Computer Standards & Interfaces 89 (2024) 103827

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

Local differential privacy and its applications: A comprehensive survey☆

Mengmeng Yang [a], Taolin Guo [b], Tianqing Zhu [c,*], Ivan Tjuawinata [d], Jun Zhao [d], Kwok-Yan Lam [d]

[a] Data61 of Commonwealth Scientific and Industrial Research Organization, Melbourne, 3168, VIC, Australia
[b] College of Computer and Information Science, Chongqing Normal University, 401331, Chongqing, China
[c] Centre for Cyber Security and Privacy and the School of Computer Science, University of Technology Sydney, Sydney, 2007, NSW, Australia
[d] Strategic Centre for Research in Privacy-Preserving Technologies & Systems, Nanyang Technological University, 639798, Singapore

ARTICLE INFO

Keywords:
Private data statistics
Local differential privacy
Private data analysis

ABSTRACT

With the rapid development of low-cost consumer electronics and pervasive adoption of next generation wireless communication technologies, a tremendous amount of data has been generated from users' smart devices and collected for research and analysis. This inevitably results in increasing concern of mobile users regarding their personal information; the problem of privacy preservation has become more urgent and it has also attracted a significant amount of attention from both academic researchers and industry practitioners. As a strong privacy tool, local differential privacy (LDP) has been widely deployed in recent years. It eliminates the need for a trusted third party by allowing users to perturb their data locally, thus providing better privacy protection. This survey provides a comprehensive and structured overview of LDP technology. We summarize and analyse state-of-the-art development in LDP and compare a range of methods from various perspectives and from the context of machine learning model training. We explore the applications of LDP in various domains. Furthermore, we identify several research challenges and discuss promising future research directions.

1. Introduction

Over the last few years, a large volume of data has been generated and collected for various data analyses towards decision-making or service improvement. This data can be acquired from end-user devices or even wearable devices, which include users' private data that can even be highly sensitive. With the emergence of new technology capable of in-depth mining and analysing users' data, users have raised concerns over their data privacy [1]. Various authorities have also enacted privacy laws to regulate organizations in handling and using their users' data, such as General Data Protection Regulation (GDPR) [2], California Consumer Privacy Act (CCPA) [3], and Personal Data Protection Act (PDPA) [4]. This has increased the urgency and importance of privacy preservation which needs to be addressed.

Differential privacy [5], as a strict privacy definition, has grown to be one of the *de facto* standards for preserving privacy and has been applied in a variety of areas. Traditional differential privacy, also named centralized differential privacy (CDP), is typically done by having a service provider collect the user's original data first and then release the noisy statistical information to the public. The service provider is assumed to be trusted in the centralized model. However, the reality is not always the case. Even big reputable companies may not be able to guarantee their customers' privacy [6]. For instance, it has been reported that in 2018, hundreds of thousands of Google+ social network users had their private data leaked by Google. Later in the same year, it was reported that 52.5 million users had their accounts exposed due to a bug in the Google+ API [7]. In 2019, hundreds of millions of Facebook users' IDs, phone numbers, and names were exposed online [8]. This suggests that the assumption of having a trusted third party to manage the users' data may be very difficult to establish.

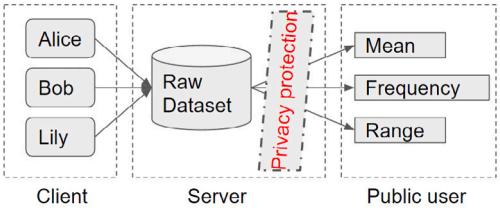
Different from centralized differential privacy, local differential privacy (LDP) allows users to perturb their data locally on their own devices. Only the perturbed data are reported to the server. Fig. 1 shows the comparison of the framework of CDP and LDP. For CDP, the server possesses the users' original data. In contrast, under the LDP model, the server holds a perturbed version of the data and queries are handled based on this perturbed dataset. This allows for the service provider to be untrusted without compromising the privacy of the data while also relieving it from the burden of preserving the data privacy.

Despite its potential that has been deployed by several large companies [9–11] to preserve their users' privacy, LDP comes with its own drawbacks. More specifically, in general, the noise introduced to the

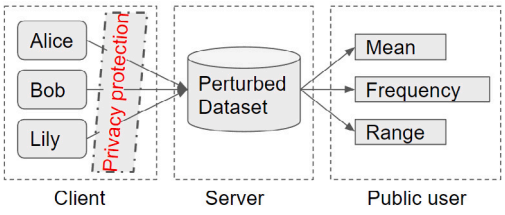
☆ This research work is supported by National Natural Science Foundation of China (No. U22A2026) and the Natural Science Foundation of Chongqing, China (No. CSTB2022NSCQ-MSX1383).

* Corresponding author.
E-mail address: tianqing.zhu@uts.edu.au (T. Zhu).

https://doi.org/10.1016/j.csi.2023.103827
Received 29 October 2023; Received in revised form 8 December 2023; Accepted 25 December 2023
Available online 30 December 2023



(a) Centralized differential privacy



(b) Local differential privacy

Fig. 1. Comparison between CDP and LDP Models.

whole dataset can be very large, which results in query responses with a much lower utility compared to that of the centralized model. Furthermore, since perturbation is done by each user without any information regarding other users' data, such perturbation must be independent and hence limit the scope of its application. These limitations cause LDP to be much less studied compared to CDP. Our motivation to write this survey is to provide a summary of studies in LDP which we also believe to be beneficial for the study and future development of the field. We note that despite the existence of a number of surveys in LDP, they tend to mainly focus on a smaller, more specific part of the field. In 2018, Cormode et al. [12] provided a brief tutorial on LDP. The work by Zhao et al. in 2019 [13] mainly surveyed the potential applications of LDP in securing the internet of connected vehicles. In the same year, Bebensee [14] provided a survey that focused on heavy hitter identification and spatial data collection. Compared to [15]'s work, we provided a simpler and clearer classification, and deeper analysis and discussion. Furthermore, the survey by Cormode et al. [16] focused on a comparative study of frequency estimation under LDP.

In this survey, we identify two research directions according to the purpose of perturbed data collection: statistical query with LDP and private learning with LDP. For statistical query with LDP, the aggregator aims to collect users' data to answer a specific query, such as frequency, mean, and range of the dataset. Due to its specificity, the data perturbation mechanism tends to be specially designed depending on the query type it is trying to solve. On the other hand, for private learning with LDP, the aggregator aims to collaborate with users to train a model, which is associated with a particular machine learning algorithm. The data perturbation method is hence dependent on the specific algorithm. In this survey, we identify the problems in such directions along with challenges unique to the two research directions. We then discuss and review existing solutions to the problems while providing some comparisons regarding their advantages and disadvantages. Next, we explore the latest research progress on the application of LDP including federated learning, reinforcement learning, location privacy, and recommendation systems. Lastly, based on such advances, we identify some challenges that need to be considered as well as some further research directions.

This survey is structured as follows. Some preliminaries are introduced in Section 2. The review of existing LDP methods for statistic query and private learning problems is presented in Section 3 and Section 4 respectively. Section 5 explores the applications of LDP. Section 6 provides an extensive discussion on research challenges and research directions. We conclude this survey in Section 7.

2. Preliminaries

2.1. Definition

Definition 1 ((ϵ, δ)-*Local Differential Privacy [17]*)**.** Let $\epsilon > 0$ and $\mathcal{X}$ be the domain of the user's data. A randomized algorithm $\mathcal{M} : \mathcal{X} \to \mathcal{M}(\mathcal{X})$, which is applied to each user's record independently, satisfies (ϵ, δ)-local differential privacy, if and only if for any pair of input values $x, x' \in \mathcal{X}$ and for any possible output $S \subseteq Range(\mathcal{M})$, we have

$$Pr[\mathcal{M}(x) \in S] \leq e^{\epsilon} Pr[\mathcal{M}(x') \in S] + \delta. \tag{1}$$

The special case when $\delta = 0$, is ϵ-Local Differential Privacy.

This definition guarantees that for any perturbed data $\hat{v} = \mathcal{M}(x)$ that the aggregator receives, regardless of any additional knowledge that it may learn, the information about the value of x that can be learned from the output $\hat{x}$ is bounded by a function of ϵ, which limits the confidence that the aggregator may have regarding the value of x.

Differential privacy sets itself apart by adding controlled noise to data queries, ensuring individual privacy while allowing accurate analysis of aggregated information. Unlike traditional methods that focus on anonymization or encryption [18,19], differential privacy safeguards against re-identification attacks and preserves privacy in aggregate data more effectively.

2.2. Fundamental mechanisms

This section summarizes a few of the basic local differential privacy mechanisms, which are building blocks for complex algorithm design.

Laplace Mechanism. Let $\epsilon > 0$ and $f : \mathcal{X} \to \mathbb{R}^d$ be a function related to a query. The Laplace mechanism $\mathcal{M}_L : \mathcal{X} \to \mathbb{R}^d$ with parameter ϵ is defined as follows. For any $x \in \mathcal{X}$,:

$$\mathcal{M}_L(x) = f(x) + (N_1, \ldots, N_d), \tag{2}$$

where $N_1, \ldots, N_d$ are independently and identically distributed from the Laplace distribution $N_j \sim Lap(0, \Delta f/\epsilon)$. Δf refers to the ℓ_1 sensitivity.

Gaussian Mechanism. Let $\epsilon, \delta > 0$ and $f : \mathcal{X} \to \mathbb{R}^d$ be a function related to a query. The Gaussian mechanism $\mathcal{M}_G : \mathcal{X} \to \mathbb{R}^d$ with parameters ϵ, δ is defined as follows. For any $x \in \mathcal{X}$,:

$$\mathcal{M}_G(x) = f(x) + (N_1, \ldots, N_d), \tag{3}$$

where $N_1, \ldots, N_d$ are independently and identically distributed from the Gaussian distribution $N_j \sim \mathcal{N}(0, \sigma^2)$ where σ is defined as a positive real number such that $\sigma \geq c\Delta_2 f/\epsilon$ for some positive constant c satisfying $c^2 \geq \ln(1.25/\delta)$. $\Delta_2 f$ refers to the ℓ_2 sensitivity.

The concept of sensitivity is used to capture the largest change that the sensitive report can have. This determines the magnitude of the noise to be added, which further affects the accuracy of the final query response [20]. We provide the definition of ℓ_1 and ℓ_2 sensitivity of the function f as follows.

Definition 2 (ℓ_1, ℓ_2-*Sensitivity*)**.** Let a query be given and let $f : \mathcal{X} \to \mathbb{R}^d$ be a function that determines the report that the aggregator requires from each user to respond to the query for some response dimension $d \in \mathbb{Z}_{>0}$. Then the ℓ_1-sensitivity of f is defined as follows:

$$\Delta f \triangleq \max_{x,y \in \mathcal{X}, x \neq y} \|f(x) - f(y)\|_1. \tag{4}$$

Similarly, the ℓ_2 sensitivity of f is defined as

$$\Delta_2 f \triangleq \max_{x,y \in \mathcal{X}, x \neq y} \|f(x) - f(y)\|_2. \tag{5}$$

$\mathcal{M}_L$ and $\mathcal{M}_G$ achieve ϵ-LDP and (ϵ, δ)-LDP respectively. One way to improve the accuracy guarantee of a query response while maintaining the privacy guarantee is by limiting the sensitivity of the corresponding

function. Technique such as clipping technique [21] or truncation is usually adopted to achieve such goal.

In contrast to Laplace and Gaussian mechanisms which were initially designed for CDP, the majority of LDP mechanisms are designed based on the idea of randomized response [22].

Randomized Response (RR). Let $p \in [0,1], \mathcal{X}$ be the domain of the user's data with size $2, t \in \mathcal{X}$ be a user's private value and $\hat{t} \in \mathcal{X}$ be its perturbed response outputted by the randomized response mechanism with probability p. Then $\hat{t}$ is a random variable such that for $v \in \mathcal{X}$,

$$Pr[\hat{t} = v] = \begin{cases} p, & \text{if } t = v \\ 1-p, & \text{if } t \neq v. \end{cases} \tag{6}$$

We denote such RR mechanism with parameter p by $RR(p)$. Holohan et al. [23] showed that the above mechanism provides ϵ-LDP if $p \in \left[\frac{1}{1+e^{\epsilon}}, \frac{e^{\epsilon}}{1+e^{\epsilon}}\right]$ and among all such choices of p, choosing $p = \frac{e^{\epsilon}}{e^{\epsilon}+1}$ yields a mechanism with the minimum expected error. Note that the RR mechanism is only defined when $|\mathcal{X}| = 2$. It is then natural to extend it to a larger domain, which is defined by *generalized randomized response (GRR),* which was proposed in [24].

Generalized Randomized Response (GRR). Suppose that $|\mathcal{X}| = d$ for some positive integer $d \geq 2$ and $p \in [0,1]$. Suppose that a user u holds a value $t \in \mathcal{X}$ and let $\hat{t} \in \mathcal{X}$ be its perturbed response outputted by the generalized randomized response mechanism. Then $\hat{t}$ is defined to be a random variable such that for any $v \in \mathcal{X}$,

$$Pr[\hat{t} = v] = \begin{cases} p, & if\ t = v \\ \frac{1-p}{d-1}, & if\ t \neq v. \end{cases} \tag{7}$$

We denote such GRR mechanism with parameter p by $GRR(p)$. Similar to the case of RR, the above mechanism provides ϵ-LDP if $p \in \left[\frac{1}{1+(d-1)e^{\epsilon}}, \frac{e^{\epsilon}}{e^{\epsilon}+d-1}\right]$ and among all such choices of p, choosing $p = \frac{e^{\epsilon}}{e^{\epsilon}+d-1}$ yields a mechanism with the minimum expected error. It is also easy to see that when $d = 2$, the generalized randomized response mechanism reduces back to the RR mechanism defined above.

2.3. Composition properties

The composition of differential privacy mechanisms can be used to help us in designing differentially private mechanisms for more sophisticated queries. The composition properties work for both centralized and local differential privacy [25].

Sequential Composition. Given a random algorithm $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_m\}$, which consists of m sequential steps. Assume each mechanism $\mathcal{M}_i$ satisfies ϵ_i-local-differential privacy and all mechanisms are performed on the same dataset, then, $\mathcal{M}$ satisfies $(\sum_{i=1}^{m} \epsilon_i)$- local differential privacy.

Parallel Composition. Assume there are a set of privacy mechanisms $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_m\}$, which are performed on disjointed data records of the dataset. If each of the mechanisms satisfies ϵ_i-local differential privacy, the $\mathcal{M}$ provides $\max\{\epsilon_1, \ldots, \epsilon_m\}$-local differential privacy guarantee.

Post-Processing. Given a randomized algorithm $\mathcal{M}$ that provides ϵ-local differential privacy guarantee. Let f be an arbitrary randomized mapping, post-processing an output of the differential privacy algorithm $f \circ M$ does not incur any additional loss of privacy.

3. Statistical query with LDP

In addition to helping service providers to better understand customer needs which is essential in providing better and more effective services, as can be observed from the composition theorems discussed above, privacy-preserving mechanisms specifically designed to address simple queries are also important building blocks for mechanisms designed for more complicated data analyses. Fig. 2 summarizes the different query types studied in the literature.

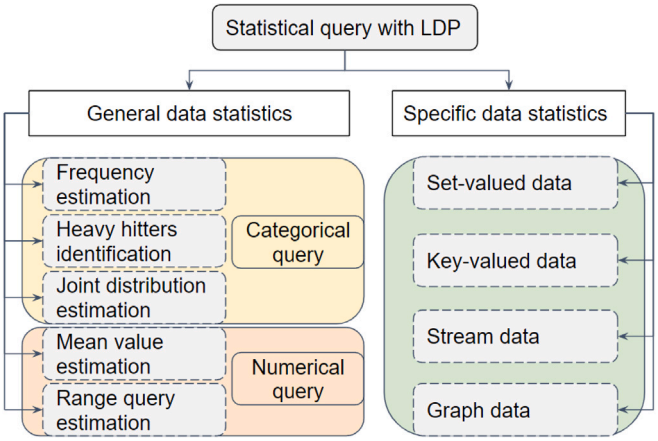


Fig. 2. Classification of Statistical Query.

3.1. General data statistics

This section discusses the statistics over traditional categorical and numerical data.

3.1.1. Frequency estimation

Let $U = \{u_1, \ldots, u_n\}$ be a set of users and $A = \{I_1, \ldots, I_d\}$ be a set of possible items that each user may hold. The aim of the data aggregator is to find the frequency of each item in A. More specifically, it aims to estimate $(f_1, \ldots, f_d)$ where for $i = 1, \ldots, d, f_i = \frac{c(I_i)}{n}$ with $c(I_i) = \left|\{j \in \{1, \ldots, n\} : u_j \text{ holds } I_i\}\right|$. In general, frequency estimation methods can be divided into 4 processes.

- *Encoding.* Suppose that for $i = 1, \ldots, n, u_i$ holds $v_i \in A$. Each user u_i encodes v_i using a predefined coding scheme to t_i, which can either be a value or a binary vector to be used as an input to a perturbation mechanism.
- *Perturbation.* Having t_i, each user u_i uses a perturbation mechanism $\mathcal{M}$ satisfying local differential privacy to perturb t_i to $\hat{t}_i$ under the guarantee that the distribution of $\hat{t}_i$ is statistically indistinguishable to the case when u_i holds any other value.
- *Aggregation.* The data aggregator receives all the perturbed values $\{\hat{t}_1, \ldots, \hat{t}_n\}$ from n users and aggregates them accordingly.
- *Estimation.* The aggregator performs post-processing to calculate an estimation $(\hat{f}_1, \ldots, \hat{f}_d)$ of $(f_1, \ldots, f_d)$ according to the perturbation strategy to ensure that the estimate is unbiased. Such post-processing techniques can also be done to improve estimation accuracy.

In the following, we discuss several groups of typical frequency estimation methods.

