provides an approximation for the variance of the protocols.

Assume there are $d \geq 2$ attributes with domain size $\mathbf{k} = [k_1, k_2, ..., k_d]$ and a privacy budget ϵ'. For each attribute j with domain size k_j, to select RS+FD[GRR], we are then left to evaluate if $\mathrm{VAR}_1 \leq \mathrm{VAR}_2$. This is equivalent to check whether,

$$\frac{d^2\delta_1(1-\delta_1)}{n(p_1-q_1)^2} - \frac{d^2\delta_2(1-\delta_2)}{n(p_2-q_2)^2} \leq 0, \tag{12}$$

in which $p_1 = \frac{e^{\epsilon'}}{e^{\epsilon'}+k_j-1}$, $q_1 = \frac{1-p_1}{k_j-1}$, $p_2 = \frac{1}{2}$, $q_2 = \frac{1}{e^{\epsilon'}+1}$, $\delta_1 = \frac{1}{d}\left(q_1 + \frac{d-1}{k_j}\right)$, and $\delta_2 = q_2$. **In other words, if** (12) **is less than or equal to zero, the utility loss is lower with RS+FD[GRR]; otherwise, if** (12) **is positive, RS+FD[OUE-z] should be selected. Throughout this paper, we will refer to this dynamic selection of our protocols as RS+FD[ADP].**

For the sake of illustration, Fig. 5 illustrates a 3D visualization of (12) by fixing $\epsilon' = \ln(3)$ and $n = 20000$, and by varying $d \in [2, 10]$ and $k_j \in [2, 20]$, which are common values for real-world datasets (cf. Subsection 4.1). In this case, one can notice in Fig. 5 that neither RS+FD[GRR] nor RS+FD[OUE-z] will always provide the lowest variance value, which reinforces the need for an adaptive mechanism. For instance, with the selected parameters, for lower values of k_j, RS+FD[GRR] can provide lower estimation errors even if d is large. On the other hand, as soon as the domain size starts to grow, e.g., $k_j \geq 10$, one is better off with RS+FD[OUE-z] even for small values of $d \geq 3$, as its variance in (9) does not depend on k_j.

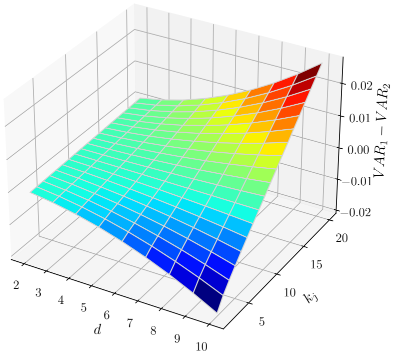

Figure 5: Analytical evaluation of (12) **that allows a dynamic selection between RS+FD[GRR] with variance** VAR_1 **and RS+FD[OUE-z] with variance** VAR_2**. Parameters were set as** $\epsilon' = \ln(3)$**,** $n = 20000$**,** $d \in [2, 10]$**, and** $k_j \in [2, 20]$**.**

4 EXPERIMENTAL VALIDATION

In this section, we present the setup of our experiments in Subsection 4.1, the results with synthetic data in Subsection 4.2, and the results with real-world data in Subsection 4.3.

4.1 Setup of experiments

Environment. All algorithms were implemented in Python 3.8.5 with NumPy 1.19.5 and Numba 0.53.1 libraries. The codes we developed and used for all experiments are available in a Github repository[1]. In all experiments, we report average results over 100 runs as LDP algorithms are randomized.

Synthetic datasets. Our first set of experiments are conducted on six synthetic datasets. The distribution of values in each attribute follows an uniform distribution, for all synthetic datasets.

[1] https://github.com/hharcolezi/ldp-protocols-mobility-cdrs

- For the first two synthetic datasets, we fix the number of attributes $d = 5$ and the domain size of each attribute as $\mathbf{k} = [10, 10, ..., 10]$ (uniform), and vary the number of users as $n = 50000$ and $n = 500000$.
- Similarly, for the third and fourth synthetic datasets, we fix the number of attributes $d = 10$ and the domain size of each attribute as $\mathbf{k} = [10, 10, ..., 10]$ (uniform), and vary the number of users as $n = 50000$ and $n = 500000$.
- Lastly, for the fifth and sixth synthetic datasets, we fix the number of users as $n = 500000$. Next, we set the number of attributes $d = 10$ with domain size of each attribute as $\mathbf{k} = [10, 20, ..., 90, 100]$ for one dataset, and we set the number of attributes $d = 20$ with domain size of each attribute as $\mathbf{k} = [10, 10, 20, 20, ..., 100, 100]$ for the other.

