Journal of Information and Data Management, 2026, 17:1, doi: 10.5753/jidm.2026.5735

Locally Differentially Private Applications with Longitudinal Data

Antonio A. Marreiras Neto [**Universidade Federal do Ceará** | ***antonio.marreiras@lsbd.ufc.br***]
Eduardo R. Duarte Neto [**Universidade Federal do Ceará** | ***eduardo.rodrigues@lsbd.ufc.br***]
José S. Costa Filho [**Universidade Federal do Ceará** | ***serafim.costa@lsbd.ufc.br***]
Javam C. Machado [**Universidade Federal do Ceará** | ***javam.machado@lsbd.ufc.br***]

Laboratório de Sistemas e Banco de Dados, Universidade Federal do Ceará, Av. Humberto Monte - Pici, Fortaleza - CE, 60455-760, Brazil.

Received: 24 March 2025 • **Published:** 13 March 2026

Abstract Local differential privacy (LDP) was developed as a version of differential privacy (DP) that does not require a trusted curator or server. Frequency oracles, a class of LDP protocols for frequency estimation, function as the building blocks for diverse applications with LDP guarantees developed for tackling specific tasks such as answering range queries, and frequent item and itemset mining. However, these applications often build on frequency oracles with no adjustments for longitudinal data, and therefore can not provide LDP. In this paper, we investigate the practical effectiveness of state-of-the-art frequency oracle (FO) protocols designed for longitudinal data in various data analysis tasks. Specifically, we implement these protocols to perform three key tasks: answering range queries, identifying frequent items, and detecting frequent itemsets. Additionally, we incorporate post-processing techniques to enhance utility and improve overall performance. Our experimental evaluation includes four real-world datasets from diverse domains, allowing us to systematically measure and compare the utility of longitudinal LDP protocols.

Keywords: Local Differential Privacy, Longitudinal Data, Frequency Oracle Protocols

1 Introduction

Differential Privacy (DP) has come to be accepted as the *de facto* standard for data privacy. Nonetheless, as the originally proposed model of DP, central DP relies on a trusted curator [Dwork *et al.*, 2006], which is not a reliable assumption for real-life scenarios; research in the field has recently pivoted towards pursuing more restrictive models in a local setting to bypass the need for a trusted curator in the central model [Erlingsson *et al.*, 2014]. Said pursuits have led to the growing popularity of Local Differential Privacy (LDP), which aims to guarantee privacy in a local setting. In this setting, user data passes through an automated sanitization process immediately after sampling [Team, 2017; Ding *et al.*, 2017; Johnson *et al.*, 2018]. Thus, once the data reaches the server, it has already been processed in a way that guarantees the user's anonymity, even in the scenario of a data leak or a malicious curator.

However, as the LDP model requires noise to be added to each new user data sample, compliant protocols may add excessive amounts of noise, resulting in data that diverges significantly from the raw counterpart; subsequent analysis may yield inaccurate results. Said concerns become even more challenging when dealing with longitudinal data, as each new data sample the user sends to the server will be correlated with prior ones, it becomes a requirement for the mechanism to divide the privacy budget across queries into increasingly smaller fractions of the original target so LDP can still be guaranteed, leading to ever greater added noise and loss of data utility [Wang *et al.*, 2021a; Ren *et al.*, 2022]. As a means to lessen utility loss, a larger budget can be selected at the cost of user privacy [Dwork *et al.*, 2006]; however, as added noise will still increase with new samples.

There have been efforts to provide more flexible alternatives to achieve some form of LDP in a streaming scenario, including longitudinal LDP protocols [Erlingsson *et al.*, 2014; Arcolezi *et al.*, 2022a,b]. In previous research, L-LDP protocols have been applied to frequency counting, which has been used in localization and census scenarios. One challenge in said application has been how the protocols tend to behave as components of an LDP framework. Most LDP protocols are not used as standalone algorithms, but rather as components in applications developed for performing specific tasks with LDP guarantees. Said tasks can vary from identifying heavy hitters [Zhu *et al.*, 2024], to answering queries on geospatial data [Hong *et al.*, 2021]. As LDP protocols can showcase varying levels of utility, depending on the characteristics of the data being processed, and applications often will add privacy guarantees, requirements, and optimizations of their own [Li *et al.*, 2022; Filho and Machado, 2023], it is reasonable to expect that no LDP protocol will be best suited in all LDP applications.

An aggravating factor for L-LDP protocols is how often LDP applications are not developed with considerations for longitudinal data, and consequently are not tested or adapted for having L-LDP protocols as components. Not having an L-LDP protocol as a component leads to no privacy guarantees for longitudinal data.

Main contribution. This paper represents an extension of our previous work Marreiras *et al.* [2024], which presented a systematic and in-depth analysis of state-of-the-art frequency oracle (FO) protocols designed for longitudinal data. This work adds a new comprehensive evaluation of the previously determined best-performing FO protocols as components in local differential privacy (LDP) frameworks developed for specific tasks. We study frameworks that tackle the tasks of answering range queries, frequent items mining, and frequent itemset mining, under the longitudinal LDP setting. To assess the practical effectiveness of these protocols, we conduct extensive experiments using four real-world datasets. We rigorously compare their utility across various scenarios, offering valuable insights into their strengths and limitations.

Paper structure: The subsequent sections of this article are divided and presented in the following order: In Section 2, we present the required theoretical background for understanding the problem of interest to this paper. In Section 3, we describe our problem of interest in greater detail. In Section 4, we present basic LDP solutions that serve as building blocks for the protocols presented in Section 5, developed for longitudinal data and subjects of this paper's evaluation. In Section 6, we present applications designed for locally differentially private tasks that make use of LDP protocols as building blocks. Said applications will be the subject of our experiments in conjunction with the protocols presented in Section 5. In Section 7, we present the datasets and experimental setup used. In Section 8, we discuss the results, and through an analysis across increasing privacy budgets, we aim to determine the most adequate LDP Protocol for each application when processing longitudinal data. Section 9 briefly summarizes and highlights our most important and promising findings.

2 Theoretical background

2.1 Longitudinal Data

We define longitudinal data as data that evolves over time, captured through repeated sampling at increasing time intervals, which are represented as timestamps in the database. In this context, different individuals send a sample to the server at each timestamp. Therefore, the server aggregates the data where each row corresponds to all the data collected from timestamp t_0 to the current timestamp t_c for a single user.