Direct perturbation. We refer direct perturbation to schemes with no to minimal encoding before the perturbation process. Typical frequency estimation methods can be constructed without any encoding where the perturbation is done using either RR or GRR. Another very commonly used direct perturbation frequency estimation is named Optimized Unary Encoding (OUE), which was proposed by Wang et al. [26] which we will discuss in the following. Suppose that $|A| = d$ and for a user u_i that holds $v_i = I_{j_i} \in A$, the encoding of v_i is done by defining a binary vector $t_i = (t_{i,1}, \ldots, t_{i,d})$ of length d, where only $t_{i,j} = 1$, all other values equal to 0. The perturbation is then done for each bit independently. More specifically, given $t_{i,j}$, the jth entry of the output $\hat{t}_{i,j}$ is generated using $RR(p)$ with possible outputs $\{0,1\}$ where

$$p = \begin{cases} \frac{1}{2}, & \text{if } t_{i,j} = 1 \\ \frac{e^{\epsilon}}{e^{\epsilon}+1}, & \text{if } t_{i,j} = 0. \end{cases} \tag{8}$$

Table 1
Comparison of frequency estimation methods.

Method	Typical papers	Description	Advantages	Disadvantages
Direct perturbation	[24,26]	Randomized response is directly done to private value or minimally encoded value	Easy to perform and minimal additional computational cost	Poor performance when the dimension is high
Hash	[9,26]	Private value from a large dataset is hashed to a hash digest from a much smaller space, in which randomized response is applied	Smaller communication cost	Decoding process is complex and collision problem needs to be considered
Transformation	[17,29,31]	Transform the user value into a single bit, perform the randomized response to this bit	Small communication cost	Additional information loss during the transformation process
Subset selection	[32,33]	Random sampling of k values to report	Good performance in the intermediate privacy region	High communication cost

Discussion. The three mechanisms mentioned above (RR, GRR, and OUE) are the most basic building blocks. In general, solutions that are based on RR and GRR are more suitable in scenarios where the dataset has a low dimension. The accuracy may deteriorate when the dimension of the data is high. Although OUE has a better performance in cases with larger datasets compared to RR and GRR, this comes at the cost of a much higher communication cost.

Hash. Hash functions are typically characterized by the mapping of inputs in a large domain to hash digests in a much smaller space. Let $\mathbb{H}$ be a family of universal hash functions and $\mathcal{H} \in \mathbb{H}$. For each user u_i with value v_i, it first encodes v_i to $\langle \mathcal{H}, \mathcal{H}(v) \rangle$. In RAPPOR [9], $\mathcal{H}(v)$ is then further encoded to a k-bit vector where the perturbation is then done by performing RR on each bit of this final encoded vector. Having these perturbed vectors from each user, the aggregator utilizes Lasso regression [27] to obtain an accurate estimation of the frequency. Unfortunately, this technique was later proven to be substantially inefficient by Chai and Nayak [28]. Another typical hash-based perturbation method was proposed by Wang et al. [26] which is named the optimal local hash (OLH). In this method, the choice of the size of the hash digest is optimized where the hash digest is perturbed using the GRR.

Discussion. Although hash function-based solutions may have a smaller communication cost and improved statistical variance due to the smaller hash digest size, they may suffer from the collision problem. In order to alleviate the effect of such collision, two typical techniques may be used, namely Bloom filter [9] and Count Mean Sketch [29]. The collision probability can be further reduced by a technique proposed by Erlingsson et al. [9] where users are permanently assigned to m different cohorts, each having a different set of hash functions. Although all techniques discussed may reduce the effect and the probability of a collision, they increase the computational complexity of the decoding process.

Transformation. Another technique that is typically used to estimate the frequency is the use of carefully designed transformations to transform A to A', which allows for aggregation to be done using some specific properties of A'. The set A' is typically represented by a matrix $\Phi \in \mathbb{R}^{m \times d}$ where values in A are transformed to the columns of Φ. More specifically, let $\{\mathbf{e}_1, \ldots, \mathbf{e}_d\} \subseteq \mathbb{R}^d$ be the standard basis of $\mathbb{R}^d$ where for $i = 1, \ldots, d, \mathbf{e}_i$ has zero entry everywhere except its ith entry, which has value 1. Then for $I_i \in A$, we transform I_i to the ith column of $\Phi, \theta_i = \Phi \cdot \mathbf{e}_i \in \mathbb{R}^m$. For each user u_i that holds $v_i = I_{k_i} \in A$, the report is defined by first randomly choosing an index $j \in \{1, \ldots, m\}$ using some distribution χ_i that depends on v_i. Having j, the user can then report $\langle j, (\theta_{k_i})_j \rangle$ to the data collector. The data collector then uses an aggregation mechanism to calculate a vector G of length m from the reports by the users. The vector G can then be used to estimate the frequency of each value v_i through the calculation of some inner products between G and each θ_i. In [17], the matrix used is defined as $\Phi \in \left\{ -\frac{1}{\sqrt{m}}, \frac{1}{\sqrt{m}} \right\}^{m \times d}$. Some other works [29,30] considered the use of the Hadamard transform matrix to define Φ. In the following sections, we denote the two approaches by JLRR and HRR respectively.

Discussion. Transformation-based methods (JLRR, HRR) transform the user's value from d bits to only 1 bit. It reduces communication cost significantly by only reporting an indicator j and 1 bit value. However, one bit of data might not represent the complete input information. The accuracy can be affected, especially when the privacy budget ϵ is big. The information loss during the transformation process dominates the statistic error.

Subset selection. The next technique that has been considered in designing a frequency estimation scheme is named the subset selection. In general, such a scheme is mainly done by letting user u_i randomly select and report a subset of A of size k with a predetermined distribution depending on the value v_i it holds. Wang et al. [32] determines the optimal value of k that minimizes the statistical error of the estimate. In their work, they established that this can be achieved by setting $k = \left\lfloor \frac{d}{e^\epsilon + 1} \right\rfloor$ or $k = \left\lceil \frac{d}{e^\epsilon + 1} \right\rceil$. In 2019, they proposed a mechanism that pads the output with trivial output which can be seen as a variant of the Exponential mechanism and an extension of the k-subset mechanism to handle discrete quantitative data [34]. In 2018, Ye and Alexander [33] analysed the k-subset mechanism in a medium privacy regime and provided a tight lower bound on its minimax risk.

Discussion. In general, subset selection-based schemes perform well in intermediate privacy regimes, say, when $\epsilon \in [\log 2, \log(d-1)]$ [32]. However, this comes with a relatively high communication cost, especially when ϵ is small.

Summary. The frequency estimation problem is one of the most basic statistical analyses and has been one of the problems that are the most extensively studied in local differential privacy setting [35]. Besides proposing new perturbation mechanisms to reduce the statistical variance, given prior knowledge regarding the noise and item distribution, further calibration can be made to improve frequency estimation schemes as well [36,37]. The same can be done by performing post-processing focusing on maintaining consistency [38–40]. The comparison between different methods and techniques in the study of the frequency estimation problem can be found in Tables 1 and 2 respectively.

3.1.2. Heavy hitters identification

Heavy hitter identification is a similar problem to the frequency estimation problem where we only aim to identify the items with sufficiently high frequency. Proposed schemes in handling the heavy hitter identification to estimate the top k frequent items have focused on both accuracy and efficiency. In general, such schemes can be divided into three based on their methods. A summary of such methods can be found in Table 3.

Naive method. As has been observed above, the problem of heavy hitters identification is similar to the frequency estimation problem where it only outputs the frequency of items that appear sufficiently frequently. A straightforward solution can then be designed by first having the aggregator estimates the frequency for all items and output only the frequency of the heavy hitters.

Discussion. The native method requires the estimation of the frequency of all data in the dataset. So this approach may become infeasible when the dataset is large. Furthermore, if direct perturbation solutions, such as GRR, are utilized, the statistical variance of the

Table 2
Comparison among typical techniques for frequency estimation.

Technique	Encoding	Perturbation	Variance	Communication (bits)
GRR [24]	$t = v$	$Pr[\hat{t} = v] = \begin{cases} \frac{e^\epsilon}{e^\epsilon+d-1}, & \text{if } t = v \\ \frac{1}{e^\epsilon+d-1}, & \text{if } t \neq v \end{cases}$	$O\left(\frac{d-2+e^\epsilon}{(e^\epsilon-1)^2}\right)$	$O(\log d)$
OUE [26]	$t = [0, \ldots, 1, \ldots, 0]$, where $t[v] = 1$	$Pr[\hat{t}[i] = 1] = \begin{cases} \frac{1}{2}, & \text{if } t[i] = 1 \\ \frac{1}{e^\epsilon+1}, & \text{if } t[i] = 0 \end{cases}$	$O\left(\frac{4e^\epsilon}{(e^\epsilon-1)^2}\right)$	$O(d)$
RAPPOR [9]	$\mathcal{H} \in \mathbb{H}$; $t = [0, \ldots, 1, \ldots]$ where $t[i] = \begin{cases} 1, \text{if } \mathcal{H}(v) = 1, \\ 0, \text{ otherwise} \end{cases}$; $r = \langle \mathcal{H}, t \rangle$	$Pr[\hat{t}[i] = 1] = \begin{cases} 1 - \frac{1}{2}f, & \text{if } t[i] = 1 \\ \frac{1}{2}f, & \text{if } t[i] = 0 \end{cases}$, where $f = \frac{2}{e^{\epsilon/2}+1}$	$O\left(\frac{e^{\epsilon/2}}{(e^{\epsilon/2}-1)^2}\right)$	$O(k)$
OLH [26]	$\mathcal{H} \in \mathbb{H}$; $t = \mathcal{H}(v)$; $r = \langle \mathcal{H}, t \rangle$	$Pr[\hat{t} = \mathcal{H}(v)] = \begin{cases} \frac{e^\epsilon}{e^\epsilon+g-1}, & \text{if } t = \mathcal{H}(v) \\ \frac{1}{e^\epsilon+g-1}, & \text{if } t \neq \mathcal{H}(v) \end{cases}$, where $g = e^\epsilon + 1$	$O\left(\frac{4e^\epsilon}{(e^\epsilon-1)^2}\right)$	$O(\log n)$, n is the number of users
JLRR [17]	$\Phi \in \left\{-\frac{1}{\sqrt{m}}, \frac{1}{\sqrt{m}}\right\}^{m \times d}$; $i \in [m]$; $t = \Phi[i, v]$; $r = \langle i, t \rangle$	$\hat{t} = \begin{cases} c_\epsilon dt, & \text{w.p. } \frac{e^\epsilon}{e^\epsilon+1} \\ -c_\epsilon dt, & \text{w.p. } \frac{1}{e^\epsilon+1} \end{cases}$, where $c_\epsilon = \frac{e^\epsilon+1}{e^\epsilon-1}$	$O\left(\frac{4e^\epsilon}{(e^\epsilon-1)^2}\right)$	$O(\log m)$
HRR [29,30]	$\Phi : 2^d \times 2^d$ Hadamard Matrix, where $\Phi[i, j] = 2^{-d/2}(-1)^{\langle i,j \rangle}$; ($\langle i, j \rangle$ refers to the inner product of their binary representations) $i \in [2^d]$; $t = \Phi[i, v]$ $r = \langle i, t \rangle$	$Pr[\hat{t} = 1] = \begin{cases} \frac{e^\epsilon}{e^\epsilon+1}, & \text{if } t = 1 \\ \frac{1}{e^\epsilon+1}, & \text{if } t = -1 \end{cases}$	$O\left(\frac{4e^\epsilon}{(e^\epsilon-1)^2}\right)$	$O(\log m)$

Table 3
Comparison of heavy hitter identification methods.

Method	Typical papers	Description	Advantages	Disadvantages
Naive method	[24,26]	Apply frequency estimation method directly to all items to find the sufficiently frequent ones	Easy to perform and no additional computation	Not efficient
Segmentation-based method	[41,42]	Data string is transformed to several shorter segments (that may overlap) where users perturb and report one or several of the segments	High efficiency for heavy hitter estimation	Additional computation cost, low accuracy
Tree-based method	[43,44]	Iteratively estimate the frequency of different prefixes, pruning the less frequent ones and growing the tree by considering longer prefixes	High efficiency, dataset size knowledge is not required	Multiple iterations requiried

estimate is also very high, significantly reducing the statistical accuracy of the estimate.

Segmentation-based method. Segmentation-based methods attempt to alleviate the disadvantages that are obtained from the use of frequency estimation methods in the naive methods discussed above. More specifically, this is done by first partitioning the encoded data into g partitions. The server finds the frequent strings in each segmentation C_i and then gets the candidate set $C = \bigcap_{i=1}^{g} C_i$. One consideration in such design is the size of the candidate set C. Various works have considered different techniques to further reduce the size of C. Fanti et al. [41] consider the possibility of having each user report two random segments instead of just one. A potential problem of the technique above is in the case that g is big, the number of reports for each segment may be small, which results in estimation with low statistical accuracy. Wang et al. [42] proposed a solution to such potential problem by having the segments overlap. Kim et al. [45] utilize such method to find frequent words based on users' keystroke data. More specifically, in their work, they consider that each user tags a hash value to the word to enable an integrity check and sends one random segment of the word based on the segmentation idea discussed before. The aggregator can then estimate the frequent strings for each segment which can then be combined.

Discussion. The segmentation-based method improves the efficiency of naive heavy hitter solutions by reducing the number of items considered from d to at most $g2^{\frac{\log d}{g}}$, which is achieved in the original segmentation-based solution of having each user reports 1 segment. However, this comes with an increase in computational complexity, which comes from the processing steps such as the construction of the candidate set. Furthermore, in general, the elements in the candidate set may not correspond to actual elements of the dataset, which further affects the accuracy of the scheme.

Tree-based method. The idea of the tree-based method is similar to the segmentation-based solution proposed by [42] where prefixes are iteratively generated for the values with high frequency under the principle that data that appears frequently also has its prefix appearing frequently. Tree-based methods are generally used for "string data" such as trajectory and English words. Intuitively, starting from a root with an empty string, we can build a tree where each node is based on possible values in the next segment of the string. Such an approach can be shown to have a much improved efficiency. Similar to the previous approaches, tree-based methods may have the domain size for each iteration to be very high, which affects the statistical accuracy of the frequency estimate. Bassily et al. [43] solved this by having the local randomizer be used twice in the full protocol. More specifically, after several items have been identified to have a high estimated frequency using the iterative method discussed above, a local randomizer is again invoked to these items to get a better estimation of their frequency. Wang et al. [44] presented an alternative candidate set construction method which is applied in each node on the tree which restricts each

Table 4
Comparison of methods for joint distribution estimation.

Method	Typical paper	Computation complexity	Communication cost	Variance	Advantages	Limitations
Naive method	–	High	High	$2^m \cdot \text{Var}$[a]	• Compute any k-way marginals	• High variance • Inefficient
EM	[41,46]	High	High	$2^m \cdot \text{Var}$	• Compute any k-way marginals • Much accurate	• High variance • Inefficient
Lasso regression	[46]	Medium	High	$2^m \cdot \text{Var}$	• Compute any k-way marginals • Much efficient	• High variance
Fourier Transformation	[47]	Medium	Low	$\sum_{i=1}^{t} \binom{m}{i} \cdot \text{Var}$	• Low communication cost	• Predefine k
Subset selection	[48]	Medium	Medium	$\frac{m}{n} \cdot 2^{\ell} \cdot \text{Var}$	• Compute any k-way marginals • Low variance	• Extra errors

[a] Var represents the variance of estimating the frequency of a possible value of an attribute.

user to only report once along the path that corresponds to his private value. This restriction is done to help in limiting the privacy budget requirement.

Discussion. In general, tree-based solutions can estimate the frequent items efficiently and they can be applied without the need of the users to know the size of the dataset. However, such solutions need multiple iterations, which increases their communication cost and delay. Furthermore, current tree-based solutions reduce the statistical variance of the estimate by partitioning users into disjoint groups. In such case, some groups may not have a sufficient number of users to produce estimates with high statistical accuracy. This is especially true when the number of groups is large.

Summary. The main challenge in the study of heavy hitter identification is to improve the efficiency of the scheme while maintaining statistical accuracy. Currently, proposed solutions do this by removing items with low frequency iteratively. Although such a direction may solve the efficiency problem, it causes some new potential problems including high computation complexity and communication costs. Further improvements on the performance of privacy-preserving heavy hitter identification remain a challenge to be considered. Furthermore, formal evaluation of efficiency improvement in such solutions also still needs to be formally defined and remains an open problem.

3.1.3. Joint distribution estimation

Joint distribution estimation is shown to be essential in various machine learning models training and basic inference to capture the correlation between different attributes [33,49,50]. In the following sections, we denote by k-way marginal, the joint distribution over a subset of k attributes.

Naive method. It is easy to see that when we treat the data record with m attributes as one data item, this problem can be reduced to the frequency estimation problem discussed before. Hence both the joint distribution of all attributes as well as the k-way marginal table can be estimated using general frequency estimation methods on the larger dataset.

Discussion. Since the m-dimensional dataset will have a much larger size, it is clear that such method is not optimal. Furthermore, if directly apply the traditional frequency estimation solutions, the large size of the dataset also increases the statistical variance of the estimate. Hence, the statistical accuracy of the estimate may not be high. Lastly, due to the large dataset, the schemes will also have very high time and space complexity.

Expectation maximization (EM). Alternatively, instead of reporting the data record with m values as one item, EM methods let the user report the m values separately with a split privacy budget. The aggregator estimates the marginal distribution for each attribute separately. The problem is reduced to the problem of estimating a joint distribution of all attributes given a potentially corrupted set of the marginal distribution. Expectation–Maximization (EM) algorithm is generally used to produce unbiased maximum likelihood estimates (MLE)s of the joint probabilities given such incomplete data. Intuitively, the EM algorithm starts from an initial estimate of the joint distributions and iteratively updates them using the estimates of the marginal distributions. This method was first proposed by Fanti et al. [41] for a dataset with two attributes and was later extended to handle a larger number of attributes by Ren et al. [46].