Real-world datasets. In addition, we also conduct experiments on four real-world datasets with non-uniform distributions.

- *Nursery.* A dataset from the UCI machine learning repository [16] with $d = 9$ categorical attributes and $n = 12960$ samples. The domain size of each attribute is $\mathbf{k} = [3, 5, 4, 4, 3, 2, 3, 3, 5]$, respectively.
- *Adult.* A dataset from the UCI machine learning repository [16] with $d = 9$ categorical attributes and $n = 45222$ samples after cleaning the data. The domain size of each attribute is $\mathbf{k} = [7, 16, 7, 14, 6, 5, 2, 41, 2]$, respectively.
- *MS-FIMU.* An open dataset from [6] with $d = 6$ categorical attributes and $n = 88935$ samples. The domain size of each attribute is $\mathbf{k} = [3, 3, 8, 12, 37, 11]$, respectively.
- *Census-Income.* A dataset from the UCI machine learning repository [16] with $d = 33$ categorical attributes and $n = 299285$ samples. The domain size of each attribute is $\mathbf{k} = [9, 52, 47, 17, 3, ..., 43, 43, 43, 5, 3, 3, 3, 2]$, respectively.

Evaluation and metrics. We vary the privacy parameter in a logarithmic range as $\epsilon = [\ln(2), \ln(3), ..., \ln(7)]$, which is within range of values experimented in the literature for multidimensional data (e.g., in [40] the range is $\epsilon = [0.5, ..., 4]$ and in [46] the range is $\epsilon = [0.1, ..., 10]$).

We use the MSE metric averaged per the number of attributes d to evaluate our results. Thus, for each attribute j, we compute for each value $v(i) \in \mathcal{D}_j$ the estimated frequency $\hat{f}(v_i)$ and the real one $f(v_i)$ and calculate their differences. More precisely,

$$MSE_{avg} = \frac{1}{d} \sum_{j \in [1,d]} \frac{1}{|\mathcal{D}_j|} \sum_{v \in \mathcal{D}_j} (f(v_i) - \hat{f}(v_i))^2. \qquad (13)$$

Methods evaluated. We consider for evaluation the following solutions (cf. Fig. 1) and protocols:

- Solution *Spl*, which splits the privacy budget per attribute ϵ/d with a best-effort approach using the adaptive mechanism presented in Subsection 2.2.3, i.e., Spl[ADP].
- Solution *Smp*, which randomly samples a single attribute and use all the privacy budget ϵ also with the adaptive mechanism, i.e., Smp[ADP].
- Our solution RS+FD, which randomly samples a single attribute and uses an amplified privacy budget $\epsilon' \geq \epsilon$ while generating fake data for each $d - 1$ non-sampled attribute:
  - RS+FD[GRR] (Alg. 1 with GRR as local randomizer $\mathcal{A}$);
  - RS+FD[OUE-z] (Alg. 2);
  - RS+FD[OUE-r] (Alg. 3);
  - RS+FD[ADP] presented in Subsection 3.4 (i.e., adaptive choice between RS+FD[GRR] and RS+FD[OUE-z]).

4.2 Results on synthetic data

Our first set of experiments were conducted on six synthetic datasets. Fig. 6 (first two synthetic datsets), Fig. 7 (third and fourth synthetic datsets), and Fig. 8 (last two synthetic datasets) illustrate for all methods, the averaged MSE_{avg} (y-axis) according to the privacy parameter ϵ (x-axis).

Impact of the number of users. In both Fig. 6 and Fig. 7, one can notice that the MSE_{avg} is inversely proportional to the number of users n. With the datasets we experimented, the MSE_{avg} decreases one order of magnitude by increasing n in one order of magnitude too. In comparison with *Smp*, the noise in our RS+FD solution comes mainly from fake data as it uses an amplified $\epsilon' \geq \epsilon$. *This suggests that, in some cases, with appropriately high number of user n, our solutions may always provide higher data utility than the state-of-the-art Smp solution (e.g., cf. Fig. 8).*

Impact of the number of attributes. One can notice the effect on increasing d comparing the results of Fig. 6 ($d = 5$) and Fig. 7 ($d = 10$) while fixing n and $\mathbf{k}$ (uniform number of values). For instance, even though there are twice the number of attributes, the accuracy (measured with the averaged MSE metric) does not suffer much. *This is because the amplification by sampling ($\frac{e^{\epsilon'}-1}{e^{\epsilon}-1} = \frac{1}{\beta}$ [31]) depends on the sampling rate $\beta = \frac{1}{d}$, which means that the more attributes one collects, the more the ϵ' is amplified, i.e., $\epsilon' = \ln(d \cdot (e^{\epsilon} - 1) + 1)$; thus balancing data utility.*

Besides, in Fig. 8, one can notice a similar pattern, i.e., increasing the number of attributes from $d = 10$ (left-side plot) to $d = 20$ (right-hand plot), with varied domain size $\mathbf{k}$, resulted in only a slightly loss of performance. This, however, is not true for the *Spl* solution, for example, in which the MSE_{avg} increased much more in order of magnitude than our RS+FD solution.