2.2 Local Differential Privacy

Under LDP, sensitive information v from each user is encoded by a randomized algorithm Ψ, and its output $\Psi(v)$ is sent to the aggregator responsible for collecting all users' reports. Intuitively, LDP guarantees that, no matter what the value of $\Psi(v)$ is, it is approximately equally as likely to be a result of perturbing v as any other v' differing from v. Therefore, if $\Psi(v)$, instead of v, is collected, the users never reveal their private value v. The user's degree of privacy is controlled by the privacy budget ϵ. More formally,

Definition 1 *(Local Differential Privacy [Erlingsson* et al.*, 2014]) An algorithm $\Psi(\cdot)$ satisfies ϵ-local differential privacy (ϵ-LDP), where $\epsilon \geq 0$, if and only if for any pair of inputs (v, v'), and any possible output y of Ψ, we have $Pr[\Psi(v) = y] \leq e^{\epsilon} Pr[\Psi(v') = y]$*

For any pair of distinct inputs, an LDP mechanism has the probability to output the same value limited by e^{ϵ}. In the same fashion as central DP, LDP is robust to post-processing and sequential composition[Dwork *et al.*, 2014].

Post-Processing: post-processing is any function that receives the output of a ϵ-LDP mechanism as input, and regardless of which function it is, the output will remain ϵ-Locally Differentially Private, *i.e.,* if M is a ϵ-LDP mechanism, then $f(M)$ is also ϵ-LDP for any function f [Dwork *et al.*, 2014].

Sequential composition: if M_t is a ϵ_t-LDP mechanism, for $t \in [\tau]$. Then, the sequence of outputs $[M_1(v), ..., M_\tau(v)]$ is $\sum_{t=1}^{\tau} \epsilon_t$-LDP. Moreover, if M is an ϵ-LDP mechanism and v is a finite sequence of k values, then the sequence $[M(v_1), ..., M(v_k)]$ of outputs is $k\epsilon$-LDP [Dwork *et al.*, 2014].

3 Problem

We consider a setting where there are many users and one aggregator. Each user has a sequence $s = [v_1, v_2, ..., v_\tau]$ of values in a domain D, and the aggregator wants to learn the frequency distribution of values among all users for τ timestamps in a way that protects the privacy of individual users. More specifically, the aggregator wants to estimate, for each value $v \in D$, the fraction of users having v in each timestamp t *i.e.*, the number of users having v divided by the population size. We measure utility using the MSE averaged by the number of data collection τ, denoted by MSE_{avg}. Thus, for each time $t \in [1...\tau]$, we compute for each value $v \in D$ the estimated frequency $f(v)_t$ and the real one $\bar{f}(v)_t$ and calculate their differences before averaging by τ. More formally,

$$MSE_{avg} = \frac{1}{\tau} \sum_{t=1}^{\tau} \frac{1}{|D|} \sum_{v \in D} \left(\bar{f}(v)_t - f(v)_t \right)^2 \qquad (1)$$

Goal. We aim to understand the behavior of different LDP longitudinal data collection protocols as components in frameworks for particular tasks.

4 Frequency Oracle Protocols

A *frequency oracle (FO)* protocol can be used to estimate the frequency of any value $v \in D$ under LDP, where D is the domain. A FO consists of two algorithms. The first one is Ψ, which users use locally to perturb their private data. The second one is Φ, which the aggregator uses to estimate the frequencies regarding the perturbed data received. In the literature, FOs have been employed in many different LDP tasks, including marginal release [Liu *et al.*, 2023], answering range queries [Filho and Machado, 2023], answering queries on geospatial data [Hong *et al.*, 2021; Duarte Neto *et al.*, 2026], and identifying heavy hitters [Zhu *et al.*, 2024].

Traditional FOs do not account for budget consumption over time when processing longitudinal data. Still, most state-of-the-art protocols designed to tackle longitudinal scenarios are adaptations of traditional ones, usually through two rounds of sanitization, a technique accomplished by sequentially composing two traditional FOs and memoization. Below, we present two traditional FOs, which serve as the basis for the ones with two rounds of sanitization that are of interest to this paper.

4.1 Generalized Randomized Response (GRR)

Randomized Response [Warner, 1965] was introduced for binary responses, but it can easily be generalized to larger domains [Kairouz *et al.*, 2016]. In GRR, users send their true private value $v \in D$ with probability p. Otherwise, with probability $1 - p$, the users send a randomly chosen value $v' \in D$. Formally, the algorithm is

$$\forall_{x \in D} \; Pr\big[\Psi_{GRR_{(\epsilon)}}(v) = x\big] = \begin{cases} p & \text{if x = v} \\ q & \text{if x} \neq \text{v} \end{cases} \tag{2}$$

where

$$p = \frac{e^{\epsilon}}{e^{\epsilon} + |D| - 1}$$

$$q = \frac{1}{e^{\epsilon} + |D| - 1}$$

GRR satisfies ϵ-LDP since $p/q = e^{\epsilon}$. From a population of n users, the aggregator receives a vector $\mathbf{x} = \langle x_1, x_2, ...x_n \rangle$ of length $|\mathbf{x}| = n$ where $x_i \in D$ is the reported value of the i-th user. Then, it estimates the frequency of $v \in D$, which consists of the ratio of users with private value v among all n users. Considering $C(n)$ as the number of times v appears in vector $\mathbf{x}$, the unbiased [Wang *et al.*, 2017b] estimator for the frequency of $v \in D$ is

$$\Phi_{f(\epsilon)}(v) := (C(v)/n - q)/(p - q)$$

4.2 Unary Encoding (UE)

In Unary Encoding, a value $v \in D$ with domain size k is encoded as a length-k binary vector $B = [0,\cdots,0,1,0,\cdots,0]$ where only the v-th position is 1. The private mechanism returns a perturbed B' as

$$Pr\big[\Psi_{UE_{(\epsilon)}} B'[i] = 1\big] = \begin{cases} p, if B[i] = 1 \\ q, if B[i] = 0 \end{cases}$$

Wang *et al.* [2017b] has proven that UE satisfies ϵ-LDP for $\epsilon = ln\big(\frac{p(1-q)}{(1-p)q}\big)$. [Wang *et al.*, 2017b] defines two UE protocols: Symmetric Unary Encoding (SUE) and Optimized Unary Encoding (OUE).

SUE selects $p = \frac{e^{\epsilon/2}}{e^{\epsilon/2}+1}$ and $q = \frac{1}{e^{\epsilon/2}+1}$. It is described as symmetric since $p + q = 1$.

OUE sets $p = 1/2$ and $q = \frac{1}{e^{\epsilon}+1}$. It is considered a better option than SUE.

Both make use of the same unbiased estimator as GRR.

5 FOs for longitudinal data

Traditional FOs are inadequate for longitudinal data due to increased budget consumption and decreased user privacy. Many modern solutions use *memoization*, where a value is first sanitized, memoized, and then sanitized again with a fraction of the original budget for extra protection. Popularized by RAPPOR and adapted across various protocols[Arcolezi *et al.*, 2022a], the *2-round* memoization approach is the most accepted and will be the focus of our evaluation.