Discussion. In contrast to naive methods, EM-based methods output estimates with higher statistical accuracy. However, similar to naive methods, EM-based methods have high time and space complexity. Lastly, it is easy to see that the convergence rate of the EM algorithm is affected by the choice of the initial distribution, which was chosen to be the uniform distribution in both works of [41,46]. Given some prior knowledge about the dataset, it is interesting to investigate if different initial distributions may be chosen to improve the convergence rate.

Lasso regression-based method. The next method for the joint distribution estimation problem is based on the regression technique, which was first considered by Ren et al. [46]. Let $\boldsymbol{\beta}$ be a vector containing all possible joint distributions of the m attributes. We further let $\mathbf{y}$ be a vector containing all the marginal distributions for each of the m distributions. By the law of total probabilities, we can define a matrix M such that $\mathbf{y} = M\boldsymbol{\beta}$. Recall that our aim is to estimate $\boldsymbol{\beta}$ based on a noisy estimate of $\mathbf{y}$. It is then easy to see that we can model this as a linear regression problem. Hence, such a problem can be solved by, for example, using Lasso regression [27].

Discussion. Although Lasso regression-based methods have better efficiency compared to the previous best solutions, they do not produce more accurate estimates. This approach has been extended to the problem of synthetic dataset generation [46,51,52].

Fourier Transform. Joint distribution estimation can also be solved using Hadamard transformation technology, which was proposed by Cormode et al. [47]. This method was originally proposed for the case when the m attributes are of size 2, allowing each value to be represented by a bit and the m attributes to be represented as a binary string of length m. By any ordering of binary strings of length m, such data can then be further represented as a binary vector of length 2^d with Hamming weight 1. Having this, transformation can be done to represent each string in a different set of orthonormal basis. In particular, when transformed using the Hadamard transform, the resulting representation requires a small number of coefficients to calculate the k-way marginals. In particular, the number of coefficients required to compute any particular k-way marginal is $\sum_{j=0}^{k} \binom{m}{j}$. *Discussion.* In general, Fourier transform-based solutions have improved the previous solutions in terms of communication cost and its statistical variance is also much lower when k is small. This advantage, however, no longer applies when k is large, where $O(m^k)$ coefficients need to be estimated. This in fact also increases the computation cost. Lastly, as has been previously discussed, such method is specifically designed for values with the binary dataset for each attribute. Therefore, to apply such method in a more general setting, each attribute needs to be transformed into several attributes, each having two possible values. This, in turn, increases the values of m and k.

Subset Marginal Selection. An alternative way to estimate the joint distribution of m attributes is to have users report a subset of the m attributes with some predetermined size ℓ. This is the main idea behind the scheme CALM, which was proposed by Zhang et al. [48]. More specifically, the aggregator generates M different views, which are subsets of attributes, each of size ℓ. The aggregator then assigns each user to one of the subsets, representing the ℓ attributes that each user needs to report. Then, each user can report the perturbed ℓ attributes assigned to him. Each of such views can then be used to construct the marginal table which can further be used to estimate other k-way marginal.

Discussion. In general, subset marginal selection-based solutions enable the estimation of any k-way marginal table for any k without the need for the estimation of the full marginals. They also can be applied to non-binary attributes. On the other hand, due to the small amount of extracted information, the statistical error of the estimate tends to be large. This is due to the following reasons. Firstly, in addition to the error introduced by the perturbation for privacy-preserving purposes, the information collected for each user is only partial. This causes further errors throughout the sampling step. Secondly, the construction of the marginal distribution may not be based on the full information required. Because of this, it may introduce further construction errors.

Summary. In general, a joint distribution estimation scheme has a much higher statistical variance compared to a frequency estimation scheme for a single attribute. It is generally also more sophisticated, in particular when marginal distributions on some of the attributes also need to be estimated. It is also a challenge to find a good balance between complexity, functionality and accuracy. A brief summary of the current work on the different methods can be found in Table 4.

3.1.4. Mean value estimation

The next statistical query we consider is defined over numerical data. The statistical query we are considering in this section is the calculation of the average values for all users. Such problem naturally extends to the multi-dimensional cases where the statistical query can then be extended to the point-wise average of the m dimensional vectors. In addition to two traditional numerical data perturbation mechanisms, Laplace and Gaussian mechanisms, there have also been two other families of mean value estimation schemes that have been proposed; extreme values perturbation and distribution perturbation.

Extreme values perturbation. Intuitively, solutions based on extreme values perturbation have restricted the possible reports of any users to just one of two extreme values where the distribution of the possible reports is based on the value held by the user. The choices of possible reports and probability distribution are made to ensure that the report is an unbiased estimator of the user's real data, guaranteeing that their sum is also an unbiased estimator of the sum of the value. Duchi et al. [53] then extend this method to a more general m dimensional data $\mathbf{t} \in [-1,1]^m$. The possible reports are then defined to be the set $\{-B, B\}^m$ following a distribution χ where B is defined as a function of m and ϵ while χ depends on m, ϵ and $\mathbf{t}$. Inspired by the multiple attribute variant proposed by Duchi et al., Nguyen et al. proposed Harmony [54], which only requires each user to report one of the attributes. This modification allows Harmony to provide a mean value estimate with a similar statistical variance while requiring a much smaller communication cost. The work of Akter and Hashem [55] extends the solution by Duchi et al. to the setting where different users have different privacy concerns. Lastly, Wang et al. [56] make an adjustment to the probability distribution from the multi-dimensional dataset variant of the solution of Duchi et al. to satisfy a less restrictive privacy requirement of (ϵ, δ)-LDP.

Discussion. Relative to other solutions, the solution by Duchi et al. outputs an estimate with relatively small variance when ϵ is small. However, their scheme has a characteristic that regardless of the value of ϵ, the statistical variance of the estimate $\hat{t}_i$ of each user's data t_i is $\left(\frac{e^\epsilon+1}{e^\epsilon-1}\right)^2$, which is always larger than 1. Hence, compared to other protocols, which generally can reduce the statistical variance arbitrarily close to 0 by increasing ϵ, such solution by Duchi et al. will perform worse when ϵ is sufficiently large. Furthermore, Kairouz et al. [57] showed that in the setting of two possible outputs utilized by Duchi et al., the choice of the two possible reports used by Duchi et al. does not always provide an estimate with the smallest statistical variance, especially when ϵ increases. Because of this an interesting matter to consider in Extreme Values Perturbation is to explore the possibilities of using different choices of reports as well as their probability distribution.

Distribution perturbation. An alternative approach to the mean estimation mechanism is first proposed by [58], called Piecewise Mechanism (PM). Similar to Laplace or Gaussian-based approach, this approach generates a report from a continuous range with a distribution that depends on the private value. Recall that by the encoding previously discussed, each user u_i holds a private value $t_i \in [-1,1]$. Instead of letting the report be any real number as Laplace or Gaussian mechanism, the output range is set to be $[-s, s]$ for some predetermined value $s > 1$. Furthermore, instead of having the output distribution be continuous, the density function is designed to be piecewise constant. More specifically, the scheme first defines two functions $\ell : [-1,1] \to [-s, s]$ and $r : [-1,1] \to [-s, s]$ such that for any $t \in [-1,1], -s \leq \ell(t) \leq t \leq r(t) \leq s$. Having these, the distribution is defined such that the density function is constant c_1 in $[\ell(t), r(t)]$ and another constant c_2 in $[-s, s] \setminus [\ell(t), r(t)]$ such that $c_1 > c_2$. The extension of such approach to m-dimensional data is also considered in [58]. A similar modification to the m-dimensional case can be done by letting each user only report k out of the m attributes. Such modification, which is named the Square Wave mechanism, was proposed by Li et al. [39] to estimate the distribution of the data. In this approach, a different encoding is used where the value is encoded to $t \in [0,1]$. Having this encoding, the domain becomes $[-b, 1+b]$ for some b which is determined based on the mutual information between the input and output of the scheme. Having b, using the notations above, $\ell(t)$ and $r(t)$ are then defined as $t-b$ and $t+b$ respectively. Furthermore, c_1 and c_2 are defined to be $\frac{e^\epsilon}{2be^\epsilon+1}$ and $\frac{1}{2be^\epsilon+1}$ respectively. We note that in this approach, t is always in the centre of the interval $[t-b, t+b]$ with higher probability while the position of such interval is not always in the centre of the overall output range $[-b, 1+b]$. Hence, such approach will not provide an unbiased estimation when used to estimate the mean of users' value.

Discussion. In general, the distribution perturbation method provides better performance in the high privacy regime. However, it usually comes with a more complicated computation and the perturbation process may also be more involved.

Summary. In local differential privacy settings, queries over numerical data have not been as extensively studied as queries over categorical data. The solutions discussed above are generally only optimal in specific privacy regimes and there has been no design that can be optimal over the whole privacy regime. A natural way to combat this is through the use of different solutions depending on the privacy requirement. This is the main idea behind the Hybrid Mechanism (HM) proposed by Wang et al. [58] where the appropriate mechanism is adaptively chosen based on the privacy need. Another limitation for local differentially private statistical analysis over numerical data comes from the large bias introduced for each data record. Because of this, designing algorithms for more sophisticated statistical queries such as max, min, and quantile are also quite challenging.

3.1.5. Range query estimation

A range query counts have two inputs ℓ, r such that $\min\{D\} \leq \ell \leq r \leq \max\{D\}$ and we want to estimate the proportion of the users holds v_i such that $\ell \leq v_i \leq r$. The aggregator may estimate the frequency for each value in D and sum up the frequency of the values within the desired interval. Although such method may work well when d is small, its accuracy will quickly worsen when the range m increases due to the accumulation of variance from the estimation of each value. It

can be easily seen that using a point estimation method with variance Var, the overall range query estimation will have $m \cdot \mathrm{Var}$. Currently, the main method of solving such problem is the hierarchy-based method, which produces an estimate with statistical variance bounded above by a polylogarithmic function of the length of the range.

Hierarchy-interval. The general idea of the hierarchy-interval-based method is to construct a b-ary tree of height h. The users are partitioned into h groups where the users in the ith group use the local randomizer to report the encoding of their value in the ith level of the tree. The aggregator can then use the estimates for each node to estimate the desired range. The hierarchical-interval-based approach in a local differential privacy setting was first proposed by Cormode et al. [59,60]. In this work, the sub-intervals that are used to decompose the data domain are defined to be the B-adic intervals. A further post-processing approach is also considered to improve the accuracy further. More specifically, such post-processing updates the estimates of each node while ensuring consistencies between the count in a parent node with the total count of its children nodes. This approach was then extended by Wang et al. [61] to multi-dimensional analytical queries where each value has m attributes. In their work, such query is handled by applying Hierarchical-interval based method to each attribute separately. The choice of sub-intervals to be used for the data domain decomposition needs to be done carefully. For example, a uniform decomposition may cause some of the sub-intervals to have low frequency, which causes its estimate to have high noise. Such challenge was considered by Du et al. [62] in their work of Adaptive Hierarchical Decomposition (AHEAD) protocol where the granularity of the domain composition is adaptively determined to reduce the impact of the perturbation towards the statistical accuracy of the estimate. Wang et al. [63] employ a dynamic decomposition approach, customizing the granularity of each domain to create distinct sub-domains. This strategy intricately accounts for the inherent variability in range query responses, enabling more precise and informed answers.

Discussion. Hierarchy-interval-based method improves the naive method in terms of its statistical variance. More specifically, given that the variance of the point estimate algorithm is Var, compared to the naive method which produces an estimate with the statistical variance of $O(I)\mathrm{Var}$, hierarchy-interval-based method produces an estimate with the statistical variance of $O(\log_b(I)\mathrm{Var})$.

Hierarchy-coefficient. An alternative way to estimate range queries is through the discrete Haar transform (DHT), which is a discrete version of the Haar wavelet transform (HWT) [64]. The encoding of a value $v \in \mathcal{D}$ is done by representing it as a binary tree of height $h = 1 + \lceil \log d \rceil$ as described in the introduction of this application. A series of coefficients, named Haar coefficients, are then generated and assigned to the nodes of the tree iteratively from its leaves. Such coefficients are used to reconstruct the count of any leaf node. Given the binary tree as well as the coefficients assigned to each node, similar to the hierarchy-interval method, the DHT-based method can then partition the users to h groups where users in the ith group will apply general frequency estimation method to report its encoded value at level i. Such mechanism was first proposed by Cormode et al. [60] where the general frequency estimation method used is the Hadamard Randomized Response (HRR). Having such perturbed reports, the Haar coefficients can then be used to estimate the count for each node, which can then be used to estimate the count for the desired range.

Discussion. Given the height of the binary tree, denoted as h, hierarchy-coefficient-based method only require at most $2h$ nodes to make the estimation. It provides an upper bound to both the complexity and statistical variance of the estimate. It can be shown that the scheme produces an estimate with statistical variance bounded above by a polylogarithmic function of d.

Summary. Range query is a very common database operation where the proportion of the records within a specified interval is required. A naive solution of using the point estimation method to estimate the frequency of all possible values before aggregating those within the required interval produces an estimate with a large statistical variance. In the literature, hierarchy-based methods have proved to be effective in solving such problem under LDP protection. When considering the natural extension of the dataset with multiple attributes, the majority of works only consider a very low-dimensional dataset (1 to 2). It should be noted that when considering a higher dimensional range query, different strategies [62,65] may need to be explored in dealing with the large dimension of the attributes.

3.2. Specific data statistics

In this section, we consider applications where the data may have a more general form.

3.2.1. Set-valued data

The first scenario we consider is when users hold a set of up to ℓ items (e.g., browsed web pages and purchased goods). In the literature, there are mainly two different statistical queries on set-valued data; item frequency and set frequency. These queries correspond to two common data mining tasks, frequent item mining and frequent itemset mining.

Frequent item mining. The frequent item mining problem provides a parameter k and requires an estimate of the set of items of size k containing the items that appear in the most number of different item sets. One of the main challenges in the study of such problem is the possibility that the sets held by the users may have a heterogeneous size where each user may hold 0 up to ℓ items. Such variety produces different sampling distributions for different users, which makes it difficult for the aggregator to process the report, which further complicates the effort to produce an accurate frequency estimation.

In the state-of-the-art solutions to such problems [66,67], trimming and padding are required to combat the problem of the user's dataset having a heterogeneous size. In general, the aggregator defines a parameter m to be the estimate of the largest size that the set a user holds may have. This m is then used to transform the user's data V_i to a set $\hat{V}_i$ of size m. If a user holds more than m items, some items can be removed to obtain a set of size m. On the other hand, when a user holds less than m items, dummy items may be added until the set has size m. After $\hat{V}_i$ is generated, each user can then sample one item from $\hat{V}_i$ to be reported to the aggregator. In order to enhance the accuracy of such approach, Qin et al. [66] propose LDPMiner, which is a two-phase mechanism. In the first phase, a candidate set of top k frequent items is identified using a part of the privacy budget while the remaining privacy budget is used in the second phase to refine such candidate set. Wang et al. [68] further reduce the privacy budget requirement by using the privacy amplification of sampling. Furthermore, Wang et al. [67] proposed the use of the exponential mechanism to allow for the report of a subset of the items without the need of splitting the privacy budget. Such approach improves the accuracy of estimation. However, the problem of choosing the size of the reported subset is still an open problem.

Frequent itemset mining. This problem tries to find the itemsets with a frequency over a given threshold. Note that this problem can be reduced to heavy hitters identification. However, it is easy to see that one of the main challenges for such problem is the size of the domain which may result in estimates with large statistical variance. More specifically, using the notation above, it is easy to see that there are $\binom{d}{\ell} + \cdots + \binom{d}{0}$ different subsets of size at most ℓ, where d is the number of items in the dataset. Such size may quickly become infeasibly large when d and ℓ grow. Hence, one of the main considerations in the study of frequent itemset mining is to identify approaches that may reduce the size of the domain. One of the solutions under LDP setting was proposed by Wang et al. [68]. Their solution is based on the frequent item mining scheme. More specifically, they first use a frequent item mining scheme to identify items with high frequencies. Under the assumption that sets that appear frequently must also contain items that appear frequently, the candidate items obtained from the frequent

item mining scheme can then be used to construct candidate sets with high frequency. One limitation of such approach is the need to split the privacy budget into multiple steps, which causes the estimate to have a higher statistical variance. Such problem is still an open problem and further investigation is required.

Discussion. The key feature of set-valued data is that users may hold sets of different sizes. The common approach of truncation and padding to allow for data aggregation introduces a further problem. As has been discussed, users with data with a size less than the estimated m need to add dummy items, which should not contribute to the statistical calculation. Hence, when m is large, many users may need to perform many padding, which reduces the proportion of effective sampling. On the other hand, if m is too small, many users may need to truncate a large number of items that they hold. This may cause the sampled item not to be a good representation of users' real data. Although the choice of m may be essential, the schemes that have been proposed in the literature have chosen m without any theoretical support. A theoretical analysis of the choice of m is hence essential and is currently still an open problem.

3.2.2. Key–value data

Key–value data is a hybrid data model used in the popular NoSQL which has been widely utilized in practice. Each data in this form is a pair (k, v) where k is an element from a finite categorical set $\mathcal{K}$ and v belongs to a continuous domain $\mathcal{V}$ which can be assumed to be $[-1, 1]$. In such case, each user u_i holds a set of ℓ_i pairs $\{(k_{i,1}, v_{i,1}), \ldots, (k_{i,\ell_i}, v_{i,\ell_i}) : k_{i,j} \in \mathcal{K}, v_{i,j} \in \mathcal{V}\}$. In the existing work, there is an obvious but not enunciated assumption that for each i, $k_{i,j} \neq k_{i,j'}$ for any $1 \leq j < j' \leq \ell_i$. In such form of data, the most general queries include frequency estimation of the keys as well as the mean value estimation of the values linked to a specific key. The problem of estimating the frequency of different keys focuses on the number of users that hold key–value data with the specified key–value. On the other hand, the problem of mean value estimation of values linked to a specific key focuses on the average of all the values corresponding to the desired key that is held by the users. It is then easy to see that the frequency estimation problem can be solved by a direct application of the traditional frequency estimation method. On the other hand, the mean value estimation problem cannot be directly solved by the mean value estimation methods we previously discussed. This is due to the condition that any value that is considered in the calculation must have a specific key.