Comparison with existing solutions. From our experiments, one can notice that the *Spl* solution always resulted in more estimation error than our RS+FD solution and than the *Smp* solution, which is in accordance with other works [7, 34, 40, 41, 46]. Besides, our RS+FD[GRR], RS+FD[OUE-z], and RS+FD[ADP] protocols achieve better or nearly the same performance than the *Smp* solution with a best-effort adaptive mechanism Smp[ADP], which uses GRR for small domain sizes k and OUE for large ones. Although this is not true with RS+FD[OUE-r], it still provides more accurate results than Spl[ADP] while "hiding" the sampled attribute from the aggregator.

Globally, on high privacy regimes (i.e., low values of ϵ), our RS+FD solution consistently outperforms the other two solutions *Spl* and *Smp*. By increasing ϵ, Smp[ADP] starts to outperform RS+FD[OUE-r] while achieving similar performance than our RS+FD[GRR], RS+FD[OUE-z], and RS+FD[ADP] solutions. In addition, one can notice in Fig. 7, for example, the advantage of RS+FD[ADP] over our protocols RS+FD[GRR] and RS+FD[OUE-z] applied individually, as it adaptively selects the protocol with the smallest *approximate variance* value.

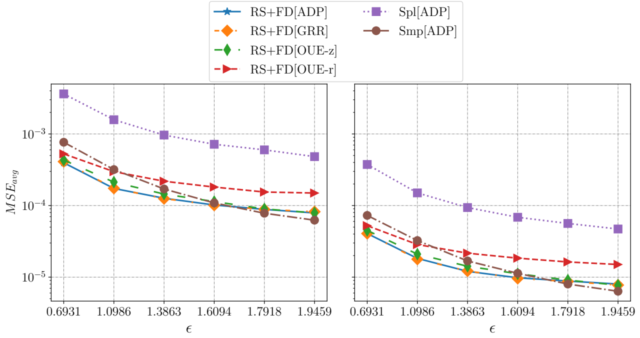

Figure 6: Averaged MSE varying ϵ on the *synthetic* datasets with $d = 5$, uniform domain size $\mathbf{k} = [10, 10, ..., 10]$, and $n = 50000$ (left-side plot) and $n = 500000$ (right-side plot).

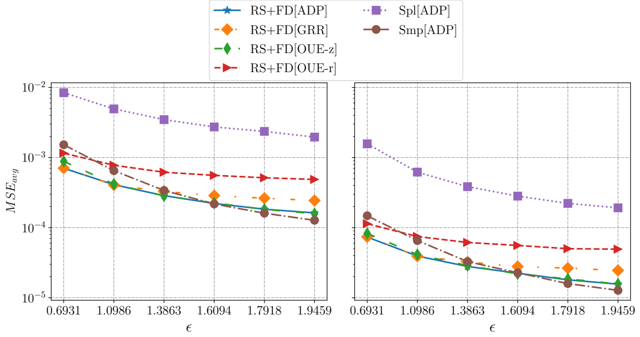

Figure 7: Averaged MSE varying ϵ on the *synthetic* datasets with $d = 10$, uniform domain size $\mathbf{k} = [10, 10, ..., 10]$, and $n = 50000$ (left-side plot) and $n = 500000$ (right-side plot).

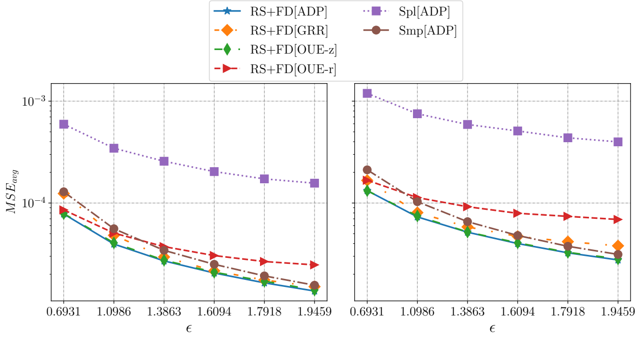

Figure 8: Averaged MSE varying ϵ on the *synthetic* datasets with $n = 500000$: the first with $d = 10$ and domain size $\mathbf{k} = [10, 20, ..., 90, 100]$ (left-side plot), and the other with $d = 20$ and domain size $\mathbf{k} = [10, 10, 20, ..., 100, 100]$ (right-side plot).