However, as proven in Arcolezi *et al.* [2022b], most influential works, such as RAPPOR, claim to be able to guarantee LDP by making bold assumptions about the data[Erlingsson *et al.*, 2014], which is not always realistic. That is why we will be adhering to a relaxed definition of LDP:

Definition 2 *(Longitudinal Local Differential Privacy [Arcolezi* et al.*, 2022b]) For a longitudinal memoizing mechanism $M : A^{\tau} \to B^{\tau}$, in which $A = [1..k]$, let M^* denote a mechanism that takes as input a permutation x of A and outputs $M^*(x) := x''$ by shuffling the k entries of x, yielding x', and letting $x''_i := M^*(x'_i)$ for each $i = 1..k$, sequentially. M is said to be ϵ-LDP on the users' values iff $M^*$is ϵ-LDP.*

All FOs in this paper comply with the above L-LDP definition. In L-LDP, the sanitization parameters ϵ_{∞} and ϵ_1, the original budget, and the α fraction of it, can be defined as the upper bound and the lower bound for ϵ-LDP, respectively. We have the upper bound guarantee when τ, the number of timestamps, tends to infinity, and we have the lower bound when $\tau = 1$. All the L-LDP FOs presented in this paper use the same unbiased estimator:

$$\Phi_{f_L}(v) := \frac{C(v) - nq_1(p_2 - q_2) - nq_2}{n(p_1 - q_1)(p_2 - q_2)} = \frac{\frac{C(v)/n - q_2}{p_2 - q_2} - q_1}{p_1 - q_1} \tag{3}$$

5.1 L-GRR (*Longitudinal Generalized Randomized Response*)

L-GRR is an adaptation of GRR to the longitudinal scenario, adding memoization with 2-round sanitization, using the full and a downsized alpha percentage of the budget for it, which is executed by two instances of the traditional GRR protocol.

The perturbation algorithm is the same as GRR for the first round:

$$\forall_{x \in D} \; Pr\big[\Psi_{L-GRR_{(\epsilon_{\infty})}}(v) = x\big] = \begin{cases} p_1 & \text{if x = v} \\ q_1 & \text{if x} \neq \text{v} \end{cases} \tag{4}$$

where

$$p_1 = \frac{e^{\epsilon}}{e^{\epsilon} + |D| - 1}$$

$$q_1 = \frac{1}{e^{\epsilon} + |D| - 1}$$

followed by a second round that outputs a report x':

$$\forall_{x' \in D} \; Pr\big[\Psi_{L-GRR_{(\epsilon_1)}}(x) = x'\big] = \begin{cases} p_2 & \text{if x' = x} \\ q_2 & \text{if x'} \neq \text{x} \end{cases} \tag{5}$$

where

$$p_2 = \frac{q_1 - e^{\epsilon_1} p_1}{(-p_1 e^{\epsilon_1}) + |D| q_1 e^{\epsilon_1} - q_1 e^{\epsilon_1} - p_1(|D| - 1) + q_1}$$

and

$$q_2 = \frac{1 - p_2}{|D| - 1}$$

5.2 RAPPOR and L-SUE (*Longitudinal Symmetric Unary Encoding*)

RAPPOR was a pioneer of the 2-round sanitization approach. Its utility-oriented implementation is equivalent to the L-SUE protocol. L-SUE follows the same structure as L-GRR but uses SUE for the two rounds and consequently requires the data to be encoded before being processed by it. The perturbation algorithm for RAPPOR and all other UE-based L-LDP FOs for the first round is

$$Pr[\Psi_{UE_{(\epsilon)}} B'[i] = 1] = \begin{cases} p1, if B[i] = 1 \\ q1, if B[i] = 0 \end{cases}$$

for the second round is

$$Pr[\Psi_{UE_{(\epsilon)}} B'[i] = 1] = \begin{cases} p2, if B[i] = 1 \\ q2, if B[i] = 0 \end{cases}$$

In L-SUE, p_1 and q_1 are the same as p and q for standard SUE presented in Section 4.2, and $p_2 + q_2 = 1$. To ensure privacy for all UE algorithms [Arcolezi *et al.*, 2022a], the following equation must be satisfied:

$$\epsilon_1 = ln(\frac{(p_1p_2 - q_2(p_1 - 1))(p_2q_1 - q_2(q_1 - 1) - 1)}{(p_2q_1 - q_2(q_1 - 1))(p_1p_2 - q_2(p_1 - 1) - 1)}) \tag{6}$$

5.3 L-OUE (*Longitudinal Optimized Unary Encoding*)

Similar to L-SUE but built on top of the OUE protocol. OUE is generally regarded as the preferred state-of-the-art solution for the traditional scenario, but L-OUE is prone to adding excessive noise [Arcolezi *et al.*, 2022a], leading to a significant loss of utility over time. The algorithm follows the same structure as RAPPOR, with $p_1 = p_2 = 0.5$, $q_1 = \frac{1}{e^{\epsilon}_{\infty}+1}$, and q_2 may be computed through the Equation (6).

5.4 L-OSUE (*Longitudinal Optimized-Symmetric Unary Encoding*)

As proven by Arcolezi *et al.* [2022a], it is valid to chain both UE protocols and still achieve L-LDP. L-OSUE is a hybrid solution that uses OUE for the first round and SUE for the second, thus avoiding the excessive addition of noise over time, as it happens with data processed under L-OUE. The L-OSUE protocol in first round has $p_1 = 0.5$, $q_1 = \frac{1}{e^{\epsilon}_{\infty}+1}$, followed by SUE with p_2 and q_2 that satisfy $p_2 + q_2 = 1$, satisfying Equation (6).

5.5 LOLOHA

Proposed in Arcolezi *et al.* [2022b], LOLOHA builds on the GRR protocol and applies the technique of Local Hashing to shrink the original domain size *k* to a reduced value of g, up to a minimum of $g = 2$, leading to slower budget consumption. LOLOHA can define g as $g = 2$ (BiLOLOHA) for the strongest longitudinal LDP guarantees or compute an optimal g (OLOLOHA) value by:

$$g = 1 + \max\left(1, \left\lceil \frac{1 - a^2 + \sqrt{A}}{6(a-b)} \right\rceil\right) \tag{7}$$

where

$$A = a^4 - 14a^2 + 12ab(1 - ab) + 12a^3b + 1$$

given $a = \epsilon_\infty$ and $b = \epsilon_1$.

As it builds on the GRR protocol, it first uses a random hash function that maps the user value to a domain of size g, and then follows the same perturbation algorithm, but with $|D| = g$ given our reduced domain size. The sanitization step outputs both the report and the hash function seed, so it can be used for counting by the aggregator. When it comes to the estimation step, it first updates the value of q_1 to $q_1 = 1/g$, and then it counts all values for which the output of the hash function matches the report given the user seed and uses the same unbiased estimator (3) as all other FOs.