For a fixed key $k \in \mathcal{K}$, assume the user u_i holds a key–value data (k, v_i), the key–value pair is encoded as $(1, v_i)$ in the literature. If the user does not hold the key k, the corresponding pair is added and is possibly encoded as $(0, 0)$. Given such encoding and the possible addition of dummy items, various perturbation schemes are proposed. Ye et al. [69,70] proposed a perturbation where the first entry is perturbed with a predetermined distribution and any dummy items with the first entry perturbed to one has its second entry being randomly sampled from $[-1, 1]$. Such perturbation is then updated iteratively to improve the accuracy of the mean estimate. In contrast, Gu et al. [71] any dummy item with its first entry being perturbed to one has its second entry being uniformly sampled from the set $\{-1, 1\}$. This ensures that the expected value of such items is 0. Furthermore, they also proposed an optimized privacy budget allocation scheme to improve the accuracy of the estimate further. In their work [72], Sun et al. [72] proposed some approaches to dealing with this problem. Among them, there is an approach that may provide a different direction of approach. More specifically, in the previous work, if after the perturbation step, the first entry of a pair is 0, its second entry is automatically set to 0. This is to represent the case when the corresponding user does not have any key–value data with the specified key and hence should not affect the mean calculation. In this work, they propose that in such case, the second entry is initialized at a predefined default value before also being perturbed. To reduce the effect of dummy items towards the report while making full use of the user's key–value data, Gu et al. [71] proposed a padding and sampling-based protocol where with some probability, sampling is done over all other key–value data that the user holds.

Discussion. In general, there are two critical issues in the investigation of key–value data statistical analysis. Firstly, there is a need to consider the correlation between the key and its related value. Secondly, generating dummy items needs to be done carefully so its impact on the final estimate can be minimal.

3.2.3. Stream data

The next data format we will consider is dynamic data that changes over time. In such form of data, multiple report collection over a long period of time may be required to obtain the latest data [80] and enable the observation of the statistical trend of the data.

A straightforward collection of data for multiple timestamps implies the gradual increase of privacy cost. A well accepted technique is the memoization technique. Specifically, for any value B that a user may hold, the user perturbs it to obtain B'. Having this, every time a report is required, instead of generating a fresh random perturbation, the user can always report the generated perturbed report following his actual value. Erlingsson et al. [9] proposed a two-staged randomized-response-based mechanism. Intuitively, this is done by first performing the memoization technique by applying randomized response to obtain B' based on the original value B. Having B', reports can then be generated by perturbing B' further. In order to handle data with frequent changes, Ding et al. [11] proposed an α-point rounding method. In this work, the data is discretized where rounding is done using parameter α. After the discretization of users' private value, memoization can then be performed. However, such changes are still limited to small changes where significant changes may cause the performance to worsen.

Another line of investigation on the statistical analysis of dynamic data focuses on noise scale reduction. Joseph et al. [73] propose a private voting mechanism to estimate the update, which only happens when the statistic is significantly inaccurate. The proposed scheme is only shown to be applicable for scenarios with a static number of users whose private data is sampled from a distribution, for instance, the Bernoulli distribution. Works [81,82] considered the stream data with a fixed length. Instead of reporting all the data points, users identify salient points of their data locally and only report such points to the aggregator, then estimate the others. Xue et al. [74] utilized binary trees in the dynamic recording of the differences between data over different times. This allows the scheme to reduce privacy consumption, in particular when no data change is observed in some period of time. Bao et al. [75] considered the correlation between the users' data over time. In order to utilize such correlation, the perturbation for each timestamp is obtained by the linear combination of noises generated in several previous timestamps and a fresh random noise generated for this timestamp. Such approach was shown to significantly reduce the noise scale, especially in the case when the correlations between such data items are strong. Cunningham et al. [83] considered the use of publicly available external knowledge to adjust the distributions in some perturbation steps to enhance the utility of the estimate. Instead of considering the budget division method to ensure a certain level of privacy, Ren et al. [76] proposed a population division framework for infinite streaming data collection. This is based on an observation from previous studies [26,42] in which schemes have a smaller statistical error when users are partitioned instead of splitting the privacy budget.

Besides, to improve the accuracy of stream data statistics, works [76,84,85] adopted the concept of ω-event privacy [86], which provides the privacy guarantee in a sliding window with size ω instead of the user (all data items in the stream). In [79,87], Erlingsson et al. and Li et al. considered the application of shuffling in a distributed system where the users' LDP reports are shuffled before further processing. This is shown to provide a strong central differential privacy guarantee to the scheme without any explicit server-side perturbation steps. Lastly, Ye et al. [88] introduced a new concept of LDP in temporal setting (TLDP)

Table 5
Summary of stream data collection.

Typical papers	Strategy	Privacy	Applicability to infinite stream	Constraint
[9]	Memoization	ϵ-LDP	Yes	Data cannot change frequently
[11]	Memoization	ϵ-LDP	Yes	Data cannot change significantly
[73]	Voting	ϵ-LDP	No	Data drawn from a specific distribution, static number of users
[74]	Privacy budget suppression	ϵ-LDP	Yes	Data values in the stream do not change often
[75]	Noise reuse	(ε, δ)-LDP	Yes	The correlation information between data items is publicly known, data changes over adjacent timestamps is within a public predetermined constant C
[76]	Population division	ω-event LDP	Yes	–
[77]	Threshold truncation	event-level LDP	Yes	The distribution stays the same
[78]	User contribution truncation	user-level LDP	Yes	–
[79]	Privacy amplification	ϵ-LDP, shuffling	No	Data values can only change by 1 and change at most k times at given time period

for time series statistics. In such setting, two time series are defined to be neighbours if they have small numbers of different timestamps. Furthermore, in order to provide privacy, in contrast to other approaches that perturb the data values while keeping the timestamps of each data the same, TLDP protects the data by perturbing the timestamps instead.

Discussion. In general, works over data stream are shown to provide reasonably accurate estimates. However, such works usually have some strong assumptions on the data stream which cause their practicality to be low. For instance, although the memoization technique allows the aggregator to request multiple reports while maintaining some privacy guarantee, it comes with strong restrictions on the dynamicity of the data. This makes such approach hard to be practical.

3.2.4. Graph data

Graph Data includes nodes and edges where nodes represent users in the system while the relations between any pair of users are represented by an edge between the corresponding nodes. Apart from schemes dedicated to some specified statistical queries as have been considered in other data formats, there have also been interests in the study of the privacy-preserving generation of the synthetic graph containing statistical information of the private graph.

Synthetic graphs releasing. One common method in generating the synthetic graph is to construct the graph following certain graph models such as Erdős-Rényi graph model [89,90], Power-Law graph model [91] and Kronecker graph model [92]. In such works, the synthetic graph is generated based on some statistical information collected from the users such as the node's degree and other structural properties of the graph. There are mainly two challenges we need to consider in designing a locally differentially private scheme to generate such synthetic graphs.

Firstly, in general, such network may have a very large number of nodes. This implies that the information regarding each user's neighbours has a very large dimension. The grouping method is a technique that has been widely adopted in the literature to handle this problem. Intuitively, the grouping method is done to partition the users based on specific criteria where reports are only done within the subgraphs, significantly reducing the dimension of each report. In general, there are two different types of grouping methods based on the partition criteria. The first possible partition criteria is to group users based on the similarity of their structural information, as has been done in [93–95]. Alternatively, the clusters or communities in the network can also be used to partition the users such that each community is identified as a subgraph, as has been done in [96]. Such partition allows the size of each subgraph to be smaller and hence significantly narrows down the range of the reports. Despite such advantage, it should be noted that the grouping method only provides privacy protection for users within the group they belong to instead of over the whole set of users.

The second main challenge we need to consider comes from the fact that each report is generated locally by users without access to any information regarding any other users. Such limitation implies that the report that each user can generate will only capture local statistical and structural information of the graph, making it more challenging to use such reports to capture more general structural information of the graph. To combat this, Qin et al. [97] estimated and utilized node-to-group connectivity instead of simply the local statistical information of each user such as its number of connections. This allows them to obtain more complete structural information about the overall network. Wei et al. [98] proposed an optimization scheme by maintaining consistency of various information, including the attribute, structural, and community information, between the synthesized graph and the original private graph. By preserving such consistencies, more graph properties can be preserved.

Statistical query releasing. This section discusses some works that consider dedicated schemes designed for specific statistical analysis tasks.

Zhang et al. [99] considered the node statistical analysis by requesting information regarding the frequency of the node's coverage over all users. Several works [100–102] considered the subgraph counting problem which is a problem of counting the number of subgraphs of a given private graph with a specific form, such as a triangle or k-star. To investigate such problem, Imola et al. [101] proposed a two-round algorithm where the additional round of interaction between users and the aggregator is used to improve the statistical accuracy of the estimate. On the other hand, Sun et al. [100] considered the scenario where users are allowed access to the connections of their neighbours. Under this scenario, they proposed a Decentralized Differential Privacy (DPP) scheme which provides privacy protection for the data of a user's neighbours instead of only focusing on the data privacy of the users themselves. It is easy to see that due to the specificity of the designs, the main drawback of dedicated solutions is the difficulty of generalizing the solution to be applicable to other types of queries. A more generic solution was proposed by Ye et al. [103,104] to estimate two metrics of a graph, namely the adjacency bit vector and the nodes' degree of each user. Such general metrics have shown to be essential in deriving a wide range of other common metrics [105–107]. Hence, such solutions allow the aggregator to perform various graph statistical analysis tasks.

Recent research has delved into graph learning with local differential privacy, where the server gathers noisy adjacency information from users to train a classifier using this perturbed graph data. A common approach is to employ a randomized response to perturb the

adjacency vector. However, this can lead to a dense and noisy vector, which negatively impacts classification accuracy. To mitigate this, researchers have explored leveraging degree information for denoising the noisy adjacency vector [108]. Zhu et al. [109] employ Bayesian estimation to enhance the quality of the adjacency vector, while Lin et al. [110] suggest calibrating the perturbed adjacency matrix as a form of regularization during the training process.

Discussion. The release of a synthetic graph possessing statistical and structural information of the initial private graph provides a generic solution to various statistical queries. However, due to its genericity, compared to dedicated solutions for particular graph analysis tasks, the estimate produced by such synthetic graphs has lower statistical accuracy.

4. Private learning with LDP

In contrast to the statistical queries that have been discussed in the previous section, the objective we are considering in this section is more sophisticated where a machine learning model needs to be trained while preserving data privacy. In general, the training process can be classified into two different classes based on the information contained in the training data, namely supervised and unsupervised learning. Furthermore, another research direction considers the learning process as an optimization problem in which an empirical risk is minimized. We classify works in such direction as private learning in empirical risk minimization (ERM). We discuss the three classes separately.

4.1. Supervised learning

In supervised learning, the training process is done using a set of labelled training datasets. One of the simplest and most efficient supervised learning techniques is the Naive Bayes classifier. The Naive Bayes classifier training phase estimates conditional probabilities of attributes under a simple assumption that the attributes are pairwise independent. It is then easy to see that a straightforward way to provide local privacy guarantee in this process is by directly applying LDP statistical query techniques to estimate the conditional probabilities of the attributes given the labels.

There are two main issues to be considered in the design of the training procedure of the Naive Bayes classifier in a local setting, namely the effect of the high dimension of the training data and the problem of maintaining the correlation between attributes of a dataset and its label. The problem of the data having high dimension is typically solved by partitioning the users into disjoint groups where the users in each group generate their reports based on lower dimensional data. The problem of feature-label correlation is handled in different ways. Yilmaz et al. [111] preserves the feature-label correlation by defining a larger report space such that different feature-label pairs are encoded to a different report. This allows the aggregator to estimate the label probability as well as the conditional distribution of each feature given the label value. Xue et al. [112] proposed a similar solution that is based on a joint distribution estimation scheme. Specifically, the feature-label pair is perturbed in a similar manner as key–value data where the label is first perturbed before perturbing the feature with a distribution that depends on the value of the perturbed label.

Discussion. Currently, the investigation of supervised learning under LDP model is mainly confined to the Naive Bayes classification. Furthermore, the proposed solutions are also just a direct application of existing LDP statistical query techniques. A more dedicated scheme may perform better in such a problem, which shows that there is still plenty of room for improvement in the study of Naive Bayes model training. Furthermore, investigation in other supervised learning algorithms, such as the decision tree, is also essential. Despite its extensive study in the centralized setting [113], it has not been considered as extensively in the local setting. Recently, Du et al. [114] initiated a study of sanitizing sentence embeddings for fine-tuning/testing LM-based pipelines. They empirically show the promise of applying local differential privacy mechanisms, such as randomized response, in defending against privacy threats on embeddings, which is an interesting research direction to explore.

4.2. Unsupervised learning

Unsupervised learning refers to machine learning algorithms that provide inferences from datasets with no pre-existing labels. A typical unsupervised learning problem is the clustering problem where, given a set of data records, the objective is to partition the data records to disjoint groups according to their similarities. k-means clustering algorithm has been one of the most fundamental clustering schemes and it has recently garnered interest in the local differential privacy setting.

In general, the aim of k-means clustering is to reduce the dimension of the data while preserving the performance of the clustering algorithm. Nissim and Stemmer [115] reduced the clustering problem to the problem of finding a minimum enclosing ball, which, given a set of n points, a radius r, and a threshold t, outputs a ball of radius r containing at least t of the given points. They proposed an algorithm for the minimum enclosing ball utilizing a locality-sensitive hash function to perturb the data points. Two points with a small distance will have a much higher probability to be hashed to the same digest while the probability becomes very small when the distance is large. This work was then significantly improved by Stemmer and Kaplan [116] from the perspective of the communication round complexity. Furthermore, its statistical additive error was further reduced by a follow-up work in 2020 [117]. Sun et al. [118] encoded the user data to the Hamming space using a privacy-preserving bit-vector mechanism to eliminate semantic information of user data while preserving the distance information between records. Indistinguishability was then ensured by the use of further traditional randomized response on each bit of the encoded data. Xia et al. [119] proposed the use of the standard interactive K-means clustering algorithm where the centroids of the K clusters are refined iteratively. In each iteration, the randomized response is used to perturb the binary representation of features for a specified precision. To enhance the privacy guarantee, reports regarding the cluster the user belongs to for each iteration are also perturbed. Such technique is shown to provide a good balance between accuracy, communication costs, and the privacy of the scheme. Chang et al. [119] proposed a privacy-preserving non-interactive K-means algorithm by representing the data in the net tree structure [120] to form a private coreset, enabling K-means clustering to be completed in one round. By the use of such technique, the proposed solution is shown to be able to produce an estimate with an approximation ratio arbitrarily close to the approximation ratio that can be achieved by the best non-private solution.

Discussion. In the study of the clustering problem, the centre of each cluster is typically calculated as the mean of values held by users belonging to the cluster. Hence, the accuracy of the estimate is significantly affected by the size of each cluster, i.e., statistical calculations done on small clusters typically have a large statistical variance. It is then interesting to consider how such problem can be mitigated and how such mitigation approaches may affect the final clustering accuracy. Furthermore, in order to address the privacy consumption requirement of clustering algorithms, there have also been some investigations [116] that consider the possibility of reducing the number of iterations required to refine the clusters. Yang et al. [121] incorporate the distance property to bound the privacy loss. Despite such studies, the problem of optimizing the privacy consumption requirement of a clustering algorithm in a distributed setting remains a challenge.

Table 6
Private learning risk bound.

Paper	Model	Learning algorithm	Assumption for loss function	Perturbation method	Risk bound/Sample complexity	Privacy level
[122]	Non-interactive	–[a]	Convex, 1-Lipschitz	Gradient	Exponentially dependent on d[b]	ϵ
[123]	Non-interactive	Sparse linear regression	Convex	Input	Logarithmically dependent on d	(ϵ, δ)
		Kernel ridge regression	Convex, Lipschitz	Input	Polynomially dependent on d	(ϵ, δ)
		–	Convex, smooth generalized linear function	Input	-/Quasi-polynomial with respect to $\frac{1}{\alpha}$[c]	(ϵ, δ)
[124]	Non-interactive	–	(∞, T)-smooth	Objective	-/Polynomially dependent on $\frac{1}{\alpha}$ for constant or low dimensional case	ϵ
		–	Convex, generalized linear function, 1-Lipschitz	Objective	Dependent on number of users and Gaussian width of the constrained set	ϵ
[125,126]	Non-interactive	Sparse linear regression	–	Input	Dependent on $d \log d$	ϵ
	Interactive	Sparse linear regression	–	Gradient	Constant dependence on $\log d$ for low dimensional case	ϵ
	Interactive	Sparse linear regression	–	Input	Constant dependence on $\log d$ under the assumption that privacy is only required for labels for the training data	(ϵ, δ)

[a] –: Not specified.
[b] d : Data dimension.
[c] α : Upper bound on the difference between the minimum total loss function and the estimated minimum total loss function.