4.3 Results on real world data

Our second set of experiments were conducted on four real-world datasets with varied parameters for n, d, and $\mathbf{k}$. Fig. 9 (*Nursery*), Fig. 10 (*Adult*), Fig. 11 (*MS-FIMU*), and Fig. 12 (*Census-Income*) illustrate for all methods, averaged MSE_{avg} (y-axis) according to the privacy parameter ϵ (x-axis).

The results with real-world datasets follow similar behavior than with synthetic ones. For all tested datasets, one can observe that the MSE_{avg} of our proposed protocols with RS+FD is still smaller than the *Spl* solution with a best-effort adaptive mechanism Spl[ADP]. As also highlighted in the literature [7, 34, 40, 41, 46], privacy budget splitting is sub-optimal, which leads to higher estimation error.

On the other hand, for both *Adult* and *MS-FIMU* datasets, our solutions RS+FD[GRR], RS+FD[OUE-z], and RS+FD[ADP] achieve nearly the same performance (sometimes better on high privacy regimes, i.e., low ϵ) than the *Smp* solution with the best-effort adaptive mechanism Smp[ADP]. For the *Nursery* dataset, with small number of users n, only RS+FD[OUE-z] and RS+FD[ADP] are competitive with Smp[ADP]. Lastly, for the *Census* dataset, with a large number of attributes $d = 33$, increasing the privacy parameter ϵ resulted in a small gain on data utility for our solutions RS+FD[GRR] and RS+FD[OUE-r]. On the other hand, both of our solutions RS+FD[OUE-z] and RS+FD[ADP] achieve nearly the same or better performance than Smp[ADP].

Moreover, one can notice that using the *approximate variance* in (12) led RS+FD[ADP] to achieve similar or improved performance over our RS+FD[GRR] and RS+FD[OUE-z] protocols applied individually. For instance, for the *Adult* dataset, with RS+FD[ADP] it was possible to outperform Smp[ADP] 3x more than with RS+FD[GRR] or RS+FD[OUE-z] (similarly, 1x more for the *MS-FIMU* dataset). Besides, for the *Census-Income* dataset, RS+FD[ADP] improves the performance of the other protocols applied individually on high privacy regimes while accompanying the RS+FD[OUE-z] curve on the lower privacy regime cases.

In general, these results help us answering the problematic of this paper (cf. Subsection 1.3) that for the same privacy parameter ϵ, one can achieve nearly the same or better data utility with our RS+FD solution than when using the state-of-the-art *Smp* solution. Besides, RS+FD enhances users' privacy by "hiding" the sampled attribute and its ϵ-LDP value among fake data. On the other hand, there is a price to pay on computation, in the generation of fake data, and on communication cost, which is similar to the *Spl* solution, i.e., send a value per attribute.

5 LITERATURE REVIEW AND DISCUSSION

In recent times, there have been several works on the local DP setting in both academia [3, 5, 13, 27, 28, 34, 40, 41, 46, 48] and practical deployment [15, 23, 29, 39]. Among many other complex tasks (e.g., heavy hitter estimation [12, 13, 44], marginal estimation [24, 35, 37, 49], frequent itemset mining [36, 43]), frequency estimation is a fundamental primitive in LDP and has received considerable attention for a single attribute [3, 5, 8, 15, 23, 27, 30, 32, 41, 46, 48, 50]. However, concerning multiple attributes, as also noticed in the survey work on LDP in [48], most studies for collecting multidimensional data with LDP mainly focused on numerical data [17, 34, 40, 46]. For instance, in [34, 40], the authors propose sampling-based LDP mechanisms for real-valued data (named Harmony and Piecewise Mechanism) and applied these protocols in a multidimensional setting using state-of-the-art LDP mechanisms from [13, 41] for categorical data. On the other hand, regarding categorical attributes, in [41], the authors prove for OUE (as well as to optimal local hashing - OLH) that sending 1 attribute with

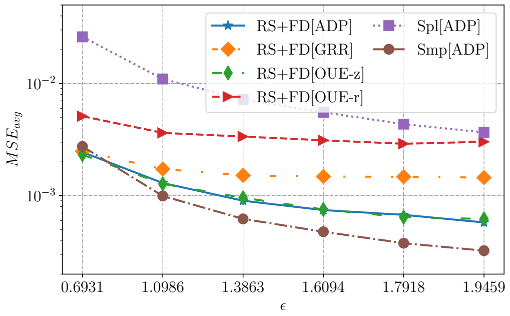

Figure 9: Averaged MSE varying ϵ on the *Nursery* dataset with $n = 12960$, $d = 9$, and domain size $\mathbf{k} = [3, 5, 4, 4, 3, 2, 3, 3, 5]$.