6 Post-processing

A Frequency Oracle Protocol has a vector $\tilde{f_v}$ of frequency estimates as its output, with size equal to $|D|$, with D being the domain of user data, and each indexed value $f\tilde{[i]}_v$ being an estimate for the frequency a value $i \in D$ among the samples sent to the aggregator. The frequencies are defined as either simply the counts for a value among the sample data, or the fraction of counts divided by the total number of samples sent to the server. It is expected that the sum of all frequencies is equal to n, where n equals the number of samples when frequencies are defined as counts, or $n = 1$ for frequencies computed as fractions. Due to the randomized nature of LDP protocols, it is common to have negative estimates in the aggregator output, and for the sum of all frequencies to differ from n, which is not representative of real data, and leads to lower utility [Wang *et al.*, 2019b].

To minimize the error in estimations in the post-processing step, it is necessary to guarantee that the output vector has no negative values and ensure it sums up to n. Algorithm 1 details a post-processing procedure to be applied. The algorithm's input is an estimation vector $\tilde{f}$ which represents the frequency for every value in the domain. First, we set all negative estimates to 0 (line 5). Then, we compute the sum of the remaining estimates and determine the average difference by dividing this sum by the number of positive estimates (lines 6 and 7). Next, we add this difference to each positive estimate (line 10). This process is repeated until all values in the output estimation vector $\tilde{f'_v}$ are non-negative and the sum of all frequencies is equal to n.

As demonstrated in Marreiras *et al.* [2024], this approach performs the best for a variety of longitudinal LDP protocols across different datasets. Thus, we choose to apply it to all protocols in the Experimental Analysis (Section 8).

7 Applications

In this section, we explore three distinct algorithms designed for locally differentially private tasks: answering range

Algorithm 1: Algorithm for removing negative estimations

```
Input   : Estimation vector $\tilde{f}_v$
Output : Estimation vector $\tilde{f}'_v$
$\tilde{f}'_v \leftarrow \tilde{f}_v$;
while any negative value in $\tilde{f}'_v$ or $\sum \tilde{f}'_v > n$ do
    for $i \leftarrow 0$ to $|\tilde{f}_v|$ do
        if $\tilde{f}'_v[i] < 0$ then $\tilde{f}'_v$[i] $\leftarrow 0$ ;
    sum $\leftarrow \sum_{j=0}^{j=|\tilde{f}_v|} \tilde{f}'_v(j)$;
    diff $\leftarrow n - sum/|\tilde{f}_v|$;
    for $i \leftarrow 0$ to $|\tilde{f}_v|$ do
        if $\tilde{f}'_v[i] > 0$ then $\tilde{f}'_v$[i] $\leftarrow \tilde{f}'_v$[i] + diff ;
return $\tilde{f}'_v$;
```

queries, mining frequent items, and mining frequent itemsets. It's important to note that these algorithms were initially proposed for single-time data collection scenarios. However, in our work, we extend their application to the longitudinal setting. This extension involves evaluating the accuracy of different longitudinal protocols in the context of each specific task.

7.1 Range Queries

Addressing an unlimited number of multi-dimensional range queries while ensuring data privacy is a critical challenge that has garnered significant attention in recent research Yang *et al.* [2020]; Filho and Machado [2023]; Wang *et al.* [2019a]; Zhang *et al.* [2018]. Range queries play a fundamental role in various data analysis tasks, including statistical analysis, geographic information systems, and machine learning applications. The ability to efficiently process these queries without compromising sensitive information is essential for maintaining data security and usability.

Filho and Machado [2023] proposed FELIP, a grid-based approach designed to answer an unbounded number of multi-dimensional queries with range and point constraints. FELIP maps users' answers to 1-D and 2-D grids, which are perturbed to satisfy LDP. For a dataset of k attributes, the aggregator constructs k 1-D grids and $\binom{k}{2}$ 2-D grids. Since grids may have different sizes, FELIP chooses the LDP protocol with the best expected utility on a per-grid basis. Leveraging the information it has about the queries' selectivity, the aggregator calculates the dimensions' size of each grid with the goal of minimizing the overall variance when answering queries. FELIP does not require the dimensions of each grid to be divisible by the domain, which helps to improve utility. It achieves that by enabling non-uniform cell sizes within a grid. Once the aggregator receives all users' LDP reports, it materializes all grids, removes negative estimations, and makes grids consistent. Finally, it can answer all queries.

7.2 Frequent Item Mining

Identifying the top-k frequent items is crucial for numerous applications across various domains, including market basket analysis, recommendation systems, network security, and healthcare analytics. Efficiently identifying top-k frequent items allows organizations to make data-driven decisions while improving efficiency, security, and user satisfaction Li *et al.* [2012]; Wu *et al.* [2023]; Xiong *et al.* [2018].

Wang *et al.* [2018] *et al.* proposed Set-Value Item Mining (SVIM), a protocol designed to identify, under local differential privacy, frequent items in datasets where each user's data consists of a set of items, all while ensuring local differential privacy (LDP). This protocol operates through a series of steps that balance privacy preservation with accurate frequency estimation. First, from a total of I possible items, it finds a subset S called candidates of frequent items. Users pad and randomly sample items from their sets, then apply LDP noise before sending reports to the aggregator. The aggregator estimates item frequencies and selects candidates. Having narrowed down the domain from I to S, the aggregator now estimates frequencies of items in S. After that, users report the intersection of their items with the set of candidates S. Finally, the aggregator estimates the updated frequency of the frequent items in S and selects the top-k.

7.3 Itemset Mining

Mining frequent itemsets is essential for a wide range of applications across business, security, healthcare, and data science. In market basket analysis, identifying frequently co-purchased items helps businesses optimize inventory, pricing, and personalized recommendations Wang *et al.* [2018, 2017a]. Fraud detection and cybersecurity benefit from frequent itemset mining by recognizing suspicious transaction patterns or detecting anomalies in network traffic. Additionally, in web usage mining and social media analysis, frequent itemsets reveal trending topics, user behavior, and content preferences, improving user engagement and decision-making.

The Set-Valued Set Mining (SVSM) Wang *et al.* [2018] protocol extends the SVIM framework to identify frequent itemsets under Local Differential Privacy (LDP). Given the exponential growth of potential itemsets, SVSM employs a strategic approach to efficiently discover frequent itemsets while preserving user privacy. The protocol first generates a list of itemset candidates. A subset of users participates in the SVIM protocol to identify the top-k frequent items and their estimated frequencies. SVSM estimates the frequencies of possible itemsets by assuming independence among items. Itemsets with the highest estimated frequencies are selected as candidate itemsets and are added to IS. Next, a different subset of users reports the intersection of their data with itemset candidates IS. The aggregator estimates the frequency of each itemset in IS and selects the most frequent ones. By leveraging the frequent items identified through SVIM, the SVSM protocol effectively narrows down the search space for frequent itemsets, enabling efficient and privacy-preserving mining of frequent itemsets in set-valued data under LDP constraints.