4.3. Private learning in ERM

Empirical risk minimization (ERM) is a typical technique used in selecting the optimal model from a given set of hypotheses through theoretical analysis of their performance. Given a dataset $D = \{r_1, \dots, r_n\}$, hypothesis $h \in \mathbf{H}$ and loss function $\ell(h(\mathbf{w}, r_i), y_i)$, the goal of empirical risk minimization is to find the $\mathbf{w}$, which can minimize the empirical risk $R_n(\mathbf{w})$ on dataset D shown in Eq. (9).

$$R_n(\mathbf{w}) = arg \min_{h \in \mathbf{H}} \frac{1}{n} \sum_{i=1}^{n} \ell(h(\mathbf{w}, r_i), y_i) \tag{9}$$

As can be observed, the empirical risk of a hypothesis is approximated by aggregating the loss function over the training dataset and it depends on the choice of the loss function. A common assumption made on the loss function to ensure the tractability of the ERM output is that it is a convex function. Based on this assumption, the problem can then be reduced to a convex optimization problem. In general, private ERM protocols can be classified as interactive and non-interactive protocols. In the interactive model, the aggregator is allowed to sequentially collect users' perturbed reports. More specifically, the report $\hat{r}_i$ from user u_i is generated as a function of his own data r_i and the previous perturbed report by the previous user u_{i-1}, $\hat{r}_{i-1}$. While for the non-interactive model, all reports are collected simultaneously. The majority of current works focus on the non-interactive model. Furthermore, private ERM protocols can also be classified into three based on the perturbation method being used, input perturbation, gradient perturbation, and objective function perturbation.

The study on private ERM was first initiated by Kasiviswanathan et al. [127] where the interactive model was considered. Following this work, Feldman et al. [128] proposed a wide range of convex ERM problems for statistical query while Duchi et al. [53,129] considered the theoretical performance bound on private schemes based on their privacy consumption. A series of works [122–124] considered the convex optimization problem under the non-interactive model. Smith et al. [122] considered convex Lipschitz functions and proposed a non-interactive scheme whose statistical accuracy decays exponentially with respect to the data dimension. Such result was then improved by Wang et al. [124] under some smoothness assumption of the loss function. More specifically, in the case when the hypotheses space has a low dimension, the exponential decay can be prevented under the assumption that all partial derivatives of the loss function at any order are bounded by a constant T. On the other hand, for higher dimensional hypotheses space, under the assumption that the loss function is a convex generalized linear function, the statistical error of the estimate can be bounded by the Gaussian width of the hypotheses space and the number of users instead of the data dimension. Zheng et al. [123] considered the case when the data has a high dimension and is ℓ_2-bounded. In such case, the linear dependence of the aggregated noise to the data dimension is prevented. Instead, they proposed algorithms with statistical error bounded by a logarithmic function of the data dimension as well as the inverse of the number of users. In such algorithms, various loss functions are considered, such as sparse linear regression and kernel ridge regression. Wang et al. [125,126] studied the ERM problem under some sparsity constraints. In such setting, they proposed an algorithm that outputs and estimates with statistical error upper bounded by the logarithm of the data dimension, which they further proved to be optimal. Van et al. [130] extended the local differential privacy framework to an unconstrained online convex optimization problem by allowing the data providers to choose their own privacy guarantees.

Discussion. The performance of privacy-preserving ERM solutions is highly dependent on the data dimension as well as the size of the training dataset. Despite the extensive studies towards the relaxation of such dependencies, for example, through the use of additional assumptions or relaxation of privacy requirements, such polynomial dependency of the performance of the solution seems unavoidable. Furthermore, restrictions made in some studies towards a more specific case of the ERM problem may hinder the practicality of the framework. The summary of some private learning algorithms along with their risk bounds can be found in Table 6.

5. Applications of LDP

5.1. Federated learning

Federated learning, which was proposed by Google in 2017 [131], is a recent advance in privacy-preserving machine learning where local

devices can participate in the model training procedure of the server using their local data without sending any information about their local data to the server except for intermediate parameters of the model. Although the only information regarding the users' private data that is provided to the server through the intermediate parameters while the actual data never leave the user's device, this training system has been shown to have various privacy vulnerabilities which are exploited in numerous attacks, for instance model inversion attack [132], membership inference attack [133], or even attacks that can be used to recover the users' sensitive data [134–141]. Although such attacks show that direct implementation of federated learning may not provide a strong privacy guarantee, it has also been empirically shown by Naseri et al. [142] that some of those attacks can be effectively defended by equipping the federated learning instantiation with LDP.

The most common tool that is adopted in federated learning is the distributed stochastic gradient descent [143]. In general, LDP can be incorporated into a federated learning scheme by having each user add Laplace or Gaussian noise to the intermediate parameter he holds before reporting it to the server [144]. In addition, some other typical LDP mechanisms such as randomized response can also be adopted to improve the scheme performance. In order to reduce the performance decay due to data dimension, the user's report can be generated based on several selected parameters instead of all the affecting parameters [145,146]. Alternatively, in order to improve the parameter aggregation accuracy, several works [147,148] proposed the use of weighted aggregation of the users' report. Solutions utilizing various privacy amplification techniques such as sub-sampling and shuffling have also been proposed to enhance privacy level [149–151]. Bhowmick et al. [152] considered a weaker adversarial assumption to obtain a scheme with higher model fitting accuracy. An alternative direction considered in [153–155] is to let each user locally train a convolutional neural network that produces a flattened vector which can then be perturbed and reported to the server to be used to train the global model. This is in contrast to previous works which have the users reporting intermediate parameters. Sun et al. [149] considered a more careful calibration of the perturbation mechanism for each training iteration that was also shown to greatly improve the accuracy of the aggregated model, especially in deeper models. In addition to the privacy and accuracy of the resulting model, various works [156–158] also incorporated other metrics such as computation complexity, communication complexity, and rate of convergence in the evaluation of a locally differentially private federated learning scheme. In those works, such a scheme was considered in various applications and they have proposed a scheme with a different balance of the performance metrics.

Discussion. Although simple incorporation of various LDP techniques may provide some privacy guarantee, it is still a challenge to effectively manage private consumption and provide a well-balanced trade-off between privacy guarantee and the accuracy of the resulting scheme. Such a challenge has been one of the main research targets in the investigation of locally differentially private federated learning, especially in the case of deep learning models that have parameter metrics with very large dimensions.

5.2. Reinforcement learning

Reinforcement learning is a type of machine learning technique that enables an agent to learn in an interactive environment. It has been well adopted in artificial intelligence (AI) [159–161] as a way of directing unsupervised machine learning through rewards and penalties in a given environment. The environment may be related to some private information, such as the private indoor layout. Pan et al. [162] showed that this private information can be inferred through the training process. Furthermore, the partial information leakage can be used to fully recover the private structure successfully.

Ono et al. [163] proposed a local differential privacy algorithm based on asynchronous advantage actor-critic (A3C) to obtain a robust policy under a distributed reinforcement learning framework. They proposed a Laplace method and a random projection method to introduce the randomness to the distributed gradient that satisfies LDP to prevent information disclosure. Gajane et al. [164] initiated the study of LDP multi-armed bandit problems and proposed an LDP bandit algorithm to hide the reward, which, in some applications, may contain users' private information. Basu et al. [165] studied the minimax lower bounds on the regret function of multi-armed bandits and they showed that the regret scales as a multiplicative factor of ϵ under the local model. Later, Ren et al. [166] proved a much tighter regret lower bound and developed corresponding LDP upper confidence bound algorithm by adding Laplace noise to the reward or converting the rewards to Bernoulli responses. Garcelon et al. [167] studied finite horizon reinforcement learning problems and established a lower bound for regret minimization.

Discussion. The target of private reinforcement learning is to hide sensitive information contained in various parameters such as states and rewards while preserving the learnability of the problem. LDP solves the privacy issue by randomizing the data before passing it to the learning agent. However, such perturbation complicates the challenging learning process. The application of LDP in this area is still at an early stage. More study is needed to help us to understand the functionality and effect of LDP in reinforcement learning.

5.3. Location privacy

GPS-enabled devices allow location information to be easily collected and provide opportunities for the development of location-based services, such as tracking systems, social network services, and location-based advertising. Despite its usefulness in the study of population distribution, location information may reveal users' private information such as their residential details, religious practice, behaviour, and habits. To combat this, LDP has been applied in the users' location data collection and analysis.

A direct method to protect the location information with LDP guarantee is to add Laplace or Gaussian noise to the location data directly, which produces unnecessarily large noise. To reduce the noise scales, relaxing the privacy requirement that protects the user's location in a smaller region instead of the whole domain is commonly used in the literature. For example, Dewri [168] proposed to protect the user's location in a region containing k locations instead of over the whole domain. Wang et al. [169] perturbed the location in a segmented region. Bi et al. [170] divided the road network space and perturbed the location in each Voronoi grid. In another work, Hong et al. [171] proposed a mechanism which gives a higher probability to output the location in a well-designed region near the original location and a low probability in the rest of the region. In addition, some work focused on protecting the aggregate information, such as the number of users at a certain location [172], the count of nearby users [173], and the number of firefighter interventions in certain localities [174]. Since location data has a large domain size, the hierarchy tree is also used as a common tool to generalize the location information to reduce the statistical variance [175]. Intuitively, a hierarchy tree contains several leaf nodes which encode several specific locations. Having this, each parent node encodes the union of the locations of its child nodes. This implies that after a specified depth, we have the root node, which is the highest level in the tree and corresponds to all the locations in the domain. Zhao et al. [176] iteratively split the node into disjoint sub-partitions and make use of the noisy count of each leaf node to construct the synthetic dataset. Chen et al. [177] allowed the user to choose a safe region (the internal node in the tree) to report, which has a much lower statistical variance compared with reporting the location in the lowest level.

Discussion. The current works on location privacy mainly focus on the population statistics or recommendations based on the user's location history [178,179]. Since the domain of location data is quite large, it is challenging to provide a strict LDP guarantee while ensuring the utility of the perturbed location in the application that targets individual location information.

5.4. Recommendation system

A recommendation system is used to predict the ratings or preferences a user would give to an item utilizing the user's past behaviour as well as similar decisions made by other users. However, given sufficient background knowledge, such a system may leak users' sensitive information including the record of their browsing, purchase as well as rating record [180]. One of the most effective techniques in constructing a privacy-preserving recommendation system is through the use of the matrix factorization decomposition method [181]. In general, locally differentially private recommendation system schemes can be classified based on the perturbation methods, namely either through the direct injection of noise to input data [182–184] or perturbation of the gradient which is then sent to the server.

Friedman et al. [185] showed that directly injecting noise into the input data leads to the best recommendation accuracy. Rahali et al. [186] encoded the users' preferences into bit string through the use of the Bloom filter and reported the obfuscated preferences to the server. Similarly, Wang et al. [187] converted user data into a feature vector and map it to a binary string through hash functions. Having the hash digests, they can then be further perturbed by the use of randomized response. Gao et al. [188] represented the user–item interaction record as a binary vector. A random bit-flipping technique is then performed on the binary vector to obtain the obfuscated data to be reported to the server. Chen et al. [189] adopted a local differential privacy mechanism to perturb the user-POI interaction data for generating dynamic POI popularity features. Neera et al. [190] perturbed user's rating data using the bounded Laplace mechanism to ensure the reported values fall within a predefined domain. Zheng et al. [191] partitioned the user's item attributes into sensitive and non-sensitive attributes then added Laplace noise to the ratings of sensitive attributes.

In matrix factorization decomposition method-based solutions, models are trained based on the gradient information of users' preference information. However, Chai et al. [192] proved that the gradient information also leaks sufficient information regarding users' sensitive data to disclose it. Jiang et al. [193] proposed a solution to this leakage problem by adding Gaussian noise to the gradient information before reporting it. The report includes both rated and unrated items to also protect such information. Shin et al. [194] proposed the perturbation of the gradient in each iteration and utilization of dimension reduction and sampling methods to improve the recommendation accuracy.

Discussion. Rating dataset is highly sparse. In practice, the recommendation based on the original dataset is not very accurate due to the numerous missing values. However, some randomized algorithms can be utilized to improve the accuracy of the recommendation scheme. For example, Yang et al. [195] utilized Johnson–Lindenstrauss transform to get a much more accurate recommendation even compared to what can be achieved by methods in the centralized model with no privacy guarantee. This suggests that such randomized methods may be explored to reduce the impact of the large noise added under the local model.

5.5. Summary

The current progress in the study of local differential privacy has mainly focused on simple statistical queries. However, despite the simplicity of the queries, they can potentially be applied to various application areas such as smart meters [196–199], smart intelligent transportation systems, crowdsourcing, and medical data analysis [200]. There have been other works considering the application of LDP in various other domains such as users' preference ranking [201], truth inference problem in sparse crowdsourcing data [202], and both interactive [203] and non-interactive [204] Private Principal Component Analysis (PCA) problem. There have also been works on the application of LDP in ultra-low power systems supporting low resolution and fixed-point hardware [205], hypothesis testing [206–208], and edge computing [209–213]. Local differential privacy still has much unknown potential, all these mentioned research is a good starting point for extending the application of local differential privacy in the future.

6. Challenges and research directions

6.1. Open problems

6.1.1. High-dimensional data

There have been extensive studies in improving the performance of LDP in applications with high-dimensional data which include the use of various techniques including user partition, binary vector encoding, and other dimension reduction techniques [214] such as the use of hash functions or matrix transformation. Although such techniques may reduce the impact of the high dimension of the data, they cause other problems such as an increase of error in construction and an increase in decoding complexity. this introduces another challenge in the design of LDP schemes, namely finding a good balance in the performance of such schemes in these metrics. Furthermore, solutions for statistical queries on multiple attributes are typically solved by the use of sampling since the majority of LDP schemes focus on single attribute queries. Sampling requires the users to be partitioned where each group of users may only report a part of the attributes. This causes the possibility that there is an insufficient number of reports in some of those attributes, causing the estimates for such attributes to have low statistical accuracy. How to effectively tackle all these issues and produce a more accurate estimate while preserving local differential privacy remains an open problem.

6.1.2. Multiple types of queries

Intuitively, LDP perturbation schemes are designed to have the noise injected in each report have zero expectation, which implies that in expectation, such noises introduced cancel each other. In order to achieve such guarantee, the perturbation mechanism is typically designed with a specific query to consider. Although the aggregation of such noisy reports may produce an accurate estimate in the predetermined query, the privacy guarantee also causes each report to have as little information as possible apart from the information it is designed to have. Because of this, such report is seldom useful for other types of queries. On the other hand, the capability to perform various data analysis is a fundamental requirement in a typical data analysis tool. The design of LDP schemes capable of providing an accurate estimate in multiple types of queries with a minimal number of reports from each user remains an open problem.

6.1.3. Dynamic data statistics

As discussed before, dynamic data is a type of data that changes over time. Hence statistical queries over such data typically also have dynamic responses, requiring periodical updates. As has previously been noted, although such problem has been extensively studied in centralized setting, it has not been extensively considered in a distributed setting. Currently, some solutions have been proposed in considered, as can be observed in Table 5. However, the proposed methods are only designed for specific scenarios and under some strong assumptions. This causes such solution to be less applicable in a real-life scenario. This suggests that such problem requires further investigation. Challenges in this problem include the reduction of the data assumption as well as the challenge of designing schemes supporting longitudinal data collection.

6.1.4. Small sample set

In general, LDP schemes require the injection of a large volume of noise to provide a strong privacy guarantee. However, in order to produce an accurate estimation based on the aggregation of the reports, a large number of reports from participants is required. This can be observed in the LDP deployment by Google and Apple which requires the collection of 12 million samples. This poses a problem in scenarios with a small number of samples. For instance, customer data for small to medium-sized enterprises or location distribution information for some places may only have a small amount of data available. This problem is amplified when high privacy is required. In a high-privacy regime, only a fraction of the sample can be effective. For instance, Duchi et al. [53] stated that when ϵ is less than $\frac{22}{35}$, the effective sample size of an ϵ-LDP scheme with n reports is only $4\epsilon^2 n$. Although some solutions have been proposed to improve statistical accuracy over small-scale populations, the effect of such solutions is limited [215,216]. Hence, designing LDP solutions that produce accurate estimates over small-scale populations remains an open problem.

6.2. Research directions

6.2.1. Relaxation of LDP

The standard local differential privacy provides a very strong guarantee in the indistinguishability of reports generated by any pair of data records. However, such guarantee may not be strictly necessary for many applications where some items in the domain may not be as sensitive to the users. This motivates the study of a relaxed local differential privacy guarantee that is defined in the context of the application. Currently, investigations of relaxed LDP have focused on location data collection such as block-structured LDP [217] and Metric LDP [216,218,219]. Liu et al. [96] considered a variant of LDP where the user is only guaranteed to be indistinguishable among a smaller group instead of the whole dataset. Zhao et al. [220] defined a flexible LDP where the report generated by the perturbation scheme may not be the same for different data values. Bhowmick [152] considered a relaxed definition of LDP, the localized version of Rényi differential privacy to account for attacker's background in the application of model training. By reducing the privacy requirement, schemes in a relaxed LDP setting may obtain significant gain in the statistical accuracy of the estimate while providing a more practical view of the application of LDP. Due to its potential in the design of practical LDP solutions, the challenge of designing privacy-preserving solutions capable of processing various data types in multiple application scenarios has become more essential and interesting.

6.2.2. Privacy amplification

In general, LDP solutions with strong LDP guarantee require large noise to be injected which causes the statistical accuracy of the resulting estimate to be low. One of the approaches that have been considered to improve statistical accuracy is privacy amplification techniques. In such approaches, ϵ_L-LDP solutions can be used to design a scheme providing ϵ_C-centralized differential privacy against the aggregator where ϵ_C is much smaller than ϵ_L. Because of this, a larger ϵ_L may instead be chosen to improve the statistical accuracy of the estimates while still maintaining a strong privacy guarantee against the aggregator. A typical privacy amplification technique is the shuffling method, which has been extensively studied [150,151,221–225]. Shuffling method places a shuffler between the users and the aggregator which obliviously shuffle the users' reports before sending them to the aggregator. Erlingsson et al. [79] showed that by using the shuffling method in a general framework that they consider, we have $\epsilon_C = O(\epsilon_L/\sqrt{n})$. This result was then extended in a simpler setting by Bittau et al. [226]. Feldman et al. [227] further improved these results by deriving a tighter bound on the privacy guarantee. Other privacy amplification techniques have also been discussed such as sub-sampling [228], iteration [229,230], diffusion mechanism [231], decentralization [232], and random check-ins [233]. Although investigations on privacy amplification techniques are still in their early stage, the current results suggest the potential of such approach towards the design of practical LDP solutions.