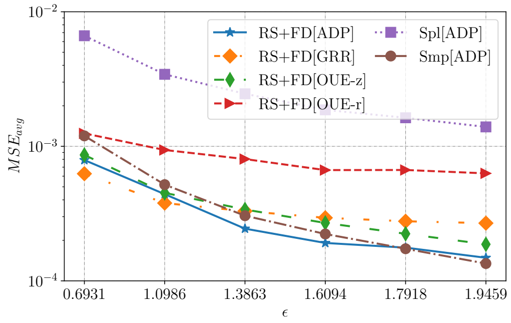

Figure 10: Averaged MSE varying ϵ on the *Adult* dataset with $n = 45222$, $d = 9$, and domain size $\mathbf{k} = [7, 16, 7, 14, 6, 5, 2, 41, 2]$.

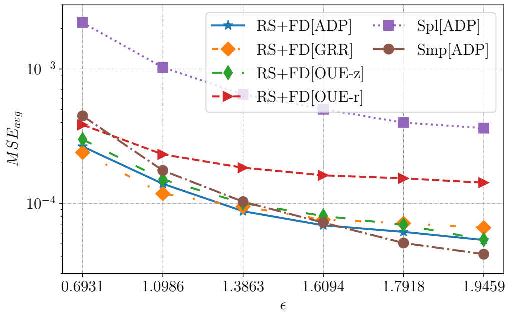

Figure 11: Averaged MSE varying ϵ on the *MS-FIMU* dataset with $n = 88935$, $d = 6$, and domain size $\mathbf{k} = [3, 3, 8, 12, 37, 11]$.

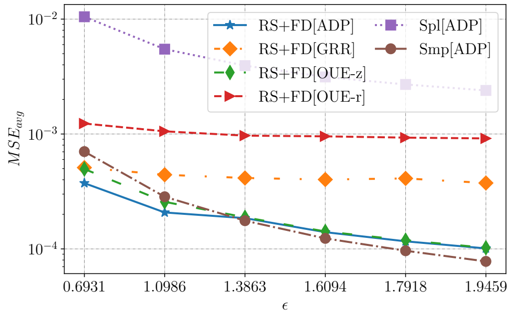

Figure 12: Averaged MSE varying ϵ on the *Census-Income* dataset with $n = 299285$, $d = 33$, and domain size $\mathbf{k} = [9, 52, 47, 17, 3, \ldots, 43, 43, 43, 5, 3, 3, 3, 2]$.

the whole privacy budget ϵ results in less variance than splitting the privacy budget ϵ/d for all attributes. The authors in [7] prove and validate experimentally that reporting a single attribute with ϵ-LDP resulted in less estimation error than splitting the privacy budget when using GRR.

However, in the aforementioned works [7, 34, 40, 41, 46], the sampling result is known by the aggregator. That is, each user samples a single attribute j, applies a local randomizer to v_j, and sends to the aggregator the tuple $y = \langle j, LDP(v_j) \rangle$ (i.e., *Smp*). While one can achieve higher data utility (cf. Figs. 6- 12) with *Smp* than splitting the privacy budget among d attributes (*Spl*), we argue that *Smp* might be "unfair" with some users. More precisely, users whose sampled attribute is socially "more" sensitive (e.g., disease or location), might hesitate to share their data as the probability bound e^{ϵ} is "less" restrictive than $e^{\epsilon/d}$. For instance, assume that GRR is used with k=2 (HIV positive or negative) and the privacy budget is $\epsilon = ln(7) \sim 2$, the user will report the true value with probability as high as $p \sim 87\%$ (even with $\epsilon = 1$, this probability is still high $p \sim 73\%$). On the other hand, if there are $d = 10$ attributes (e.g., nine demographic and HIV test), with *Spl*, the probability bound is now $e^{\epsilon/10}$ and $p \sim 55\%$.