8 Experimental Analysis

For our experiments, four distinct datasets were used to analyze the performance of the best-performing protocols as components of different LDP applications. We developed our framework in Python 3.10. All experiments were conducted on a server with Ubuntu 20.04, Intel Core i7-7820X, and 128GB of memory.

8.1 Datasets

The datasets used, their features, and how pre-processing was done for each are described as follows:

- BMS-POS [1]: A dataset of commercial transactions, with a sample of 515595 users, a mostly sparse frequency distribution, a large domain, and diverse itemset sizes, interpolated to achieve 15 timestamps.
- Adult [2]: Composed of demographic data from the 1994 US census. From this, we selected numerical attributes and interpolated values to achieve 260 timestamps.
- Loan [3]: Randomized sample of the lending club dataset with values interpolated to achieve 15 timestamps.
- Bfive[4]: A dataset representing personality test results. We selected an attribute and interpolated its values to achieve 20 timestamps.

Data interpolation was done by having the default dataset values as the first timestamp, and for each subsequent timestamp, the values were shuffled among users. Preserving the domain range and frequency distribution, only changing the value each user held at a new timestamp.

BMS-POS has a mostly skewed frequency distribution towards smaller values, but can generally be considered sparse due to a large domain size. With a large sample size and varying itemset sizes, it can be described as representative of a database best suited for tasks such as frequent item and itemset mining.

Adult and *Loan* both have heavily skewed frequency distributions and present a more realistic scenario for applying LDP [Erlingsson *et al.*, 2014; Wang *et al.*, 2019b]. Both should provide insight into the performance of LDP applications regardless of the intended task.

Bfive has a small domain, a large sample size, and a smooth underlying frequency distribution. Due to its small domain and fixed itemset sizes, it should provide an edge case for applications designed for item and itemset mining.

8.2 Setup for experiments

In our methodology, we varied the privacy budget ϵ_∞ across five distinct values: 0.1, 0.5, 1.0, 1.5, and 2.0. Additionally, we defined a lower bound $\epsilon_1 = 0.1\epsilon_\infty$, using $\alpha = 0.1$. The selected values aim to showcase a progression of varying privacy levels.

For experiments with FELIP, we executed 100 three-dimensional queries, with 20 repetitions for each dataset. The query attributes were all numerical. From the datasets *Bfive*, *Loan*, and *Adult*, 12 attributes were sampled, while 6 attributes were selected from *BMS-POS*. The choice of attributes and dimension size followed the normal distribution.

Experiments with SVIM and SVSM aimed to identify the 64 most frequent items and itemsets in each dataset. However, if the dataset had fewer than 64 items or itemsets, as in the case of *Bfive*, it would need to estimate the frequency of the existing ones best. The experiments for SVIM and SVSM were jointly executed 20 times for each dataset, as SVSM builds on the output of SVIM.

As demonstrated in Marreiras *et al.* [2024], OLOLOHA and L-OSUE have been identified as the most effective and promising Frequency Oracles (FOs) for handling longitudinal data. Therefore, the scope of our experimental analysis in this paper considers these two methods. By focusing on OLOLOHA and L-OSUE, we aim to conduct a more in-depth evaluation of their performance, ensuring a thorough comparison.

8.3 Evaluation Metrics

As proven in Arcolezi *et al.* [2022a], the estimator given by Equation (3) is unbiased, thus the variance of a frequency oracle output $Var[\tilde{f}_v]$ is equal to the Mean Squared Error (MSE) between the original frequencies f_v and the private estimates $\tilde{f}_v$. As such, MSE is a common accuracy metric for measuring the amount of noise an FO adds to user data [Wang *et al.*, 2019b, 2021b], which is equivalent to data utility loss.

In our analysis, we utilize MSE_{avg}, detailed in Section 3 and given by Equation (1), an alternative to MSE presented in Arcolezi *et al.* [2022a] suited for adequately measuring data utility loss in longitudinal data. An experiment is conducted by computing MSE_{avg} for each unique set of dataset, application, and frequency oracle. All experiments are executed 20 times with different random seeds, with MSE_{avg} being averaged across all runs. This procedure is necessary to account for the random nature of the noise introduced by LDP protocols.

8.4 Results

Results are shown in Figs. 1, 2, and 3 and are discussed for each LDP application. We define a scenario as a dataset–application pair together with the experimental variables that emulate a realistic deployment of that application. For each scenario, we compare the resulting MSE_{avg} of L-OSUE and OLOLOHA (as application components) across increasing privacy budgets.

We expect MSE_{avg} to decrease as the privacy budget increases since lower noise corresponds to greater data utility and reduced privacy, divergent results are an indication of unexpected behavior.

One potential cause is a mismatch between the frequency oracle's design assumptions and the scenario (e.g., underlying frequency distribution, privacy budget, or LDP application), which can lead to over- or under-perturbation. Another is bias introduced by post-processing. To mitigate the latter, we adopt the method discussed in Sec. 6, which has been shown to introduce minimal bias.

When the output presents unexpected behavior, the frequency oracle will be deemed inadequate for use with the LDP application, as even if the results showcase high utility