6.2.3. Security of LDP against malicious users

The main aim of the design of differential privacy is to preserve the privacy of users' data from the aggregator. Users only return a perturbed value to the aggregator. A natural question to consider is whether there is a way to verify that the users honestly follow the perturbation protocol and whether such solution has any tolerance against dishonest users. Currently, the security consideration of LDP against malicious users has been largely unexplored in contrast to the accuracy consideration of LDP solutions which have become the main considerations in an extensive number of studies [234]. Furthermore, this concern is confirmed by Cheu et al. [235], which showed that general LDP protocols are indeed highly vulnerable to report manipulation which causes the statistical performance of the protocols to degrade. Poisoning attack has also been shown to significantly affect common LDP solutions such as frequency estimation, mean value estimation, and heavy hitter identification [236,237]. Although it is clear that data manipulation will skew the resulting analysis, its severity and success probability differ among different LDP protocols. Such difference was studied in [238] while various prevention methods are considered by Kato et al. [239]. Developing robust and private methods [240,241] for data analysis is a promising direction to be explored in the future.

6.2.4. Privacy risk evaluation

The definitions of both CDP and LDP utilize a parameter ϵ, which represents the privacy level that is provided to the user. However, such ϵ does not provide total privacy guarantee of users' data against any kind of leakage. Bernau et al. [242] showed that even with LDP protection, attacks that distinguish datasets with the addition or removal of one user can still be successful. Moreover, although the use of ϵ provides a well-defined definition of privacy and allows a theoretical bound between privacy and accuracy to be established, it does not provide an intuitive way to measure the leakage that each user can expect and which ϵ is sufficient to provide the required privacy protection in practice. More specifically, there needs to be investigations that relate the privacy budget ϵ in some specific statistical or analytical task with the proportion of users whose data is protected and cannot be inferred from the report. Such practical privacy risk evaluation is critical in the effort of bridging the gap between the theoretical study and the practical application. This will also be helpful to organizations in judging whether their privacy solution complies with various data privacy regulations.

7. Conclusion

In general, local differential privacy provides a strong privacy guarantee to statistical calculations allowing users to keep their data on their own devices without the need of any trusted curator. However, such guarantee generally comes with a large statistical variance. It is generally accepted that to achieve the same level of statistical accuracy, LDP solutions require a quadratic amount of reports compared to their CDP counterparts. Establishing a tight bound for privacy and utility in various applications is critical for its deployment in real life. However, in general, the existing theoretical bound is very far from the trade-off achieved by existing protocols [243]. This suggests that a deeper and more extensive theoretical analysis of LDP is critical to close such a gap.

Overall, this paper presented a comprehensive survey of local differential privacy techniques. We discussed existing methods of answering different queries over different data types, and further analysed and compared the typical methods and techniques, which provides a comparative review for further research. We classified the studies of private

learning into supervised learning, unsupervised learning, and private learning in ERM. We discussed and analysed the works on such application. We also explored the application of LDP in various domains. Lastly, we identified several research challenges and discussed some potential research directions. In conclusion, the study of local differential privacy is far from complete and it still has much unknown potential.

CRediT authorship contribution statement

Mengmeng Yang: Writing – original draft, Writing – review & editing. **Taolin Guo:** Writing – original draft, Writing – review & editing. **Tianqing Zhu:** Methodology, Resources, Supervision, Writing – review & editing. **Ivan Tjuawinata:** Validation, Writing – original draft. **Jun Zhao:** Writing – review & editing. **Kwok-Yan Lam:** Writing – review & editing.

Data availability

No data was used for the research described in the article.

Acknowledgments

This research / project is supported by the National Research Foundation, Singapore under its Strategic Capability Research Centres Funding Initiative, the National Natural Science Foundation of China (No. U22A2026) and the Natural Science Foundation of Chongqing (No. CSTB2022NSCQ-MSX1383). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Y.-N. Cao, Y. Wang, Y. Ding, Z. Guo, Q. Wu, H. Liang, Blockchain-empowered security and privacy protection technologies for smart grid, Comput. Stand. Interfaces (2022) 103708.
[2] General data protection regulation, URL https://en.wikipedia.org/wiki/General_Data_Protection_Regulation.
[3] California consumer privacy act, URL https://en.wikipedia.org/wiki/California_Consumer_Privacy_Act.
[4] PDPA overview, URL https://www.pdpc.gov.sg/Overview-of-PDPA/The-Legislation/Personal-Data-Protection-Act.
[5] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, M. Naor, Our data, ourselves: Privacy via distributed noise generation, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2006, pp. 486–503.
[6] S.F. Daniel Victor, I. Kershner, Personal data of all 6.5 million Israeli voters is exposed, 2020, URL https://www.nytimes.com/2020/02/10/world/middleeast/israeli-voters-leak.html.
[7] L.H. Newman, A new Google+ blunder exposed data from 52.5 million users, 2018, URL https://www.wired.com/story/google-plus-bug-52-million-users-data-exposed/.
[8] C. Fisher, Over 267 million Facebook users reportedly had data exposed online, 2019, URL https://www.engadget.com/2019/12/19/facebook-data-exposed-online/.
[9] Ú. Erlingsson, V. Pihur, A. Korolova, Rappor: Randomized aggregatable privacy-preserving ordinal response, in: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2014, pp. 1054–1067.
[10] Apple differential privacy technical overview, URL https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf.
[11] B. Ding, J. Kulkarni, S. Yekhanin, Collecting telemetry data privately, in: Advances in Neural Information Processing Systems, 2017, pp. 3571–3580.
[12] G. Cormode, S. Jha, T. Kulkarni, N. Li, D. Srivastava, T. Wang, Privacy at scale: Local differential privacy in practice, in: Proceedings of the 2018 International Conference on Management of Data, 2018, pp. 1655–1658.
[13] P. Zhao, G. Zhang, S. Wan, G. Liu, T. Umer, A survey of local differential privacy for securing internet of vehicles, J. Supercomput. (2019) 1–22.
[14] B. Bebensee, Local differential privacy: A tutorial, 2019, arXiv preprint arXiv:1907.11908.
[15] X. Xiong, S. Liu, D. Li, Z. Cai, X. Niu, A comprehensive survey on local differential privacy, Secur. Commun. Netw. 2020 (2020).
[16] G. Cormode, S. Maddock, C. Maple, Frequency estimation under local differential privacy, Proc. VLDB Endowment 14 (11) (2021) 2046–2058.
[17] R. Bassily, A. Smith, Local, private, efficient protocols for succinct histograms, in: Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing, 2015, pp. 127–135.
[18] T.-S. Chen, K.-H. Huang, Y.-F. Chung, A practical authenticated encryption scheme based on the elliptic curve cryptosystem, Comput. Stand. Interfaces 26 (5) (2004) 461–469.
[19] V. Seničar, B. Jerman-Blažič, T. Klobučar, Privacy-enhancing technologies—approaches and development, Comput. Stand. Interfaces 25 (2) (2003) 147–158.
[20] C. Dwork, A. Roth, et al., The algorithmic foundations of differential privacy, Found. Trends® Theoretical Comput. Sci. 9 (3–4) (2014) 211–407.
[21] M. Abadi, A. Chu, I. Goodfellow, H.B. McMahan, I. Mironov, K. Talwar, L. Zhang, Deep learning with differential privacy, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 308–318.
[22] S.L. Warner, Randomized response: A survey technique for eliminating evasive answer bias, J. Amer. Statist. Assoc. 60 (309) (1965) 63–69.
[23] N. Holohan, D.J. Leith, O. Mason, Optimal differentially private mechanisms for randomised response, IEEE Trans. Inf. Forensics Secur. 12 (11) (2017) 2726–2735.
[24] P. Kairouz, K. Bonawitz, D. Ramage, Discrete distribution estimation under local privacy, in: Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, ICML '16, JMLR.org, 2016, pp. 2436—2444.
[25] T. Wang, J.Q. Chen, Z. Zhang, D. Su, Y. Cheng, Z. Li, N. Li, S. Jha, 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, 2021, pp. 1237–1253.
[26] T. Wang, J. Blocki, N. Li, S. Jha, Locally differentially private protocols for frequency estimation, in: 26th {$USENIX$} Security Symposium ({$USENIX$} Security 17), 2017, pp. 729–745.
[27] R. Tibshirani, Regression shrinkage and selection via the Lasso, J. R. Stat. Soc. Ser. B Stat. Methodol. 58 (1) (1996) 267–288.
[28] J. Chai, T.K. Nayak, Minimax randomized response methods for providing local differential privacy, Statistics (2019) 04.
[29] learning with privacy at scale, URL https://machinelearning.apple.com/docs/learning-with-privacy-at-scale/appledifferentialprivacysystem.pdf.
[30] J. Acharya, Z. Sun, H. Zhang, Hadamard response: Estimating distributions privately, efficiently, and with little communication, 2018, arXiv preprint arXiv:1802.04705.
[31] Z. Xiong, J. Sun, X. Mao, J. Wang, Y. Shan, Z. Huang, Compressive sensing approaches for sparse distribution estimation under local privacy, in: Proceedings of the ACM Web Conference 2022, 2022, pp. 599–609.
[32] S. Wang, L. Huang, P. Wang, Y. Nie, H. Xu, W. Yang, X.-Y. Li, C. Qiao, Mutual information optimally local private discrete distribution estimation, 2016, arXiv preprint arXiv:1607.08025.
[33] M. Ye, A. Barg, Optimal schemes for discrete distribution estimation under locally differential privacy, IEEE Trans. Inform. Theory 64 (8) (2018) 5662–5676.
[34] S. Wang, L. Huang, Y. Nie, X. Zhang, P. Wang, H. Xu, W. Yang, Local differential private data aggregation for discrete distribution estimation, IEEE Trans. Parallel Distrib. Syst. 30 (9) (2019) 2046–2059.
[35] H.H. Arcolezi, J.-F. Couchot, B. Al Bouna, X. Xiao, Random sampling plus fake data: Multidimensional frequency estimates with local differential privacy, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 47–57.
[36] J. Jia, N.Z. Gong, Calibrate: Frequency estimation and heavy hitter identification with local differential privacy via incorporating prior knowledge, in: IEEE INFOCOM 2019-IEEE Conference on Computer Communications, IEEE, 2019, pp. 2008–2016.
[37] H. Fang, L. Chen, Y. Liu, Y. Gao, Locally differentially private frequency estimation based on convolution framework, in: 2023 IEEE Symposium on Security and Privacy, SP, IEEE Computer Society, 2023, pp. 2208–2222.
[38] T. Wang, Z. Li, N. Li, M. Lopuhaä-Zwakenberg, B. Skoric, Consistent and accurate frequency oracles under local differential privacy, 2019, arXiv preprint arXiv:1905.08320.
[39] Z. Li, T. Wang, M. Lopuhaä-Zwakenberg, N. Li, B. Škoric, Estimating numerical distributions under local differential privacy, in: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, 2020, pp. 621–635.
[40] T. Wang, M. Lopuhaä-Zwakenberg, Z. Li, B. Skoric, N. Li, Locally differentially private frequency estimation with consistency, NDSS (2020).

[41] G. Fanti, V. Pihur, Ú. Erlingsson, Building a RAPPOR with the unknown: Privacy-preserving learning of associations and data dictionaries, Proc. Privacy Enhan. Technol. 2016 (3) (2016) 41–61.

[42] T. Wang, N. Li, S. Jha, Locally differentially private heavy hitter identification, IEEE Trans. Dependable Secure Comput. (2019).

[43] R. Bassily, K. Nissim, U. Stemmer, A.G. Thakurta, Practical locally private heavy hitters, in: Advances in Neural Information Processing Systems, 2017, pp. 2288–2296.

[44] N. Wang, X. Xiao, Y. Yang, T.D. Hoang, H. Shin, J. Shin, G. Yu, PrivTrie: Effective frequent term discovery under local differential privacy, in: 2018 IEEE 34th International Conference on Data Engineering, ICDE, IEEE, 2018, pp. 821–832.

[45] S. Kim, H. Shin, C. Baek, S. Kim, J. Shin, Learning new words from keystroke data with local differential privacy, IEEE Trans. Knowl. Data Eng. 32 (3) (2018) 479–491.

[46] X. Ren, C.-M. Yu, W. Yu, S. Yang, X. Yang, J.A. McCann, S.Y. Philip, LoPub: High-dimensional crowdsourced data publication with local differential privacy, IEEE Trans. Inf. Forensics Secur. 13 (9) (2018) 2151–2166.

[47] G. Cormode, T. Kulkarni, D. Srivastava, Marginal release under local differential privacy, in: Proceedings of the 2018 International Conference on Management of Data, ACM, 2018, pp. 131–146.

[48] Z. Zhang, T. Wang, N. Li, S. He, J. Chen, Calm: Consistent adaptive local marginal for marginal release under local differential privacy, in: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2018, pp. 212–229.

[49] Y. Nie, S. Wang, W. Yang, L. Huang, Z. Zhao, Classification learning from private data in heterogeneous settings, in: International Conference on Database Systems for Advanced Applications, Springer, 2018, pp. 577–585.

[50] E. Yilmaz, M. Al-Rubaie, J.M. Chang, Locally differentially private naive Bayes classification, 2019, arXiv preprint arXiv:1905.01039.

[51] X. Yang, T. Wang, X. Ren, W. Yu, Copula-based multi-dimensional crowdsourced data synthesis and release with local privacy, in: GLOBECOM 2017-2017 IEEE Global Communications Conference, IEEE, 2017, pp. 1–6.

[52] T. Wang, X. Yang, X. Ren, W. Yu, S. Yang, Locally private high-dimensional crowdsourced data release based on copula functions, IEEE Trans. Serv. Comput. (2019).

[53] J. Duchi, M.J. Wainwright, M.I. Jordan, Local privacy and minimax bounds: Sharp rates for probability estimation, in: Advances in Neural Information Processing Systems, 2013, pp. 1529–1537.

[54] T.T. Nguyên, X. Xiao, Y. Yang, S.C. Hui, H. Shin, J. Shin, Collecting and analyzing data from smart device users with local differential privacy, 2016, arXiv preprint arXiv:1606.05053.

[55] M. Akter, T. Hashem, Computing aggregates over numeric data with personalized local differential privacy, in: Australasian Conference on Information Security and Privacy, Springer, 2017, pp. 249–260.

[56] T. Wang, J. Zhao, X. Yang, X. Ren, Locally differentially private data collection and analysis, 2019, arXiv preprint arXiv:1906.01777.

[57] P. Kairouz, S. Oh, P. Viswanath, Extremal mechanisms for local differential privacy, in: Advances in Neural Information Processing Systems, 2014, pp. 2879–2887.

[58] N. Wang, X. Xiao, Y. Yang, J. Zhao, S.C. Hui, H. Shin, J. Shin, G. Yu, Collecting and analyzing multidimensional data with local differential privacy, in: 2019 IEEE 35th International Conference on Data Engineering, ICDE, IEEE, 2019, pp. 638–649.

[59] T. Kulkarni, Answering range queries under local differential privacy, in: Proceedings of the 2019 International Conference on Management of Data, 2019, pp. 1832–1834.

[60] G. Cormode, T. Kulkarni, D. Srivastava, Answering range queries under local differential privacy, Proc. VLDB Endow. 12 (10) (2019) 1126–1138.

[61] T. Wang, B. Ding, J. Zhou, C. Hong, Z. Huang, N. Li, S. Jha, Answering multi-dimensional analytical queries under local differential privacy, in: Proceedings of the 2019 International Conference on Management of Data, ACM, 2019, pp. 159–176.

[62] L. Du, Z. Zhang, S. Bai, C. Liu, S. Ji, P. Cheng, J. Chen, AHEAD: Adaptive hierarchical decomposition for range query under local differential privacy, in: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, 2021, pp. 1266–1288.

[63] N. Wang, Y. Wang, Z. Wang, J. Nie, Z. Wei, P. Tang, Y. Gu, G. Yu, PrivNUD: Effective range query processing under local differential privacy, in: 2023 IEEE 39th International Conference on Data Engineering, ICDE, IEEE, 2023, pp. 2660–2672.

[64] E.J. Stollnitz, T.D. DeRose, A.D. DeRose, D.H. Salesin, Wavelets for Computer Graphics: Theory and Applications, Morgan Kaufmann, 1996.

[65] J. Yang, T. Wang, N. Li, X. Cheng, S. Su, Answering multi-dimensional range queries under local differential privacy, VLDB (2020).

[66] Z. Qin, Y. Yang, T. Yu, I. Khalil, X. Xiao, K. Ren, 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, 2016, pp. 192–203.

[67] S. Wang, L. Huang, Y. Nie, P. Wang, H. Xu, W. Yang, PrivSet: Set-valued data analyses with locale differential privacy, in: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, IEEE, 2018, pp. 1088–1096.

[68] T. Wang, N. Li, S. Jha, Locally differentially private frequent itemset mining, in: 2018 IEEE Symposium on Security and Privacy, SP, IEEE, 2018, pp. 127–143.

[69] Q. Ye, H. Hu, X. Meng, H. Zheng, PrivKV: Key-value data collection with local differential privacy, in: 2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019, 2019, pp. 317–331, http://dx.doi.org/10.1109/SP.2019.00018.

[70] Q. Ye, H. Hu, X. Meng, H. Zheng, K. Huang, C. Fang, J. Shi, PrivKVM*: Revisiting key-value statistics estimation with local differential privacy, IEEE Trans. Dependable Secure Comput. (2021).

[71] X. Gu, M. Li, Y. Cheng, L. Xiong, Y. Cao, PCKV: Locally differentially private correlated key-value data collection with optimized utility, 2019, arXiv preprint arXiv:1911.12834.

[72] L. Sun, J. Zhao, X. Ye, S. Feng, T. Wang, T. Bai, Conditional analysis for key-value data with local differential privacy, 2019, arXiv preprint arXiv:1907.05014.