Motivated by this privacy-utility trade-off between the solutions *Spl* and *Smp*, we proposed a solution named random sampling plus fake data (RS+FD), which generates uncertainty over the sampled attribute in the view of the aggregator. In this context, since the sampling step randomly selects an attribute with sampling probability $\beta = \frac{1}{d}$, there is an amplification effect in terms of privacy, a.k.a. amplification by sampling [9, 10, 14, 28, 31]. A similar privacy amplification for sampling a random item of a single attribute has been noticed in [43] for frequent itemset mining in the LDP model too. Indeed, *amplification* is an active research field on DP literature, which aims at finding ways to measure the privacy introduced by non-compositional sources of randomness, e.g., sampling [9, 10, 14, 28, 31], iteration [25], and shuffling [11, 21, 22, 42].

6 CONCLUSION AND PERSPECTIVES

This paper investigates the problem of collecting multidimensional data under ϵ-LDP for the fundamental task of frequency estimation. As shown in the results, our proposed RS+FD solution achieves nearly the same or better performance than the state-of-the-art *Smp* solution. In addition, RS+FD generates uncertainty over the sampled attribute in the view of the aggregator, which enhances users' privacy. For future work, we suggest and intend to investigate if given a reported tuple $\mathbf{y}$ one can state which attribute value is "fake" or not by seeing the estimated frequencies. Indeed, we intend to investigate this phenomenon in both single-time collection and longitudinal studies (i.e., throughout time) by extending RS+FD with two rounds of privatization, i.e., using *memoization* [15, 23].

ACKNOWLEDGMENTS

This work was supported by the Region of Bourgogne Franche-Comté CADRAN Project and by the EIPHI-BFC Graduate School (contract “ANR-17-EURE-0002"). All computations have been performed on the "Mésocentre de Calcul de Franche-Comté".

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy *(CCS '16)*. Association for Computing Machinery, New York, NY, USA, 308–318. https://doi.org/10.1145/2976749.2978318

[2] John M. Abowd. 2018. The U.S. Census Bureau Adopts Differential Privacy. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM. https://doi.org/10.1145/3219819.3226070

[3] Jayadev Acharya, Ziteng Sun, and Huanyu Zhang. 2019. Hadamard Response: Estimating Distributions Privately, Efficiently, and with Little Communication. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 89)*, Kamalika Chaudhuri and Masashi Sugiyama (Eds.). PMLR, 1120–1129.

[4] Ahmet Aktay et al. 2020. Google COVID-19 community mobility reports: Anonymization process description (version 1.0). *arXiv preprint arXiv:2004.04145* (2020).

[5] Mario Alvim, Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Anna Pazii. 2018. Invited Paper: Local Differential Privacy on Metric Spaces: Optimizing the Trade-Off with Utility. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*. IEEE. https://doi.org/10.1109/csf.2018.00026

[6] Héber H. Arcolezi, Jean-François Couchot, Oumaya Baala, Jean-Michel Contet, Bechara Al Bouna, and Xiaokui Xiao. 2020. Mobility modeling through mobile data: generating an optimized and open dataset respecting privacy. In *2020 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE. https://doi.org/10.1109/iwcmc48107.2020.9148138

[7] Héber H. Arcolezi, Jean-François Couchot, Bechara Al Bouna, and Xiaokui Xiao. 2021. Longitudinal Collection and Analysis of Mobile Phone Data with Local Differential Privacy. In *IFIP Advances in Information and Communication Technology*. Springer International Publishing, 40–57. https://doi.org/10.1007/978-3-030-72465-8_3

[8] Héber H. Arcolezi, Jean-François Couchot, Selene Cerna, Christophe Guyeux, Guillaume Royer, Béchara Al Bouna, and Xiaokui Xiao. 2020. Forecasting the number of firefighter interventions per region with local-differential-privacy-based data. *Computers & Security* 96 (Sept. 2020), 101888. https://doi.org/10.1016/j.cose.2020.101888

[9] Borja Balle, Gilles Barthe, and Marco Gaboardi. 2018. Privacy amplification by subsampling: tight analyses via couplings and divergences. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. 6280–6290.

[10] Borja Balle, Gilles Barthe, and Marco Gaboardi. 2020. Privacy profiles and amplification by subsampling. *Journal of Privacy and Confidentiality* 10, 1 (2020).

[11] Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. 2019. The Privacy Blanket of the Shuffle Model. In *Advances in Cryptology – CRYPTO 2019*. Springer International Publishing, 638–667. https://doi.org/10.1007/978-3-030-26951-7_22

[12] Raef Bassily, Kobbi Nissim, Uri Stemmer, and Abhradeep Thakurta. 2017. Practical locally private heavy hitters. *arXiv preprint arXiv:1707.04982* (2017).