[1] https://www.kdd.org/kdd-cup/view/kdd-cup-2000

[2] http://archive.ics.uci.edu/ml/datasets/Adult

[3] https://www.kaggle.com/datasets/wordsforthewise/lending-club

[4] https://www.kaggle.com/datasets/tunguz/big-five-personality-test

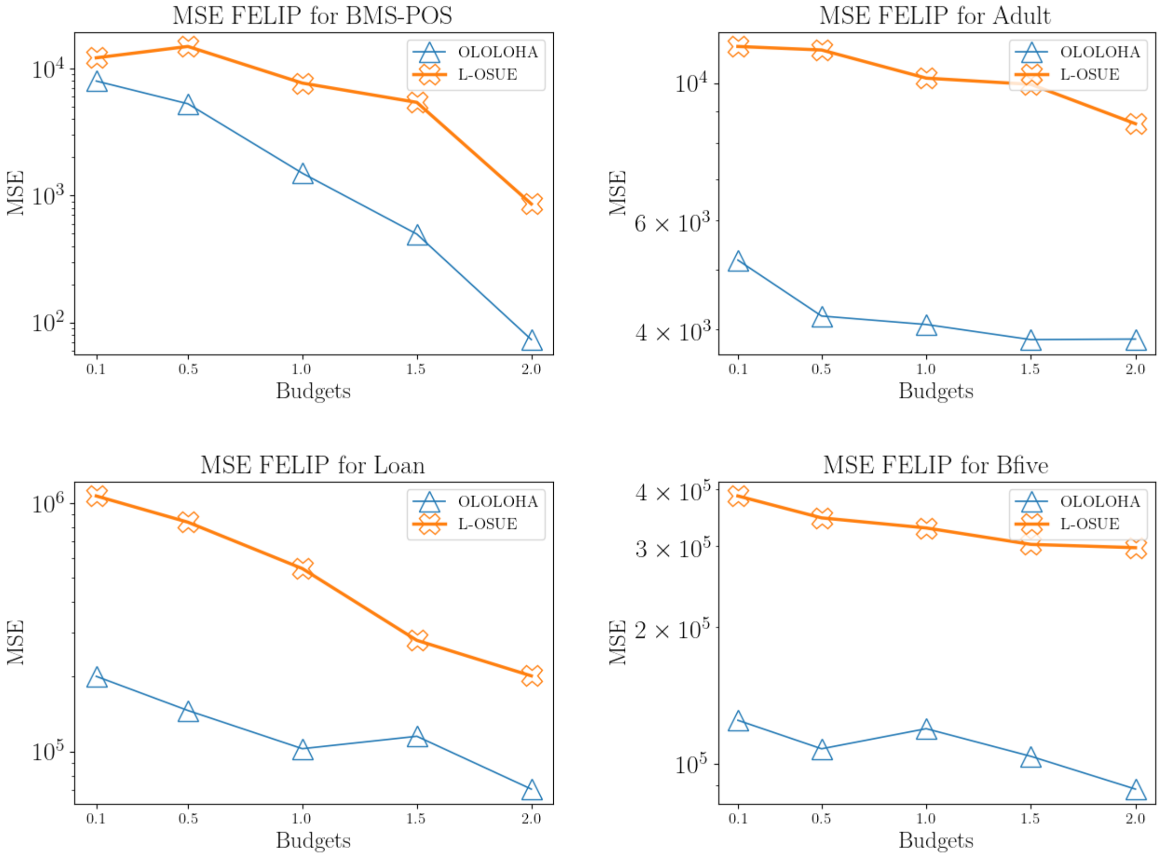

Figure 1. FELIP MSE results, lower is better

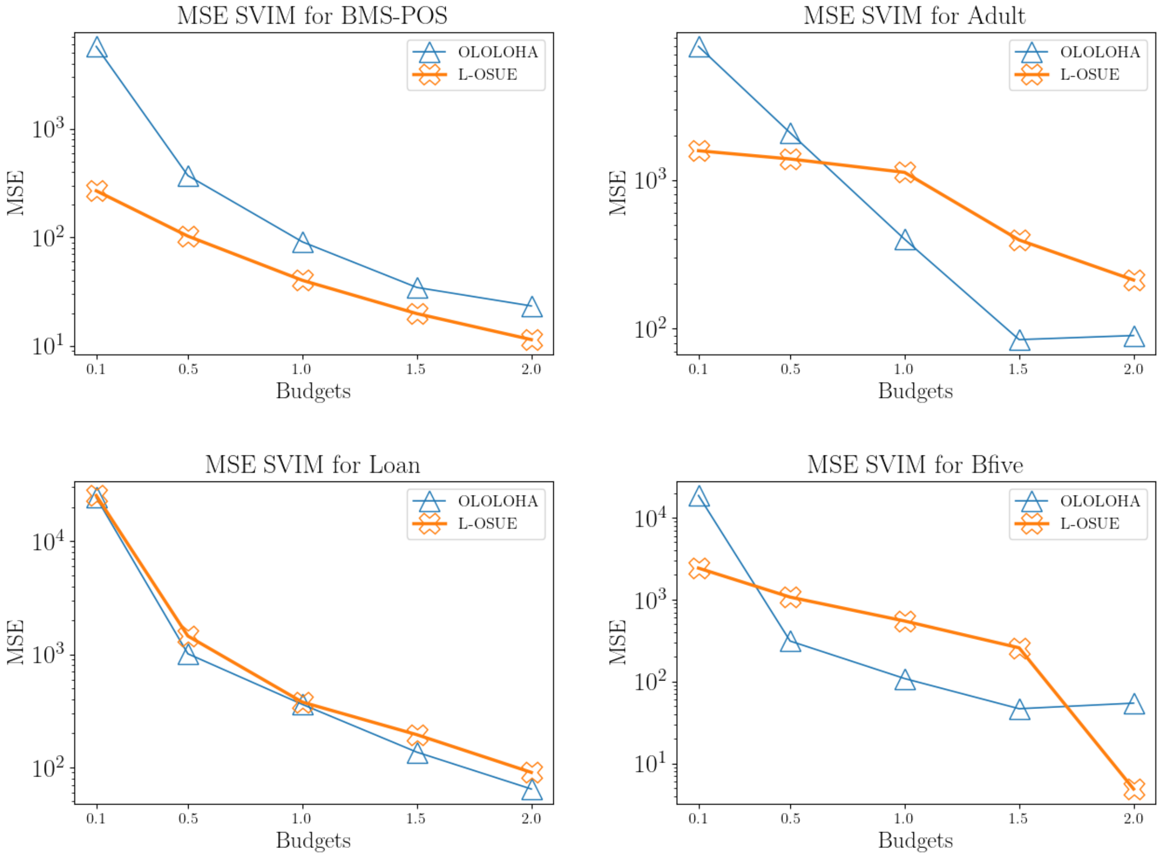

Figure 2. SVIM MSE results, lower is better

and privacy guarantees, there will be no adequate way of controlling the amount of noise by choosing a privacy budget, as well as the output utility can be subject to change from subsequent user data being sent to the aggregator, potentially altering the underlying data distribution [Marreiras *et al.*, 2024], resulting in unpredictable noise levels and output utility for the application.

For most cases, the frequency oracle exhibiting the lowest overall MSE_{avg} for the selected datasets and privacy budgets, particularly smaller values of ϵ, will be highlighted as the most suitable for a given application, as it will be understood to provide the best trade-off between privacy and utility. However, if an experiment presents a given scenario, which is understood to be representative of a more realistic deployment of an LDP application (such as SVIM or SVSM processing data from *BMS-POS*), our evaluation will lean more heavily on results showcased by it, with results from other experiments serving as criteria for a tiebreaker, if required.

8.5 FELIP: Answering range queries

Across all datasets and privacy budgets tested for FELIP, OLOLOHA exhibited the best performance, as observed in Figure 1. The narrowest utility gap between the two frequency oracles occurred for smaller privacy budgets with the *BMS-POS* dataset. However, with the same dataset, the gap increased exponentially for larger values of ϵ. This behavior was not observed in the other datasets and is likely due to OLOLOHA's greater sensitivity to large domain sizes and budget variations, compared to a protocol based on unary encoding, such as L-OSUE [Arcolezi *et al.*, 2022b].