[73] M. Joseph, A. Roth, J. Ullman, B. Waggoner, Local differential privacy for evolving data, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, NeurIPS '18, Curran Associates Inc., Red Hook, NY, USA, 2018, pp. 2381—2390.

[74] Q. Xue, Q. Ye, H. Hu, Y. Zhu, J. Wang, DDRM: A continual frequency estimation mechanism with local differential privacy, IEEE Trans. Knowl. Data Eng. (2022).

[75] E. Bao, Y. Yang, X. Xiao, B. Ding, CGM: An enhanced mechanism for streaming data collection with local differential privacy, Proc. VLDB Endow. 14 (11) (2021) 2258–2270.

[76] X. Ren, L. Shi, W. Yu, S. Yang, C. Zhao, Z. Xu, LDP-IDS: Local Differential Privacy for Infinite Data Streams, SIGMOD, 2022.

[77] T. Wang, J.Q. Chen, Z. Zhang, D. Su, Y. Cheng, Z. Li, N. Li, S. Jha, 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, 2021, pp. 1237–1253.

[78] W. Dong, Q. Luo, K. Yi, Continual observation under user-level differential privacy, in: 2023 IEEE Symposium on Security and Privacy, SP, IEEE Computer Society, 2023, pp. 2190–2207.

[79] Ú. Erlingsson, V. Feldman, I. Mironov, A. Raghunathan, K. Talwar, A. Thakurta, Amplification by shuffling: From local to central differential privacy via anonymity, in: Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2019, pp. 2468–2479.

[80] S. Chen, W. Susilo, Y. Zhang, B. Yang, M. Zhang, Privacy-preserving anomaly counting for time-series data in edge-assisted crowdsensing, Comput. Stand. Interfaces 85 (2023) 103707.

[81] J.W. Kim, B. Jang, H. Yoo, Privacy-preserving aggregation of personal health data streams, PLoS One 13 (11) (2018).

[82] M. Yang, K.-Y. Lam, T. Zhu, C. Tang, SPoFC: A framework for stream data aggregation with local differential privacy, Concurr. Comput.: Pract. Exper. (2022) http://dx.doi.org/10.1002/cpe.7572.

[83] T. Cunningham, G. Cormode, H. Ferhatosmanoglu, D. Srivastava, Real-world trajectory sharing with local differential privacy, 2021, arXiv preprint arXiv:2108.02084.

[84] T. Wang, Z. Hu, Real-time stream statistics via local differential privacy in mobile crowdsensing, in: International Conference on Mobile Multimedia Communications, Springer, 2021, pp. 432–445.

[85] X. Fang, Q. Zeng, G. Yang, Local differential privacy for data streams, in: International Conference on Security and Privacy in Digital Economy, Springer, 2020, pp. 143–160.

[86] G. Kellaris, S. Papadopoulos, X. Xiao, D. Papadias, Differentially private event sequences over infinite streams, Proc. VLDB Endow. 7 (12) (2014) 1155–1166.

[87] X. Li, Y. Cao, M. Yoshikawa, Locally private streaming data release with shuffling and subsampling, in: 2023 IEEE 39th International Conference on Data Engineering Workshops, ICDEW, IEEE, 2023, pp. 125–131.

[88] Q. Ye, H. Hu, N. Li, X. Meng, H. Zheng, H. Yan, Beyond value perturbation: Local differential privacy in the temporal setting, in: IEEE INFOCOM 2021-IEEE Conference on Computer Communications, IEEE, 2021, pp. 1–10.

[89] P. Erdös, A. Rényi, On random graphs', math, Debrecen 6 (1959) 290–297.

[90] C. Seshadhri, T.G. Kolda, A. Pinar, Community structure and scale-free collections of Erdős-Rényi graphs, Phys. Rev. E 85 (5) (2012) 056109.

[91] W. Aiello, F. Chung, L. Lu, A random graph model for massive graphs, in: STOC, vol. 2000, Citeseer, 2000, pp. 1–10.

[92] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, Z. Ghahramani, Kronecker graphs: An approach to modeling networks, J. Mach. Learn. Res. 11 (Feb) (2010) 985–1042.

[93] T. Gao, F. Li, Y. Chen, X. Zou, Preserving local differential privacy in online social networks, in: International Conference on Wireless Algorithms, Systems, and Applications, Springer, 2017, pp. 393–405.

[94] T. Gao, F. Li, Y. Chen, X. Zou, Local differentially privately anonymizing online social networks under HRG-based model, IEEE Trans. Comput. Soc. Syst. 5 (4) (2018) 1009–1020, http://dx.doi.org/10.1109/TCSS.2018.2877045.

[95] Y. Zhang, J. Wei, X. Zhang, X. Hu, W. Liu, A two-phase algorithm for generating synthetic graph under local differential privacy, in: Proceedings of the 8th International Conference on Communication and Network Security, ACM, 2018, pp. 84–89.

[96] P. Liu, Y.X. Xu, Q. Jiang, Y. Tang, Y. Guo, L. Wang, X. Li, Local differential privacy for social network publishing, Neurocomputing 391 (2020) 273–279, http://dx.doi.org/10.1016/j.neucom.2018.11.104.

[97] Z. Qin, T. Yu, Y. Yang, I. Khalil, X. Xiao, K. Ren, Generating synthetic decentralized social graphs with local differential privacy, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2017, pp. 425–438.

[98] C. Wei, S. Ji, C. Liu, W. Chen, T. Wang, AsgLDP: Collecting and generating decentralized attributed graphs with local differential privacy, IEEE Trans. Inform. Forensics Secur. 15 (2020) 3239–3254, http://dx.doi.org/10.1109/TIFS.2020.2985524.

[99] H. Zhang, S. Latif, R. Bassily, A. Rountev, Differentially-private control-flow node coverage for software usage analysis, in: USENIX Security Symposium, USENIX Security, 2020.

[100] H. Sun, X. Xiao, I. Khalil, Y. Yang, Z. Qin, H. Wang, T. Yu, Analyzing subgraph statistics from extended local views with decentralized differential privacy, in: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 703–717.

[101] J. Imola, T. Murakami, K. Chaudhuri, Locally differentially private analysis of graph statistics, in: 30th USENIX Security Symposium, USENIX Security 21, 2021, pp. 983–1000.

[102] J. Imola, T. Murakami, K. Chaudhuri, Communication-Efficient triangle counting under local differential privacy, in: 31st USENIX Security Symposium, USENIX Security 22, USENIX Association, Boston, MA, 2022, pp. 537–554.

[103] Q. Ye, H. Hu, M.H. Au, X. Meng, X. Xiao, Towards locally differentially private generic graph metric estimation, in: 2020 IEEE 36th International Conference on Data Engineering, ICDE, IEEE, 2020, pp. 1922–1925.

[104] Q. Ye, H. Hu, M.H. Au, X. Meng, X. Xiao, LF-GDPR: A framework for estimating graph metrics with local differential privacy, IEEE Trans. Knowl. Data Eng. (2020).

[105] J. Han, M. Kamber, J. Pei, Data mining: Concepts and techniques third edition [m], Morgan Kaufmann Ser. Data Manag. Syst. 5 (4) (2011) 83–124.

[106] T. Martin, X. Zhang, M.E. Newman, Localization and centrality in networks, Phys. Rev. E 90 (5) (2014) 052808.

[107] X. Xu, N. Yuruk, Z. Feng, T.A. Schweiger, Scan: A structural clustering algorithm for networks, in: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2007, pp. 824–833.

[108] F. Wu, Y. Long, C. Zhang, B. Li, Linkteller: Recovering private edges from graph neural networks via influence analysis, in: 2022 IEEE Symposium on Security and Privacy, SP, IEEE, 2022, pp. 2005–2024.

[109] X. Zhu, V.Y. Tan, X. Xiao, Blink: Link local differential privacy in graph neural networks via Bayesian estimation, 2023, arXiv preprint arXiv:2309.03190.

[110] W. Lin, B. Li, C. Wang, Towards private learning on decentralized graphs with local differential privacy, IEEE Trans. Inf. Forensics Secur. 17 (2022) 2936–2946.

[111] E. Yilmaz, M. Al-Rubaie, J.M. Chang, Naive Bayes classification under local differential privacy, in: 2020 IEEE 7th International Conference on Data Science and Advanced Analytics, DSAA, IEEE, 2020, pp. 709–718.

[112] Q. Xue, Y. Zhu, J. Wang, Joint distribution estimation and Naïve Bayes classification under local differential privacy, IEEE Trans. Emerg. Top. Comput. (2019).

[113] S. Fletcher, M.Z. Islam, Decision tree classification with differential privacy: A survey, ACM Comput. Surv. 52 (4) (2019) 1–33.

[114] M. Du, X. Yue, S.S. Chow, H. Sun, Sanitizing sentence embeddings (and labels) for local differential privacy, in: Proceedings of the ACM Web Conference 2023, 2023, pp. 2349–2359.

[115] K. Nissim, U. Stemmer, Clustering algorithms for the centralized and local models, in: Algorithmic Learning Theory, PMLR, 2018, pp. 619–653.

[116] U. Stemmer, H. Kaplan, Differentially private k-means with constant multiplicative error, in: Advances in Neural Information Processing Systems, 2018, pp. 5431–5441.

[117] U. Stemmer, Locally private k-means clustering, in: Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2020, pp. 548–559.

[118] L. Sun, J. Zhao, X. Ye, Distributed clustering in the anonymized space with local differential privacy, 2019, arXiv preprint arXiv:1906.11441.

[119] A. Chang, B. Ghazi, R. Kumar, P. Manurangsi, Locally private k-means in one round, in: International Conference on Machine Learning, PMLR, 2021, pp. 1441–1451.

[120] S. Har-Peled, M. Mendel, Fast construction of nets in low-dimensional metrics and their applications, SIAM J. Comput. 35 (5) (2006) 1148–1184.

[121] M. Yang, I. Tjuawinata, K.-Y. Lam, K-means clustering with local d-privacy for privacy-preserving data analysis, IEEE Trans. Inf. Forensics Secur. 17 (2022) 2524–2537, http://dx.doi.org/10.1109/TIFS.2022.3189532.

[122] A. Smith, A. Thakurta, J. Upadhyay, Is interaction necessary for distributed private learning? in: 2017 IEEE Symposium on Security and Privacy, SP, IEEE, 2017, pp. 58–77.

[123] K. Zheng, W. Mou, L. Wang, Collect at once, use effectively: Making non-interactive locally private learning possible, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org, 2017, pp. 4130–4139.

[124] D. Wang, M. Gaboardi, J. Xu, Empirical risk minimization in non-interactive local differential privacy revisited, in: Advances in Neural Information Processing Systems, 2018, pp. 965–974.

[125] D. Wang, A. Smith, J. Xu, High dimensional sparse linear regression under local differential privacy: Power and limitations, in: 2018 NIPS Workshop in Privacy-Preserving Machine Learning. Vol. 235, 2018.

[126] D. Wang, J. Xu, On sparse linear regression in the local differential privacy model, in: International Conference on Machine Learning, 2019, pp. 6628–6637.

[127] S.P. Kasiviswanathan, H.K. Lee, K. Nissim, S. Raskhodnikova, A. Smith, What can we learn privately? SIAM J. Comput. 40 (3) (2011) 793–826.

[128] V. Feldman, C. Guzman, S. Vempala, Statistical query algorithms for mean vector estimation and stochastic convex optimization, in: Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2017, pp. 1265–1277.

[129] J.C. Duchi, M.I. Jordan, M.J. Wainwright, Minimax optimal procedures for locally private estimation, J. Amer. Statist. Assoc. 113 (521) (2018) 182–201.

[130] D. van der Hoeven, User-specified local differential privacy in unconstrained adaptive online learning, in: Advances in Neural Information Processing Systems, 2019, pp. 14080–14089.

[131] B. McMahan, D. Ramage, Federated Learning: Collaborative Machine Learning Without Centralized Training Data. Vol. 3, Google Research Blog, 2017.

[132] H. Ren, J. Deng, X. Xie, GRNN: Generative regression neural network —- a data leakage attack for federated learning, ACM Trans. Intell. Syst. Technol. 13 (4) (2022) 1–24.

[133] M.A. Rahman, T. Rahman, R. Laganière, N. Mohammed, Y. Wang, Membership inference attack against differentially private deep learning model, Trans. Data Privacy 11 (1) (2018) 61–79.

[134] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, H. Qi, Beyond inferring class representatives: User-level privacy leakage from federated learning, in: IEEE INFOCOM 2019-IEEE Conference on Computer Communications, IEEE, 2019, pp. 2512–2520.

[135] L. Lyu, H. Yu, Q. Yang, Threats to federated learning: A survey, 2020, arXiv preprint arXiv:2003.02133.

[136] X. Luo, Y. Wu, X. Xiao, B.C. Ooi, Feature inference attack on model predictions in vertical federated learning, in: 2021 IEEE 37th International Conference on Data Engineering, ICDE, IEEE, 2021, pp. 181–192.

[137] Y. Huang, S. Gupta, Z. Song, K. Li, S. Arora, Evaluating gradient inversion attacks and defenses in federated learning, Adv. Neural Inf. Process. Syst. 34 (2021).

[138] M. Lam, G.-Y. Wei, D. Brooks, V.J. Reddi, M. Mitzenmacher, Gradient disaggregation: Breaking privacy in federated learning by reconstructing the user participant matrix, in: International Conference on Machine Learning, PMLR, 2021, pp. 5959–5968.

[139] X. Jiang, X. Zhou, J. Grossklags, Comprehensive analysis of privacy leakage in vertical federated learning during prediction, Proc. Privacy Enhan. Technol. 2 (2022) 263–281.

[140] C. Chen, L. Lyu, H. Yu, G. Chen, Practical attribute reconstruction attack against federated learning, IEEE Trans. Big Data (2022).

[141] X. Zhang, C. Chen, Y. Xie, X. Chen, J. Zhang, Y. Xiang, A survey on privacy inference attacks and defenses in cloud-based deep neural network, Comput. Stand. Interfaces 83 (2023) 103672.

[142] M. Naseri, J. Hayes, E. De Cristofaro, Local and central differential privacy for robustness and privacy in federated learning, in: Network and Distributed System Security Symposium, 2022.

[143] C. Wang, X. Wu, G. Liu, T. Deng, K. Peng, S. Wan, Safeguarding cross-silo federated learning with local differential privacy, Digit. Commun. Netw. (2021).

[144] A.K. Nair, J. Sahoo, E.D. Raj, Privacy preserving federated learning framework for IoMT based big data analysis using edge computing, Comput. Stand. Interfaces 86 (2023) 103720.

[145] R. Liu, Y. Cao, M. Yoshikawa, H. Chen, FedSel: Federated SGD under local differential privacy with top-k dimension selection, in: International Conference on Database Systems for Advanced Applications, Springer, 2020, pp. 485–501.

[146] Y. Zhao, J. Zhao, M. Yang, T. Wang, N. Wang, L. Lyu, D. Niyato, K.-Y. Lam, Local differential privacy-based federated learning for Internet of Things, IEEE Internet Things J. 8 (11) (2020) 8836–8853.

[147] Y. Lu, X. Huang, Y. Dai, S. Maharjan, Y. Zhang, Differentially private asynchronous federated learning for mobile edge computing in urban informatics, IEEE Trans. Ind. Inform. 16 (3) (2019) 2134–2143.

[148] Z. Lian, Q. Yang, Q. Zeng, C. Su, WebFed: Cross-platform federated learning framework based on web browser with local differential privacy, in: ICC 2022 - IEEE International Conference on Communications, 2022, pp. 2071–2076, http://dx.doi.org/10.1109/ICC45855.2022.9838421.

[149] L. Sun, J. Qian, X. Chen, LDP-FL: Practical private aggregation in federated learning with local differential privacy, in: Z.-H. Zhou (Ed.), Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, International Joint Conferences on Artificial Intelligence Organization, 2021, pp. 1571–1578, http://dx.doi.org/10.24963/ijcai.2021/217, Main Track.

[150] A. Girgis, D. Data, S. Diggavi, P. Kairouz, A.T. Suresh, Shuffled model of differential privacy in federated learning, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2021, pp. 2521–2529.

[151] A.M. Girgis, D. Data, S. Diggavi, P. Kairouz, A.T. Suresh, Shuffled model of federated learning: Privacy, accuracy and communication trade-offs, IEEE J. Selected Areas Inform. Theory 2 (1) (2021) 464–478.

[152] A. Bhowmick, J.C. Duchi, J. Freudiger, G. Kapoor, R. Rogers, Protection against reconstruction and its applications in private federated learning, 2018, CoRR abs/1812.00984, arXiv:1812.00984.

[153] P.C.M. Arachchige, P. Bertok, I. Khalil, D. Liu, S. Camtepe, M. Atiquzzaman, Local differential privacy for deep learning, IEEE Internet Things J. (2019).

[154] M. Chamikara, D. Liu, S. Camtepe, S. Nepal, M. Grobler, P. Bertok, I. Khalil, Local differential privacy for federated learning in industrial settings, 2022, arXiv preprint arXiv:2202.06053.

[155] P.C. Mahawaga Arachchige, D. Liu, S. Camtepe, S. Nepal, M. Grobler, P. Bertok, I. Khalil, Local differential privacy for federated learning, in: European Symposium on Research in Computer Security, Springer, 2022, pp. 195–216.

[156] H. Cao, S. Liu, R. Zhao, X. Xiong, IFed: A novel federated learning framework for local differential privacy in power Internet of Things, Int. J. Distrib. Sens. Netw. 16 (5) (2020) 1550147720919698.

[157] M. Kim, O. Günlü, R.F. Schaefer, Federated learning with local differential privacy: Trade-offs between privacy, utility, and communication, in: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2021, pp. 2650–2654.