[13] Raef Bassily and Adam Smith. 2015. Local, Private, Efficient Protocols for Succinct Histograms. In *Proceedings of the forty-seventh annual ACM symposium on Theory of Computing*. ACM. https://doi.org/10.1145/2746539.2746632

[14] Kamalika Chaudhuri and Nina Mishra. 2006. When Random Sampling Preserves Privacy. In *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 198–213. https://doi.org/10.1007/11818175_12

[15] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. 2017. Collecting Telemetry Data Privately. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 3571–3580.

[16] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[17] John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. 2018. Minimax Optimal Procedures for Locally Private Estimation. *J. Amer. Statist. Assoc.* 113, 521 (Jan. 2018), 182–201. https://doi.org/10.1080/01621459.2017.1389735

[18] Cynthia Dwork. 2006. Differential Privacy. In *Automata, Languages and Programming*. Springer Berlin Heidelberg, 1–12. https://doi.org/10.1007/11787006_1

[19] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography*. Springer Berlin Heidelberg, 265–284. https://doi.org/10.1007/11681878_14

[20] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3–4 (2014), 211–407.

[21] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Shuang Song, Kunal Talwar, and Abhradeep Thakurta. 2020. Encode, shuffle, analyze privacy revisited: Formalizations and empirical evaluation. *arXiv preprint arXiv:2001.03618* (2020).

[22] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. 2019. Amplification by Shuffling: From Local to Central Differential Privacy via Anonymity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 2468–2479. https://doi.org/10.1137/1.9781611975482.151

[23] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (Scottsdale, Arizona, USA). ACM, New York, NY, USA, 1054–1067. https://doi.org/10.1145/2660267.2660348

[24] Giulia Fanti, Vasyl Pihur, and Úlfar Erlingsson. 2016. Building a RAPPOR with the Unknown: Privacy-Preserving Learning of Associations and Data Dictionaries. *Proceedings on Privacy Enhancing Technologies* 2016, 3 (May 2016), 41–61. https://doi.org/10.1515/popets-2016-0015

[25] Vitaly Feldman, Ilya Mironov, Kunal Talwar, and Abhradeep Thakurta. 2018. Privacy Amplification by Iteration. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*. 521–532. https://doi.org/10.1109/FOCS.2018.00056

[26] Simson Garfinkel. 2021. Implementing Differential Privacy for the 2020 Census. USENIX Association.

[27] Peter Kairouz, Keith Bonawitz, and Daniel Ramage. 2016. Discrete distribution estimation under local privacy. In *International Conference on Machine Learning*. PMLR, 2436–2444.

[28] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2008. What Can We Learn Privately?. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*. IEEE. https://doi.org/10.1109/focs.2008.27

[29] Stephan Kessler, Jens Hoff, and Johann-Christoph Freytag. 2019. SAP HANA goes private. *Proceedings of the VLDB Endowment* 12, 12 (Aug. 2019), 1998–2009. https://doi.org/10.14778/3352063.3352119

[30] Jong Wook Kim, Dae-Ho Kim, and Beakcheol Jang. 2018. Application of Local Differential Privacy to Collection of Indoor Positioning Data. *IEEE Access* 6 (2018), 4276–4286. https://doi.org/10.1109/access.2018.2791588

[31] Ninghui Li, Wahbeh Qardaji, and Dong Su. 2012. On sampling, anonymization, and differential privacy or, k-anonymization meets differential privacy. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security - ASIACCS '12*. ACM Press. https://doi.org/10.1145/2414456.2414474

[32] Zitao Li, Tianhao Wang, Milan Lopuhaä-Zwakenberg, Ninghui Li, and Boris Škoric. 2020. Estimating Numerical Distributions under Local Differential Privacy. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. ACM. https://doi.org/10.1145/3318464.3389700

[33] David McCandless, Tom Evans, Miriam Quick, Ella Hollowood, Christian Miles, Dan Hampson, and Duncan Geere. 2021. World's Biggest Data Breaches & Hacks. https://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/. Online; accessed 11 March 2021.

[34] Thông T. Nguyên, Xiaokui Xiao, Yin Yang, Siu Cheung Hui, Hyejin Shin, and Junbum Shin. 2016. Collecting and Analyzing Data from Smart Device Users with Local Differential Privacy. *ArXiv* abs/1606.05053 (2016).