Given that the standalone usage of the frequency oracles has shown varied results when comparing OLOLOHA and L-OSUE performance in testing with multiple distinct datasets [Marreiras *et al.*, 2024]. It is valid to infer from our findings that FELIP heavily favors OLOLOHA over L-OSUE, as it showcased better utility for all datasets. Therefore, OLOLOHA can be considered the most adequate FO to be used in FELIP for longitudinal data processing, as even when accounting for different features and frequency distributions, L-OSUE presented no advantage.

8.6 SVIM: Frequent item mining

In experiments with SVIM, with results shown in Figure 2, L-OSUE mostly outperformed OLOLOHA when the privacy budget was smaller. L-OSUE presented the best utility over OLOLOHA for the *BMS-POS* dataset. Meanwhile, for *Adult* and *Bfive*, OLOLOHA showcased higher utility for smaller privacy budgets, and both frequency oracles matched results in experiments with *Loan*.

As per the experimental parameters defined in 8.1, the results found for the *Bfive* and *Adult* datasets are only representative when SVIM is tasked with identifying the frequencies of all values in user data, and not just the top-k most frequent. However, when SVIM is set to identify the top-k items, given a certain $k < |D|$ with D being the domain of user data, such as the 64 most frequent items and their frequencies for the datasets *Loan* and *BMS-POS* (a more realistic and adequate deployment of the application), the frequency oracles showcase similar results for former, which has a smaller domain size, while L-OSUE performed better when processing the latter. Thus, we can infer that given an adequate deployment scenario, both protocols are expected to present similar results for data with small domain sizes, while L-OSUE performs at its best for larger ones.

8.7 SVSM: Frequent itemset mining

Experiments with SVSM produced the least useful results. The memoization of dummy values already caused utility loss for SVIM; however, because SVSM not only relies on the results of SVIM but also generates exponentially more possible dummy itemsets from a few items, this utility loss was significantly amplified. For *BMS-POS*, no frequency oracle pairing could identify the most frequent itemsets. The large domain of this dataset made the described utility loss effect unavoidable.

Results for other datasets are shown in Figure 3. *Bfive* only showcases results for L-OSUE, as OLOLOHA was not able to identify any of the most frequent itemsets. As the local hashing technique used for shrinking the domain also makes the memoization of dummy itemsets more likely, with a stronger impact to be expected when the original domain size is already small. This is also supported by OLOLOHA having consistently worse utility than L-OSUE, even when it does identify some frequent itemsets, as seen in the results for *Loan* and *Adult*.

Overall, while L-OSUE can be outlined as a better fit for SVSM than OLOLOHA, neither can be considered adequate. Both failed to deliver useful and private output in experiments with the *BMS-POS* dataset, which can be described as a desired deployment scenario for SVSM: finding the most frequent itemsets in a large commerce transactions platform. As such, adjustments to either the SVSM application or the LDP protocols tested will be necessary for acceptable results to be achieved.

8.8 Summary

In summary, OLOLOHA was found to be the best-performing FO as a component of FELIP, with no scenarios found where L-OSUE had a utility advantage. L-OSUE and OLOLOHA presented competitive results and components of SVIM; however, L-OSUE showcased the best result for the *BMS-POS* dataset and smaller privacy budgets, thus being the most promising choice for SVIM.

Neither frequency oracle presented acceptable results in experiments with SVSM, in failing to adequately identify frequent itemsets for *BMS-POS*, and in the case of OLOLOHA, *Bfive* as well. Therefore, no alternative for adapting SVSM to longitudinal data processing has been determined.

9 Conclusion

In this paper, we conducted an evaluation of two frequency oracles adapted for longitudinal data as components of task-specific LDP applications. By building on previous work, we selected OLOLOHA and L-OSUE as the most promising

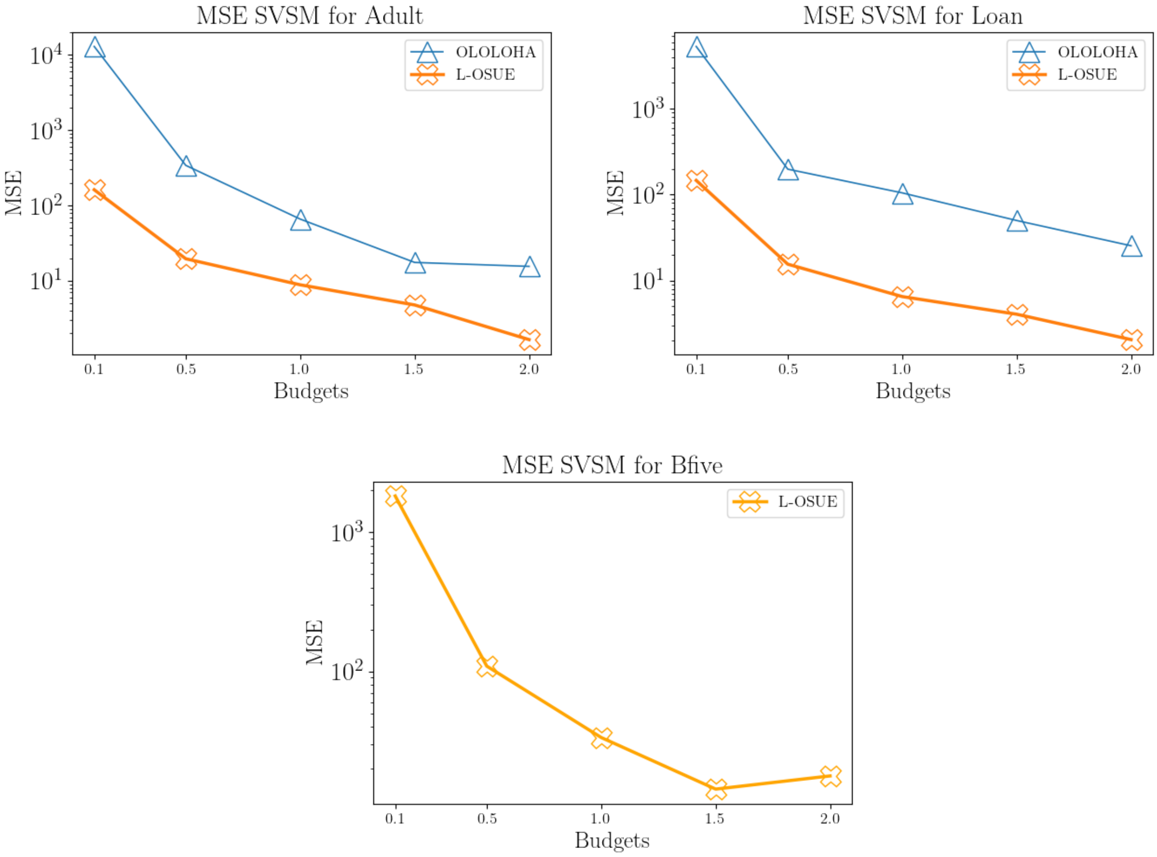

Figure 3. SVSM MSE results, lower is better

frequency oracles with Longitudinal LDP guarantees. FELIP was selected as an LDP application for Answering Range Queries, while SVIM and SVSM were selected for the tasks of frequent item and itemset mining, respectively.