[158] K. Wei, J. Li, M. Ding, C. Ma, H.H. Yang, F. Farokhi, S. Jin, T.Q. Quek, H.V. Poor, Federated learning with differential privacy: Algorithms and performance analysis, IEEE Trans. Inf. Forensics Secur. 15 (2020) 3454–3469.

[159] T. Zhu, D. Ye, W. Wang, W. Zhou, S.Y. Philip, More than privacy: Applying differential privacy in key areas of artificial intelligence, IEEE Trans. Knowl. Data Eng. 34 (6) (2020) 2824–2843.

[160] H. Xu, T. Zhu, L. Zhang, W. Zhou, P.S. Yu, Machine unlearning: A survey, ACM Comput. Surv. 56 (1) (2023) 1–36.

[161] S. Zhou, C. Liu, D. Ye, T. Zhu, W. Zhou, P.S. Yu, Adversarial attacks and defenses in deep learning: From a perspective of cybersecurity, ACM Comput. Surv. 55 (8) (2022) 1–39.

[162] X. Pan, W. Wang, X. Zhang, B. Li, J. Yi, D. Song, How you act tells a lot: Privacy-leaking attack on deep reinforcement learning, in: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, 2019, pp. 368–376.

[163] H. Ono, T. Takahashi, Locally private distributed reinforcement learning, 2020, arXiv preprint arXiv:2001.11718.

[164] P. Gajane, T. Urvoy, E. Kaufmann, Corrupt bandits for preserving local privacy, in: Algorithmic Learning Theory, 2018, pp. 387–412.

[165] D. Basu, C. Dimitrakakis, A. Tossou, Differential privacy for multi-armed bandits: What is it and what is its cost? 2019, arXiv preprint arXiv:1905.12298.

[166] W. Ren, X. Zhou, J. Liu, N.B. Shroff, Multi-armed bandits with local differential privacy, 2020, arXiv preprint arXiv:2007.03121.

[167] E. Garcelon, V. Perchet, C. Pike-Burke, M. Pirotta, Local differential privacy for regret minimization in reinforcement learning, Adv. Neural Inf. Process. Syst. 34 (2021).

[168] R. Dewri, Local differential perturbations: Location privacy under approximate knowledge attackers, IEEE Trans. Mob. Comput. 12 (12) (2012) 2360–2372.

[169] J. Wang, Y. Wang, G. Zhao, Z. Zhao, Location protection method for mobile crowd sensing based on local differential privacy preference, Peer–Peer Netw. Appl. 12 (5) (2019) 1097–1109.

[170] M. Bi, Y. Wang, Z. Cai, X. Tong, A privacy-preserving mechanism based on local differential privacy in edge computing, China Commun. 17 (9) (2020) 50–65.

[171] D. Hong, W. Jung, K. Shim, Collecting geospatial data with local differential privacy for personalized services, in: 2021 IEEE 37th International Conference on Data Engineering, ICDE, IEEE, 2021, pp. 2237–2242.

[172] X. Gu, M. Li, Y. Cao, L. Xiong, Supporting both range queries and frequency estimation with local differential privacy, in: 2019 IEEE Conference on Communications and Network Security, CNS, IEEE, 2019, pp. 124–132.

[173] F.Z. Errounda, Y. Liu, Collective location statistics release with local differential privacy, Future Gener. Comput. Syst. 124 (2021) 174–186.

[174] H.H. Arcolezi, J.-F. Couchot, S. Cerna, C. Guyeux, G. Royer, B. Al Bouna, X. Xiao, Forecasting the number of firefighter interventions per region with local-differential-privacy-based data, Comput. Secur. 96 (2020) 101888.

[175] J.W. Kim, B. Jang, Workload-aware indoor positioning data collection via local differential privacy, IEEE Commun. Lett. 23 (8) (2019) 1352–1356.

[176] X. Zhao, Y. Li, Y. Yuan, X. Bi, G. Wang, LDPart: Effective location-record data publication via local differential privacy, IEEE Access 7 (2019) 31435–31445.

[177] R. Chen, H. Li, A.K. Qin, S.P. Kasiviswanathan, H. Jin, Private spatial data aggregation in the local setting, in: 2016 IEEE 32nd International Conference on Data Engineering, ICDE, IEEE, 2016, pp. 289–300.

[178] M. Asada, M. Yoshikawa, Y. Cao, "When and where do you want to hide?"–recommendation of location privacy preferences with local differential privacy, in: IFIP Annual Conference on Data and Applications Security and Privacy, Springer, 2019, pp. 164–176.

[179] T. Bao, L. Xu, L. Zhu, L. Wang, T. Li, Successive point-of-interest recommendation with personalized local differential privacy, IEEE Trans. Veh. Technol. 70 (10) (2021) 10477–10488.

[180] J.A. Calandrino, A. Kilzer, A. Narayanan, E.W. Felten, V. Shmatikov, " You might also like:" Privacy risks of collaborative filtering, in: 2011 IEEE Symposium on Security and Privacy, IEEE, 2011, pp. 231–246.

[181] R. Barathy, P. Chitra, Applying matrix factorization in collaborative filtering recommender systems, in: 2020 6th International Conference on Advanced Computing and Communication Systems, ICACCS, IEEE, 2020, pp. 635–639.

[182] T. Guo, J. Luo, K. Dong, M. Yang, Locally differentially private item-based collaborative filtering, Inform. Sci. 502 (2019) 229–246.

[183] E. Xue, R. Guo, F. Zhang, L. Wang, X. Zhang, G. Qu, Distributed differentially private matrix factorization based on ADMM, in: 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems, HPCC/SmartCity/DSS, IEEE, 2019, pp. 2502–2507.

[184] Y. Shen, H. Jin, Epicrec: Towards practical differentially private framework for personalized recommendation, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2016, pp. 180–191.

[185] A. Friedman, S. Berkovsky, M.A. Kaafar, A differential privacy framework for matrix factorization recommender systems, User Model. User-Adapted Interact. 26 (5) (2016) 425–458.

[186] S. Rahali, M. Laurent, S. Masmoudi, C. Roux, B. Mazeau, A validated privacy-utility preserving recommendation system with local differential privacy, 2021, arXiv preprint arXiv:2109.11340.

[187] X. Wang, R. Yuan, J. Xu, S. Meng, A privacy-aware multi-preference-based collaborative filtering recommendation system with LSH, in: 2021 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress, DASC/PiCom/CBDCom/CyberSciTech, IEEE, 2021, pp. 397–404.

[188] C. Gao, C. Huang, D. Lin, D. Jin, Y. Li, DPLCF: Differentially private local collaborative filtering, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 961–970.

[189] C. Chen, J. Zhou, B. Wu, W. Fang, L. Wang, Y. Qi, X. Zheng, Practical privacy preserving POI recommendation, ACM Trans. Intell. Syst. Technol. 11 (5) (2020) 1–20.

[190] J. Neera, X. Chen, N. Aslam, Z. Shu, Local differentially private matrix factorization with MoG for recommendations, in: IFIP Annual Conference on Data and Applications Security and Privacy, Springer, 2020, pp. 208–220.

[191] X. Zheng, M. Guan, X. Jia, L. Guo, Y. Luo, A matrix factorization recommendation system-based local differential privacy for protecting users' sensitive data, IEEE Trans. Comput. Soc. Syst. (2022).

[192] D. Chai, L. Wang, K. Chen, Q. Yang, Secure federated matrix factorization, IEEE Intell. Syst. 36 (5) (2020) 11–20.

[193] J.-Y. Jiang, C.-T. Li, S.-D. Lin, Towards a more reliable privacy-preserving recommender system, Inform. Sci. 482 (2019) 248–265.

[194] H. Shin, S. Kim, J. Shin, X. Xiao, Privacy enhanced matrix factorization for recommendation with local differential privacy, IEEE Trans. Knowl. Data Eng. 30 (9) (2018) 1770–1782.

[195] M. Yang, T. Zhu, L. Ma, Y. Xiang, W. Zhou, Privacy preserving collaborative filtering via the Johnson-Lindenstrauss transform, in: 2017 IEEE Trustcom/BigDataSE/ICESS, IEEE, 2017, pp. 417–424.

[196] L. Ou, Z. Qin, S. Liao, T. Li, D. Zhang, Singular spectrum analysis for local differential privacy of classifications in the smart grid, IEEE Internet Things J. (2020).

[197] H. Cao, S. Liu, R. Zhao, X. Xiong, Ifed: A novel federated learning framework for local differential privacy in power Internet of Things, Int. J. Distrib. Sens. Netw. 16 (5) (2020) http://dx.doi.org/10.1177/1550147720919698.

[198] N. Gai, K. Xue, B. Zhu, J. Yang, J. Liu, D. He, An efficient data aggregation scheme with local differential privacy in smart grid, Digit. Commun. Netw. (2022).

[199] R. Leszczyna, Cybersecurity and privacy in standards for smart grids–a comprehensive survey, Comput. Stand. Interfaces 56 (2018) 62–73.

[200] J.W. Kim, B. Jang, H. Yoo, Privacy-preserving aggregation of personal health data streams, PLoS One 13 (11) (2018) e0207639.

[201] J. Yang, X. Cheng, S. Su, R. Chen, Q. Ren, Y. Liu, Collecting preference rankings under local differential privacy, in: 2019 IEEE 35th International Conference on Data Engineering, ICDE, IEEE, 2019, pp. 1598–1601.

[202] H. Sun, B. Dong, H.W. Wang, T. Yu, Z. Qin, Truth inference on sparse crowdsourcing data with local differential privacy, in: 2018 IEEE International Conference on Big Data (Big Data), IEEE, 2018, pp. 488–497.

[203] J. Ge, Z. Wang, M. Wang, H. Liu, Minimax-optimal privacy-preserving sparse pca in distributed systems, in: International Conference on Artificial Intelligence and Statistics, 2018, pp. 1589–1598.

[204] D. Wang, J. Xu, Principal component analysis in the local differential privacy model, Theoret. Comput. Sci. 809 (2020) 296–312.

[205] W.-S. Choi, M. Tomei, J.R.S. Vicarte, P.K. Hanumolu, R. Kumar, Guaranteeing local differential privacy on ultra-low-power systems, in: 2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture, ISCA, IEEE, 2018, pp. 561–574.
[206] B. Ding, H. Nori, P. Li, J. Allen, Comparing population means under local differential privacy: With significance and power, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
[207] M. Gaboardi, R. Rogers, O. Sheffet, Locally private mean estimation: Z-test and tight confidence intervals, 2018, arXiv preprint arXiv:1810.08054.
[208] O. Sheffet, Locally private hypothesis testing, 2018, arXiv preprint arXiv:1802.03441.
[209] M. Yang, I. Tjuawinata, K.Y. Lam, J. Zhao, L. Sun, Secure hot path crowdsourcing with local differential privacy under fog computing architecture, IEEE Trans. Serv. Comput. 15 (4) (2022) 2188–2201, http://dx.doi.org/10.1109/TSC.2020.3039336.
[210] Y. Tan, W. Wu, J. Liu, H. Wang, M. Xian, Lightweight edge-based kNN privacy-preserving classification scheme in cloud computing circumstance, Concurr. Comput.: Pract. Exper. 32 (19) (2020) http://dx.doi.org/10.1002/cpe.5804.
[211] Y. Tian, J. Yuan, S. Yu, Y. Hou, LEP-CNN: A lightweight edge device assisted privacy-preserving CNN inference solution for IoT, 2019, CoRR abs/1901.04100, arXiv:1901.04100.
[212] C. Xu, J. Ren, L. She, Y. Zhang, Z. Qin, K. Ren, EdgeSanitizer: Locally differentially private deep inference at the edge for mobile data analytics, IEEE Internet Things J. 6 (3) (2019) 5140–5151, http://dx.doi.org/10.1109/JIOT.2019.2897005.
[213] M. Usman, M.A. Jan, D. Puthal, PAAL: A framework based on authentication, aggregation, and local differential privacy for Internet of Multimedia Things, IEEE Internet Things J. 7 (4) (2020) 2501–2508, http://dx.doi.org/10.1109/JIOT.2019.2936512.
[214] D. Wang, J. Xu, Principal component analysis in the local differential privacy model, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, AAAI Press, 2019, pp. 4795–4801.
[215] T. Murakami, H. Hino, J. Sakuma, Toward distribution estimation under local differential privacy with small samples, Proc. Privacy Enhan. Technol. 2018 (3) (2018) 84–104.
[216] M.E. Gursoy, A. Tamersoy, S. Truex, W. Wei, L. Liu, Secure and utility-aware data collection with condensed local differential privacy, IEEE Trans. Dependable Secure Comput. 18 (5) (2019) 2365–2378.
[217] J. Acharya, K. Bonawitz, P. Kairouz, D. Ramage, Z. Sun, Context-aware local differential privacy, 2019, arXiv preprint arXiv:1911.00038.
[218] M. Alvim, K. Chatzikokolakis, C. Palamidessi, A. Pazii, Local differential privacy on metric spaces: Optimizing the trade-off with utility, in: 2018 IEEE 31st Computer Security Foundations Symposium, CSF, IEEE, 2018, pp. 262–267.
[219] Z. Xiang, B. Ding, X. He, J. Zhou, Linear and range counting under metric-based local differential privacy, 2019, arXiv:1909.11778.
[220] D. Zhao, H. Chen, S. Zhao, R. Liu, C. Li, X. Zhang, FLDP: Flexible strategy for local differential privacy, in: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2022, pp. 2974–2978.
[221] A. Cheu, A. Smith, J. Ullman, D. Zeber, M. Zhilyaev, Distributed differential privacy via shuffling, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2019, pp. 375–403.
[222] A.M. Girgis, D. Data, S. Diggavi, A.T. Suresh, P. Kairouz, On the renyi differential privacy of the shuffle model, in: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, 2021, pp. 2321–2341.
[223] T. Wang, M. Xu, B. Ding, J. Zhou, N. Li, S. Jha, Practical and robust privacy amplification with multi-party differential privacy, 2019, arXiv preprint arXiv:1908.11515.
[224] M. Scott, G. Cormode, C. Maple, Aggregation and transformation of vector-valued messages in the shuffle model of differential privacy, IEEE Trans. Inf. Forensics Secur. 17 (2022) 612–627.
[225] V. Feldman, A. McMillan, K. Talwar, Stronger privacy amplification by shuffling for Rényi and approximate differential privacy, in: Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, SIAM, 2023, pp. 4966–4981.
[226] B. Balle, J. Bell, A. Gascón, K. Nissim, The privacy blanket of the shuffle model, in: Annual International Cryptology Conference, Springer, 2019, pp. 638–667.
[227] V. Feldman, A. McMillan, K. Talwar, Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling, in: 2021 IEEE 62nd Annual Symposium on Foundations of Computer Science, FOCS, IEEE, 2022, pp. 954–964.
[228] B. Balle, G. Barthe, M. Gaboardi, Privacy profiles and amplification by subsampling, J. Privacy Confidential. 10 (1) (2020).
[229] V. Feldman, I. Mironov, K. Talwar, A. Thakurta, Privacy amplification by iteration, in: 2018 IEEE 59th Annual Symposium on Foundations of Computer Science, FOCS, IEEE, 2018, pp. 521–532.
[230] S. Asoodeh, M. Diaz, F.P. Calmon, Privacy amplification of iterative algorithms via contraction coefficients, 2020, arXiv preprint arXiv:2001.06546.
[231] B. Balle, G. Barthe, M. Gaboardi, J. Geumlek, Privacy amplification by mixing and diffusion mechanisms, in: Advances in Neural Information Processing Systems, 2019, pp. 13277–13287.
[232] E. Cyffers, A. Bellet, Privacy Amplification by Decentralization, AISTATS, 2022.
[233] B. Balle, P. Kairouz, B. McMahan, O. Thakkar, A. Guha Thakurta, Privacy amplification via random check-ins, Adv. Neural Inf. Process. Syst. 33 (2020) 4623–4634.
[234] K. Cai, X. Lei, J. Wei, X. Xiao, Data synthesis via differentially private markov random fields, Proc. VLDB Endow. 14 (11) (2021) 2190–2202.
[235] A. Cheu, A. Smith, J. Ullman, Manipulation attacks in local differential privacy, in: 2021 IEEE Symposium on Security and Privacy, SP, IEEE, 2021, pp. 883–900.
[236] X. Cao, J. Jia, N.Z. Gong, Data poisoning attacks to local differential privacy protocols, in: 30th USENIX Security Symposium, USENIX Security 21, 2021, pp. 947–964.
[237] Y. Wu, X. Cao, J. Jia, N.Z. Gong, Poisoning attacks to local differential privacy protocols for $\{Key-Value\}$ data, in: 31st USENIX Security Symposium, USENIX Security 22, 2022, pp. 519–536.
[238] M.E. Gursoy, L. Liu, K.-H. Chow, S. Truex, W. Wei, An adversarial approach to protocol analysis and selection in local differential privacy, IEEE Trans. Inf. Forensics Secur. (2022).
[239] F. Kato, Y. Cao, M. Yoshikawa, Preventing manipulation attack in local differential privacy using verifiable randomization mechanism, in: IFIP Annual Conference on Data and Applications Security and Privacy, Springer, 2021, pp. 43–60.
[240] J. Chhor, F. Sentenac, Robust estimation of discrete distributions under local differential privacy, in: International Conference on Algorithmic Learning Theory, PMLR, 2023, pp. 411–446.
[241] M. Li, T.B. Berrett, Y. Yu, On robustness and local differential privacy, Ann. Statist. 51 (2) (2023) 717–737.
[242] D. Bernau, J. Robl, P.W. Grassal, S. Schneider, F. Kerschbaum, Comparing local and central differential privacy using membership inference attacks, in: IFIP Annual Conference on Data and Applications Security and Privacy, Springer, 2021, pp. 22–42.
[243] M. Lopuhaä-Zwakenberg, Z. Li, B. Škorić, N. Li, Improving frequency estimation under local differential privacy, in: Proceedings of the 19th Workshop on Privacy in the Electronic Society, 2020, pp. 123–135.