[35] Fan Peng, Shaohua Tang, Bowen Zhao, and Yuxian Liu. 2019. A privacy-preserving data aggregation of mobile crowdsensing based on local differential privacy. In *Proceedings of the ACM Turing Celebration Conference - China*. ACM. https://doi.org/10.1145/3321408.3321602

[36] Zhan Qin, Yin Yang, Ting Yu, Issa Khalil, Xiaokui Xiao, and Kui Ren. 2016. Heavy Hitter Estimation over Set-Valued Data with Local Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM. https://doi.org/10.1145/2976749.2978409

[37] Xuebin Ren, Chia-mu Yu, Weiren Yu, Shusen Yang, Senior Member, Xinyu Yang, Julie A Mccann, Philip S Yu, and Life Fellow. 2018. LoPub : High-Dimensional Crowdsourced Data. 13, 9 (2018), 2151–2166. https://doi.org/10.1109/TIFS.2018.2812146

[38] Ryan Rogers, Subbu Subramaniam, Sean Peng, David Durfee, Seunghyun Lee, Santosh Kumar Kancha, Shraddha Sahay, and Parvez Ahammad. 2020. LinkedIn's Audience Engagements API: A privacy preserving data analytics system at scale. *arXiv preprint arXiv:2002.05839* (2020).

[39] Apple Differential Privacy Team. 2017. Learning with privacy at scale. https://docs-assets.developer.apple.com/ml-research/papers/learning-with-privacy-at-scale.pdf. Online; accessed 11 March 2021.

[40] Ning Wang, Xiaokui Xiao, Yin Yang, Jun Zhao, Siu Cheung Hui, Hyejin Shin, Junbum Shin, and Ge Yu. 2019. Collecting and Analyzing Multidimensional Data with Local Differential Privacy. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE. https://doi.org/10.1109/icde.2019.00063

[41] Tianhao Wang, Jeremiah Blocki, Ninghui Li, and Somesh Jha. 2017. Locally Differentially Private Protocols for Frequency Estimation. In *26th USENIX Security*

Symposium (USENIX Security 17). USENIX Association, Vancouver, BC, 729–745.

[42] Tianhao Wang, Bolin Ding, Min Xu, Zhicong Huang, Cheng Hong, Jingren Zhou, Ninghui Li, and Somesh Jha. 2020. Improving utility and security of the shuffler-based differential privacy. *Proceedings of the VLDB Endowment* 13, 13 (Sept. 2020), 3545–3558. https://doi.org/10.14778/3424573.3424576

[43] Tianhao Wang, Ninghui Li, and Somesh Jha. 2018. Locally Differentially Private Frequent Itemset Mining. In *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE. https://doi.org/10.1109/sp.2018.00035

[44] Tianhao Wang, Ninghui Li, and Somesh Jha. 2021. Locally Differentially Private Heavy Hitter Identification. *IEEE Transactions on Dependable and Secure Computing* 18, 2 (March 2021), 982–993. https://doi.org/10.1109/tdsc.2019.2927695

[45] Tianhao Wang, Milan Lopuhaa-Zwakenberg, Zitao Li, Boris Skoric, and Ninghui Li. 2020. Locally Differentially Private Frequency Estimation with Consistency. In *Proceedings 2020 Network and Distributed System Security Symposium*. Internet Society. https://doi.org/10.14722/ndss.2020.24157

[46] Teng Wang, Jun Zhao, Zhi Hu, Xinyu Yang, Xuebin Ren, and Kwok-Yan Lam. 2021. Local Differential Privacy for data collection and analysis. *Neurocomputing* 426 (Feb. 2021), 114–133. https://doi.org/10.1016/j.neucom.2020.09.073

[47] Stanley L. Warner. 1965. Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias. *J. Amer. Statist. Assoc.* 60, 309 (March 1965), 63–69. https://doi.org/10.1080/01621459.1965.10480775

[48] Xingxing Xiong, Shubo Liu, Dan Li, Zhaohui Cai, and Xiaoguang Niu. 2020. A Comprehensive Survey on Local Differential Privacy. *Security and Communication Networks* 2020 (Oct. 2020), 1–29. https://doi.org/10.1155/2020/8829523

[49] Zhikun Zhang, Tianhao Wang, Ninghui Li, Shibo He, and Jiming Chen. 2018. CALM: Consistent adaptive local marginal for marginal release under local differential privacy. *Proceedings of the ACM Conference on Computer and Communications Security* (2018), 212–229. https://doi.org/10.1145/3243734.3243742

[50] Dan Zhao, Hong Chen, Suyun Zhao, Xiaoying Zhang, Cuiping Li, and Ruixuan Liu. 2019. Local Differential Privacy with K-anonymous for Frequency Estimation. In *2019 IEEE International Conference on Big Data (Big Data)*. IEEE. https://doi.org/10.1109/bigdata47090.2019.9006022