In experiments with FELIP, OLOLOHA consistently delivered greater utility than L-OSUE. For SVIM, L-OSUE presented the best trade-off between privacy and utility for smaller budgets and larger domain sizes. The results for SVSM were not promising for either frequency oracle, as both failed to identify any frequent itemset when the domain was large enough.

For future work, we aim to research security vulnerabilities and exploits in LDP applications and protocols, such as data poisoning attacks [Cao *et al.*, 2021], and how these can be enhanced, or more effectively mitigated when processing longitudinal data.

Acknowledgements

This work was funded by Lenovo Brasil as part of its R&D investment under Brazil's Informatics Law.

References

Arcolezi, H. H., Couchot, J.-F., Al Bouna, B., and Xiao, X. (2022a). Improving the utility of locally differentially private protocols for longitudinal and multidimensional frequency estimates. *Digital Communications and Networks*.

Arcolezi, H. H., Pinzón, C., Palamidessi, C., and Gambs, S. (2022b). Frequency estimation of evolving data under local differential privacy. *arXiv preprint arXiv:2210.00262*.

Cao, X., Jia, J., and Gong, N. Z. (2021). Data poisoning attacks to local differential privacy protocols. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 947–964.

Ding, B., Kulkarni, J., and Yekhanin, S. (2017). Collecting telemetry data privately. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 3574–3583, Red Hook, NY, USA. Curran Associates Inc.

Duarte Neto, E. R., Costa Filho, J. S., Neto, A. A. M., and Machado, J. C. (2026). Alog: Adaptive longitudinal grids for geospatial data using local differential privacy. In *EDBT*, pages 1–14.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer.

Dwork, C., Roth, A., *et al.* (2014). The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407.

Erlingsson, Ú., Pihur, V., and Korolova, A. (2014). Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067.

Filho, J. S. C. and Machado, J. C. (2023). Felip: A local differentially private approach to frequency estimation on multidimensional datasets. In *International Conference on Extending Database Technology*.

Hong, D., Jung, W., and Shim, K. (2021). Collecting geospatial data with local differential privacy for personalized services. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 2237–2242.

Johnson, N., Near, J. P., and Song, D. (2018). Towards practical differential privacy for sql queries. 11(5).

Kairouz, P., Bonawitz, K., and Ramage, D. (2016). Discrete distribution estimation under local privacy. In *International Conference on Machine Learning*, pages 2436–2444. PMLR.

Li, J., Gan, W., Gui, Y., Wu, Y., and Yu, P. S. (2022). Frequent itemset mining with local differential privacy. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 1146–1155, New York, NY, USA. Association for Computing Machinery.

Li, N., Qardaji, W., Su, D., and Cao, J. (2012). Privbasis: frequent itemset mining with differential privacy. *Proc. VLDB Endow.*, 5(11):1340–1351. DOI: 10.14778/2350229.2350251.

Liu, G., Tang, P., Hu, C., Jin, C., and Guo, S. (2023). Multidimensional data publishing with local differential privacy. In *Proceedings 26th International Conference on Extending Database Technology, EDBT 2023, Ioannina, Greece, March 28-31, 2023*, pages 183–194. OpenProceedings.org.

Marreiras, A. A., Neto, E. R. D., Costa Filho, J. S., and Machado, J. C. (2024). Locally differentially private and consistent frequency estimation of longitudinal data. In *Simpósio Brasileiro de Banco de Dados (SBBD)*, pages 367–380. SBC.

Ren, X., Shi, L., Yu, W., Yang, S., Zhao, C., and Xu, Z. (2022). Ldp-ids: Local differential privacy for infinite data streams. In *Proceedings of the 2022 International Conference on Management of Data*, SIGMOD '22, page 1064–1077, New York, NY, USA. Association for Computing Machinery.

Team, A. D. P. (2017). Learning with privacy at scale.

Wang, N., Xiao, X., Yang, Y., Zhang, Z., Gu, Y., and Yu, G. (2017a). Privsuper: A superset-first approach to frequent itemset mining under differential privacy. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pages 809–820. DOI: 10.1109/ICDE.2017.131.

Wang, T., Blocki, J., Li, N., and Jha, S. (2017b). Locally differentially private protocols for frequency estimation. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 729–745.

Wang, T., Chen, J. Q., Zhang, Z., Su, D., Cheng, Y., Li, Z., Li, N., and Jha, S. (2021a). Continuous release of data streams under both centralized and local differential privacy. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, New York, NY, USA. Association for Computing Machinery.

Wang, T., Ding, B., Zhou, J., Hong, C., Huang, Z., Li, N., and Jha, S. (2019a). Answering multi-dimensional analytical queries under local differential privacy. SIGMOD '19, page 159–176, New York. ACM.

Wang, T., Li, N., and Jha, S. (2018). Locally differentially private frequent itemset mining. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 127–143. DOI: 10.1109/SP.2018.00035.

Wang, T., Lopuhaä-Zwakenberg, M., Li, Z., Skoric, B., and Li, N. (2019b). Locally differentially private frequency estimation with consistency. *arXiv preprint arXiv:1905.08320*.

Wang, T., Zhao, J., Hu, Z., Yang, X., Ren, X., and Lam, K.-Y. (2021b). Local differential privacy for data collection and analysis. *Neurocomputing*, 426:114–133.

Warner, S. L. (1965). Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69.

Wu, H., Ran, R., Peng, S., Yang, M., and Guo, T. (2023). Mining frequent items from high-dimensional set-valued data under local differential privacy protection. *Expert Systems with Applications*, 234:121105. DOI: https://doi.org/10.1016/j.eswa.2023.121105.

Xiong, X., Chen, F., Huang, P., Tian, M., Hu, X., Chen, B., and Qin, J. (2018). Frequent itemsets mining with differential privacy over large-scale data. *IEEE Access*, 6:28877–28889. DOI: 10.1109/ACCESS.2018.2839752.

Yang, J., Wang, T., Li, N., Cheng, X., and Su, S. (2020). Answering multi-dimensional range queries under local differential privacy. *Proc. VLDB Endow.*, 14(3).

Zhang, Z., Wang, T., Li, N., He, S., and Chen, J. (2018). Calm: Consistent adaptive local marginal for marginal release under local differential privacy. CCS '18, page 212–229, New York, NY, USA. Association for Computing Machinery.

Zhu, Y., Cao, Y., Xue, Q., Wu, Q., and Zhang, Y. (2024). Heavy hitter identification over large-domain set-valued data with local differential privacy. *IEEE Transactions on Information Forensics and Security*, 19:414–426. DOI: 10.1109/TIFS.2023.3324726.